

## Design and Validation of a Route Planner for Logistic UAV Swarm

Meng-Tse Lee<sup>1,\*</sup>, Ying-Chih Lai<sup>2</sup>, Ming-Lung Chuang<sup>1</sup> and Bo-Yu Chen<sup>1</sup>

<sup>1</sup>Department of Automation Engineering, National Formosa University, Yunlin, 632, Taiwan

<sup>2</sup>Department of Aeronautics and Astronautics Engineering, National Cheng-Kung University, Tainan, 701, Taiwan

\*Corresponding Author: Meng-Tse Lee. Email: mtlee@nfu.edu.tw

Received: 16 November 2020; Accepted: 17 December 2020

**Abstract:** Unmanned Aerial Vehicles (UAV) are widely used in different fields of aviation today. The efficient delivery of packages by drone may be one of the most promising applications of this technology. In logistic UAV missions, due to the limited capacities of power supplies, such as fuel or batteries, it is almost impossible for one unmanned vehicle to visit multiple wide areas. Thus, multiple unmanned vehicles with well-planned routes become necessary to minimize the unnecessary consumption of time, distance, and energy while carrying out the delivery missions. The aim of the present study was to develop a multiple-vehicle mission dispatch system that can automatically compile a set of optimal paths and avoid passing through no-travel zones. For this function, the A\* search algorithm was adopted to determine an alternative path that does not cross the no-travel zone when the distance array is set, and an improved two-phased Tabu search was applied to converge any initial solutions into a feasible solution. In this study, a group of five multicopters was set up to validate the swarm system, and the result shows that our improved 2TS+2OPT is able to converge to a better solution that allows logistic UAV swarms to operate in a more efficient way.

**Keywords:** Logistic UAV swarm; route planner; 4G LTE; tabu search

### 1 Introduction

#### 1.1 Introduction and Objectives

With the boom in UAV development in the past ten years, the vehicles themselves and related technology have become more popular and well-developed. Now, UAVs are not only used in military applications but also applied in various commercial and leisure activities. While UAVs have been used in different commercial fields, such as the video and film industry, and in visual applications such as lightshows, the use of drones in logistics to deliver parcels to the end customer has already begun development and evaluation, as shown in Fig. 1.

With regard to logistic missions, aiming at logistic mission, UAV delivery would not be possible without communication technology, and long distance communication, like 4G LTE, is definitely one of the current areas of focus. With its higher data transmission rate, 4G LTE has been widely implemented for smart phones



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

since the year 2013. With the development of communication technology, users are able to access 4G networks and connect to the Internet at anytime, anywhere [1].



**Figure 1:** Delivery drone

Along with the mission scale and complexity of UAV delivery, mission areas have become wider and a single vehicle lacks the range to complete the all mission. For example, a multi-robot team can usually shorten the time required for a search-and-rescue mission or reduce the energy cost for large-area cargo delivery. Controlling unmanned vehicle swarms via human supervision is of great interest to private logistics companies. As the area of drone missions expands, to complete tasks efficiently (in time or in energy), a well-planned path program is a key for an unmanned vehicle swarm . The aim of this research was to develop a group of drones as a multi-agent system (MAS) with multi-path programming under the constraint of a balance distance [2,3].

## 1.2 Related Works

Most multi-target, multi-vehicle path planning and scheduling problems revolve around vehicle routing problems (VRP) and multiple traveling salesman problem (MTSP). MTSP seeks to minimize the total travel distance by multiple salespeople visiting multiple cities at most one time each, and each starting and ending in the same city. One possible solution is that all but one salesperson visits one city only, while the remaining salesperson visits all the remaining cities, however this solution is inappropriate for practical applications. In fact, each salesperson has similar abilities, and these abilities are limited [4]. Therefore, the MTSP problem with capacity constraints is more appropriate for real world conditions. Assume that each salesman travels to a limited number of cities. The multi-target multi-robot exploration problem is extended by TSP to create the multi-traveling robot problem (MTRP), in which the robot team visits the target point at least once. The quality of the overall solution depends on the quality of the robot's route and the effectiveness of assigning targets to all robots [5].

Since this is a large-scale mathematical model problem (i.e., NP-hard), the complexity of the solution will increase with the number of visited points. Finding all possible paths and an optimal solution in a reasonable time is very difficult, so heuristic algorithms are generally used to find approximate solutions. Yousefikhoshbakht et al. [6] used a new modified ant colony optimization (NMACO) that improves on the original ant algorithm by integrates an insertion method, and exchange method and 2-Opt conversion rules, along with an effective candidate list to provide improved efficiency over the original ant colony optimization (ACO). They applied NMACO to the open international benchmark test questions, and while they found it provided better results than other algorithms, it still did not obtain the best solution. Necula et al. [7] used five improved ACOs to solve the MTSP, constraining the upper limit L and lower limit K of the number of cities that each salesperson can visit, and then tested it with the International

Benchmarking Test (TSPLIB). Xu et al. [8] used genetic algorithm (GA) combined with simulated annealing (SA) to solve the MTSP problem. Compared with the original GA, their approach achieved better local search ability, and outperformed SA, again without obtaining the best solution. Neither GA nor ACO can obtain the same solution each time, and the optimal solution can only be obtained through continuous testing.

Côté et al. [9] used the Tabu search (TS) to solve the vehicle routing problem (VRP) and tested it with the VRP benchmark test, not only achieving convergence in a reasonable time, but also breaking many records for obtaining optimal solutions.

Huang and Liao [10] proposed a two-stage dynamic route planning method to solve the time-sensitive Dynamic Vehicle Routing Problem (DVRP). Different from MTSP, this is limited by vehicle capacity. In the first stage, the Fuzzy C-mean grouping technique is used to group existing customers, and an initial solution is obtained using the cost method. Then, in the second stage, the taboo search method combined with 2-Opt. The exchange method is used for inter-route improvement and Or-Opt is used for intra-route improvement to solve multi-vehicle path planning.

Sariel-Talay et al. [5] developed a platform on which MTRP can instantly find the best path for multiple target points, with the robot group visiting the target points. Although it fulfills the function of path planning and failure compensation, it does not obtain the best solution because it requires successful robots to complete their own tasks and then attempt the tasks left by failed robots, rather than having failed robots re-attempt their own tasks. In addition, it has not been tested outdoors or in authentic environments, and does not consider maximum distance limitations (e.g., battery limit).

Many studies have examined multi-vehicle path planning, but few have verified the results of multi-vehicle path planning in field experiments using drones, and most also fail to consider prohibited areas and vehicle failures. Moreover, given that MTSP includes the upper and lower limits for how many times each salesman visits a city, it is still possible that a drone will be unable to complete its assigned task due to insufficient battery life, so this limit is changed to a maximum distance limit to ensure that each drone can reach each target point.

In addition, an elite policy similar to a genetic algorithm is added to effectuate improvement between routes. The best and second best solutions are recorded and selected as the initial solution for improvement in the next route, thus avoiding falling into the best local solution. This research uses 2-Opt as the exchange method for improvement between routes, and then returns to the continuous improvement algorithm between the routes until the stop condition is achieved. The results are then measured on the actual machine, and the difference in distance between the path plan and the theoretical value of the overall path are discussed.

In this study, the ultimate goal of path planning is to allocate waypoints to multiple UAVs to minimize total energy consumption. Due to the drones' battery life restrictions, their maximum distance must be limited.

## 2 Design and Principle of Path Programming

Path optimization is used to minimize energy and time waste by multi-aircraft logistics drones. The goal of this research is to develop a multi-aircraft drone system that can automatically plan optimal vehicle paths and avoid entering no-fly zones. The problem definition model is modified from the standard Single-Depot Multiple Traveling Salesman Problem (SD-MTSP) [11], and its target and restrictive expressions are as follows:

$$x_{ijk} = \begin{cases} 1 & \text{the } k \text{ path goes from city } i \text{ to city } j \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$$\forall (i, j) \in A, k = 1, \dots, m \quad (2)$$

Object:

$$\min \left( \max_k \left( \sum_{(i,j) \in A} c_{ij} x_{ijk} \right), k = 1, \dots, m \right) \quad (3)$$

Subject to:

$$\sum_{k=1}^m \sum_{j=2}^n x_{1jk} = m \quad (4)$$

$$\sum_{k=1}^m \sum_{j=2}^n x_{j1k} = m \quad (5)$$

$$\sum_{k=1}^m \sum_{i=1}^n x_{ijk} = 1, j = 2, \dots, n \quad (6)$$

$$\sum_{k=1}^m \sum_{j=1}^n x_{ijk} = 1, i = 2, \dots, n \quad (7)$$

$$\sum_{k=1}^m x_{i1k} + x_{1ik} \leq 1, i = 2, \dots, n \quad (8)$$

$$u_i \geq 1, i = 2, \dots, n \quad (9)$$

Where  $c_{ij}$  is the distance array of  $A$ ,  $m$  is the number of UAVs, and  $n$  is the number of target points, referring to the number of cities designated as target points  $i$  departing from the origin. Eq. (1) is an integer variable, used to determine whether the  $k$ th machine has a path from the target point  $i$  to the target point  $j$ . In Eq. (2),  $A$  is the set of all road sections (including alternative road sections that avoid the no-fly zone), and Eq. (3) is the target function of this problem. Eqs. (4–9) are restrictions, while Eq. (4) mainly restricts all drones from exceeding the maximum distance limit, and Eqs. (5) and (6) ensure that all drones start from and return to the origin after visiting the target point. Eqs. (7) and (8) require a drone to access and leave each target point. Eq. (9) is a restriction to avoid sub-loops.

This problem is NP-hard, so heuristic algorithms are generally used to solve this type of combinatorial optimization problem. While there are many kinds of heuristic algorithms, this research mainly relies on the  $A^*$  search method and tabu algorithm.

### 2.1 $A^*$ Search Algorithm

When searching for a path, the evaluation function  $f(n)$  is the main core. As shown in Eq. (10), this is the sum of  $g(n)$  and  $h(n)$ , while  $g(n)$  represents the travel distance from the starting point to any node  $n$ .  $h(n)$  represents the linear distance from any node  $n$  to the target point.

$$f(n) = g(n) + h(n) \quad (10)$$

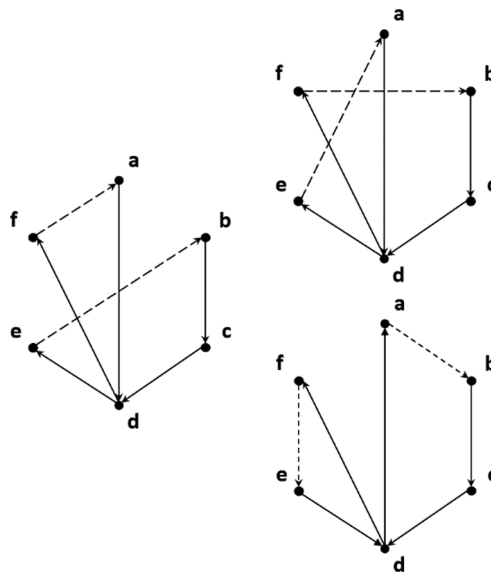
This algorithm provides the best solution to avoid the no-fly zone, and the best efficient path for decision-making between the start and end points.

### 2.2 Tabu Search (TS)

The tabu algorithm imitates human memory function, remembering past experiences to avoid redundant searches by building a taboo list which can remember past solutions to avoid identifying the optimal neighboring solution as the global optimal solution. As long as the initial solution and the taboo list are well established, the tabu search method can quickly achieve convergence and the obtained global best solution is relatively stable [12]. The tabu search method first establishes an initial solution, and then finds the optimal neighboring solution or the solution that meets the requirement to lift the ban as the

basis for moving. That is, it searches for the optimal solution in the vicinity of the current solution. The memory mechanism of the Tabu list is based on recording solutions that have already been searched to avoid repeated or meaningless searches. After searching all the neighboring areas, it chooses an optimal direction for movement. If a better solution exists, the current best solution will be updated until the termination condition is met [13]. We use 2-Opt, combined with the tabu search method. The 2-Opt nodal line exchange method was proposed by Lin [14] to change the route order. It was originally designed on the TSP problem, and it has been widely applied to major routing issues.

The improved tabu search method is combined with the 2-Opt exchange method which is used as a step to exchange sections between routes to improve the current solution, and the exchange method between routes is different from a single route. As shown in Fig. 2, the (f, a) and (e, b) road segments can be exchanged by using two different routes, namely (f, b) and (e, a), and (f, e) and (a, b). The direction of the latter changes in comparison to the former, so the 2-Opt exchange method will likely reverse the route direction.



**Figure 2:** The 2-opt swap method concept

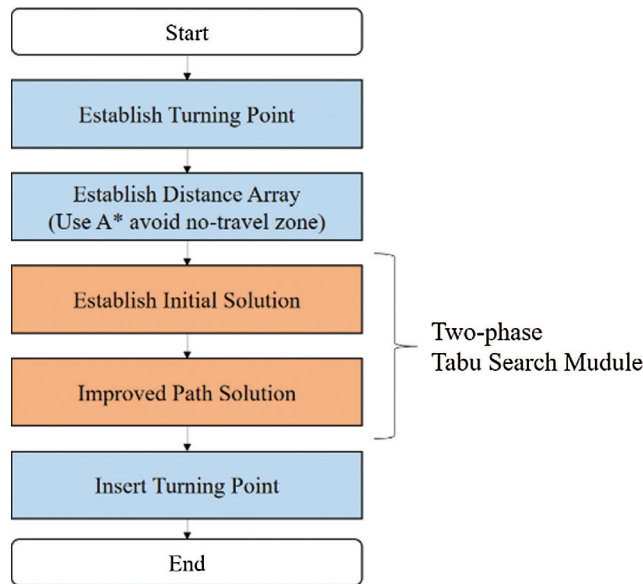
Therefore, the optimal solution has been improved by using 2-opt as a step-by-step exchange of different route sections. The stop search condition and initial solution are set, so that the improved TS can provide a multi-path optimal solution.

### 2.3 Workflow of Multi-Vehicle Path Programming

Based on tabu search, this study uses the A\* search method to establish the path planning algorithm shown in Fig. 3.

#### 2.3.1 Establishing the Turning Points and Distance Array (Using A\* to Avoid the No-fly zone)

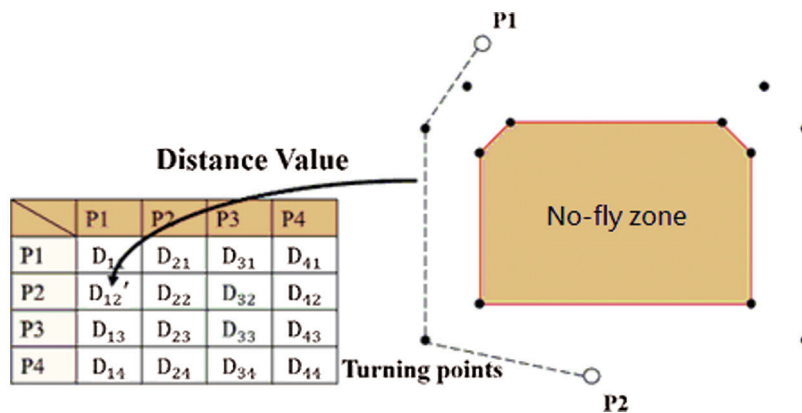
Traveling between targets may involve passing through the no-fly zone. Therefore, before calculating the target point, we first determine whether a straight path between the two target points intersects the no-fly zone. If so, we reroute the path to bypass the restricted space by a few meters using defined turning points which are expanded turning point to avoid the no-fly zone.



**Figure 3:** Main flow chart

The distances between all target points are stored as an array, which can serve as a quick reference for future determination of distances without recalculation.

Fig. 4 shows the position of the no-fly zone. The A\* search method determines whether a straight line between any two points passes through the no-fly zone (e.g., P1 and P2). The algorithm uses turning points to bypass the no-fly zone, creating a new path.



**Figure 4:** Schematic diagram of avoiding the no-fly zone

*2.3.2 Establishing the Initial Solution*

The establishment of the initial solution can be divided into three steps. The first step uses the matrix with the closest distance to establish a rough single path plan. The distance matrix  $c_{ij}$  has been established and the 0th line is infinite  $c_{ij} = \infty$ . Next, set the point with the minimum selected value in the  $c_{ij}$  matrix row and column as the first point, and set the  $c_{ij}$  value of the row to infinity. Repeat the search method to establish the second and third points in sequence until all points have been searched. The second step uses TS and 2-opt to optimize the single path. The third step divides the single path into

multiple sub-paths. The shorter each sub-path, the better. These sub-paths become the initial solution for the next stage.

### 2.3.3 Improved Path Solution

The key contribution of the present research is integrating the path into the optimal solution.

Here, according to the 2-opt rule, the node lines between and within the route are exchanged using a moving method, so that the solution gradually converges and gradually approaches the optimal solution. This minimizes the total distance, and ensures the distance allocated to each drone is balanced. As shown in the flowchart in Fig. 5, we first apply the improved tabu algorithm for route optimization until each group is improved, and then return to the improved tabu algorithm for inter-route improvement.

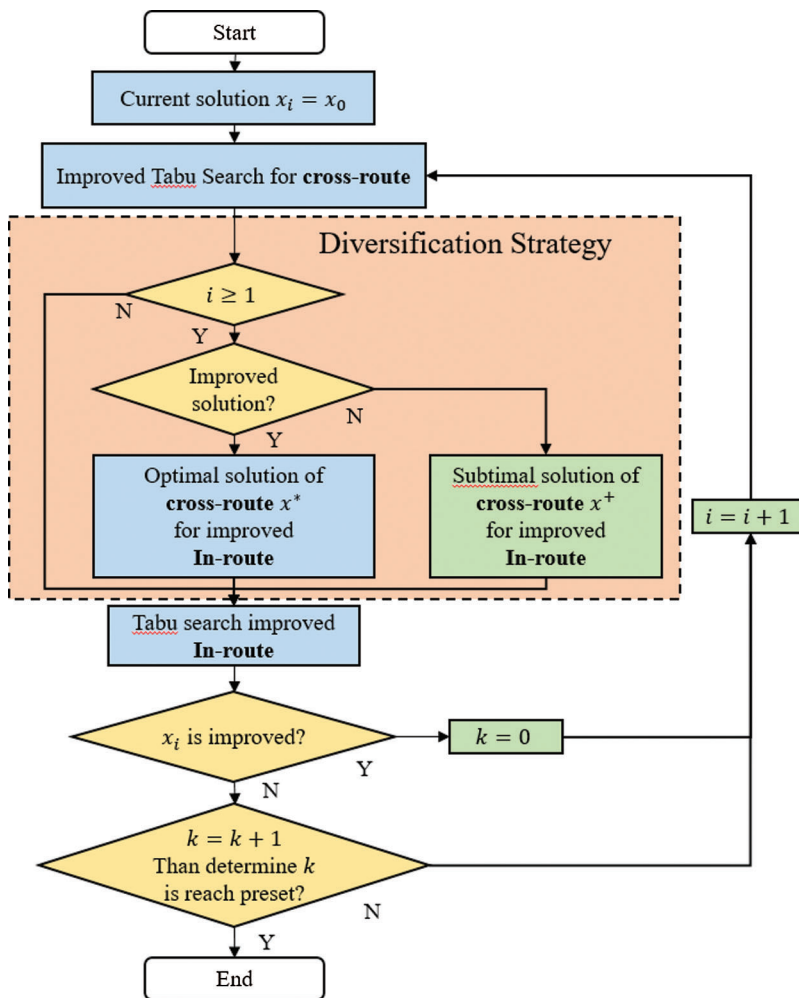


Figure 5: Flowchart of path improvement

Upon reaching the second generation, the inter-route improvement may result in an unimproved route solution identical as the route solution obtained in the first generation. If this route solution is substituted into the route improvement operation, the path solution will not be improved, and will fall into the best local solution. To avoid this, the mechanism will determine whether the path solution has been improved between the routes at this stage after the second generation. If no improvement is achieved, the next best

solution exchanged between the routes will be substituted for route improvement, making it jump of the best local solution to obtain the best global solution.

#### 2.3.4 Inserting the Turning Point

If no new turning point is generated after the initial path solution, the last turning point is inserted into the path to keep the drone away from the no-fly zone.

#### 2.3.5 Bench Test

The new algorithm developed for this research aims to solve the MTS. Three—cases Pr76, Pr299, and Pr439—were used as the instances to compare with the result from the modified genetic algorithm (MGA). In these tests, the waypoints are the total nodes that must be visited. Each vehicle must visit more than two targets, and maximum capacity stands for the maximum nodes that each vehicle may visit (owing to its onboard power limitation). In these tests, the Tabu list length was set to 30 and the stopping condition was that the optimal solution had not been updated and improved within the last 50 generations.

Tab. 1 shows the comparison results of the total traveling distance and CPU times. It obviously shows that our improved 2TS+2OPT is able to converge to a better solution that allows a logistic UAV swarm to operate in a more efficient way, no matter the distance or CPU time.

**Table 1:** The Comparison Result of 2TS+2OPT Algorithm in TSPLIB Instances

Name	Case A	Case B	Case C
Waypoints	76	299	439
Quantity of Vehicles	5	5	5
Maximum Capacity	20	70	100
MGA Distance (m)	178597	82106	173839
MGA CPU Time (s)	43	363	623
2TS+2OPT Distance (m)	153840	72813	141526
2TS+2OPT CPU Time (s)	11.3	190.5	455.4

### 3 4G-LTE Communication

For the purpose of drone delivery missions, a UAV swarm is not only capable of completing missions with greater efficiency but also of executing multiple tasks simultaneously via multiple vehicles within the swarm. Furthermore, empowered by 4G LTE communication technology, a UAV swarm is able to transmit data between the vehicles and the ground control station free from the long-standing constraint of the line-of-sight rule, breaking the limitation of radio range.

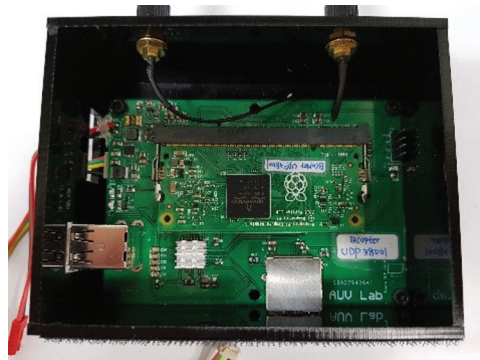
#### 3.1 4G Module

The UAV swarm in this thesis used a 4G LTE network to transmit the UAVs' data to the ground control station. Therefore, our team developed a 4G module that integrates a Raspberry Compute Module 3 (CM3) and a Huawei 4G LTE Cat4 Module into one PCB board (see Fig. 6) for this purpose.

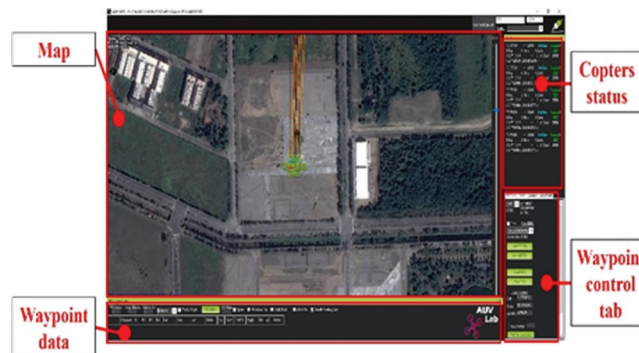
#### 3.2 Ground Control System

The ground control system (See Fig. 7) was modified from the open source Mission Planner, and the team added a new interface to monitor the whole UAV swarm.





**Figure 6:** 4G module



**Figure 7:** UAV swarm ground controller interface

The path programming interface is part of the waypoint control tab (See Fig. 7), with a text box for users to input the number of UAVs. Once the user inputs the number, the system collects the waypoint data input by users, and then produces new UAV paths based on the optimized Average Load Balance after re-programming.

After re-programming, the research team directs and controls each UAV with Auto-Guide mode. Each UAV is able to complete the task through the thread in Auto-Guide mode.

“Auto-Guide” is one of the navigation point interfaces. The system creates new threads for each copter, and the threads send the individual navigation points to each copter via 4G datalink. Once the UAV arrives at the last navigation point, the system switches the vehicle’s mode from “guided mode” to “RTL mode,” which drives the UAV to return back to the starting point. Controlled by the thread, each UAV can be managed by the system simultaneously. This function enables a UAV swarm to complete various tasks, such as multi-UAV surveillance of a large area or parcel delivery.

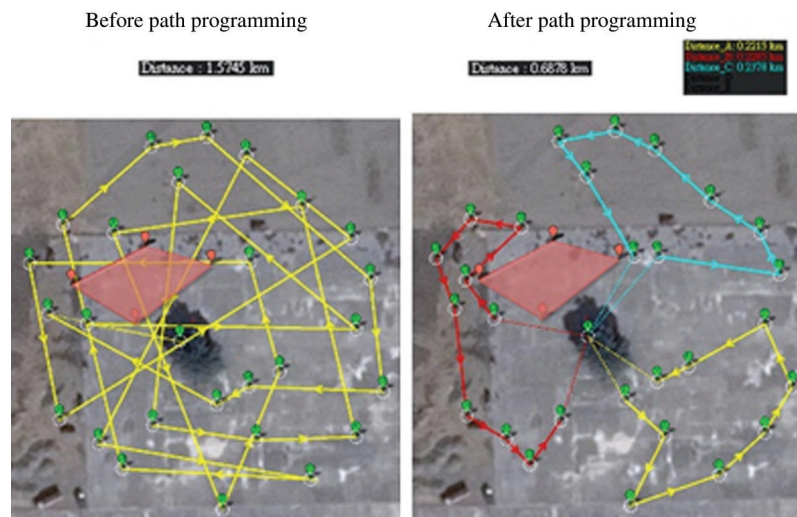
#### 4 Validation Results

Currently, most of the existing UAVs are not able to complete tasks that cover larger areas due to insufficient battery capacity. The research team tried to solve this issue by optimizing the path programming, because a poorly organized path program wastes battery power and leads to task failure. The research team used five copters to verify that the algorithm and control system work well. In the experiment, the system automatically made all copters take off and execute the mission, as seen in Fig. 8.



**Figure 8:** Function validation with five copters  
(Video Link: <https://youtu.be/JB9CG-X0R0Q>)

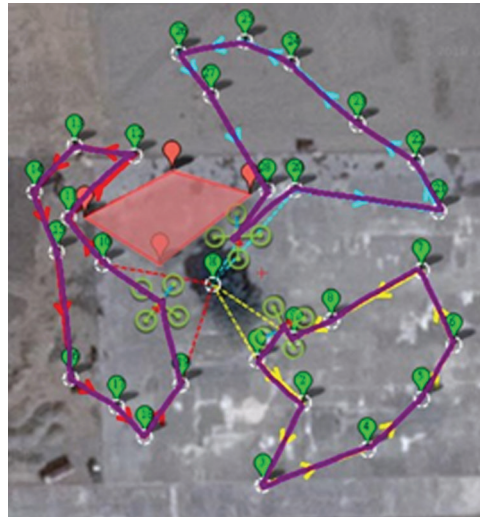
For the first test, three copters were tasked with carrying out a 26-point mission in an area that included a no-fly zone. As the left image shows in Fig. 9, the initial solution from our algorithm does cross the no-fly zone. Once this happened, the system automatically inserted a new waypoint (at the outer corners of the zone) and then put these newly generated paths back into routing optimization to keep the UAV outside the no-fly zone. The red-marked square in Fig. 9 is the no-fly zone given to the algorithm. On the right part of Fig. 9, the new path set has successfully avoided the no-fly zone through turning point guidance. The real flight trajectory of this experiment is shown in Fig. 10.



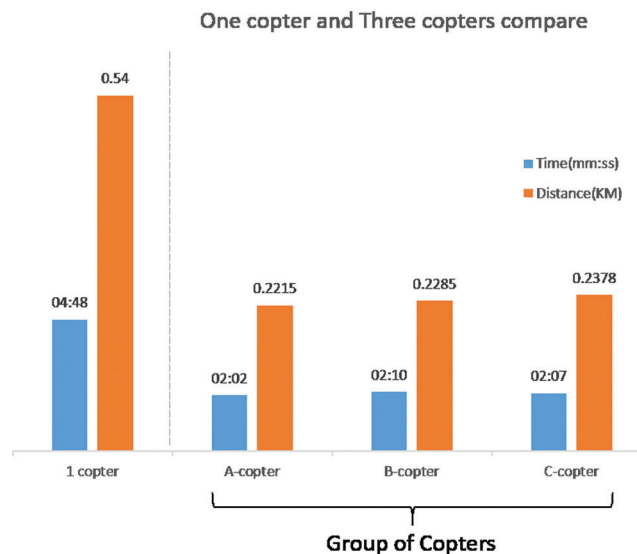
**Figure 9:** Paths of three copters with no-fly zone

From the data, it is clear that, compared with the 1-copter set, the 3-copter set ran a shorter distance (under onboard power limitation) to cover each waypoint and then complete the task in half of the time consumed by the 1-copter set. The mission distance and time consumed by the two different sets are compared in the bar chart of Fig. 11.

In a much larger mission area that even three copters are not able to cover, the research decided to make a UAV swarm up of five copters for larger coverage to verify its performance. As there is a giant tree at the middle of the experiment site, we marked an area around the tree as a no-fly zone to prevent any UAV crashes. The presence of the tree was an added benefit to this experiment because the team had a chance to verify if the system is able to correctly avoid the no-fly zone as programmed. Fig. 12 shows that the new paths successfully avoided the no-fly zone, proving that our algorithm functions well. The five copters' real flight trajectory with a no-fly zone is shown in Fig. 13.

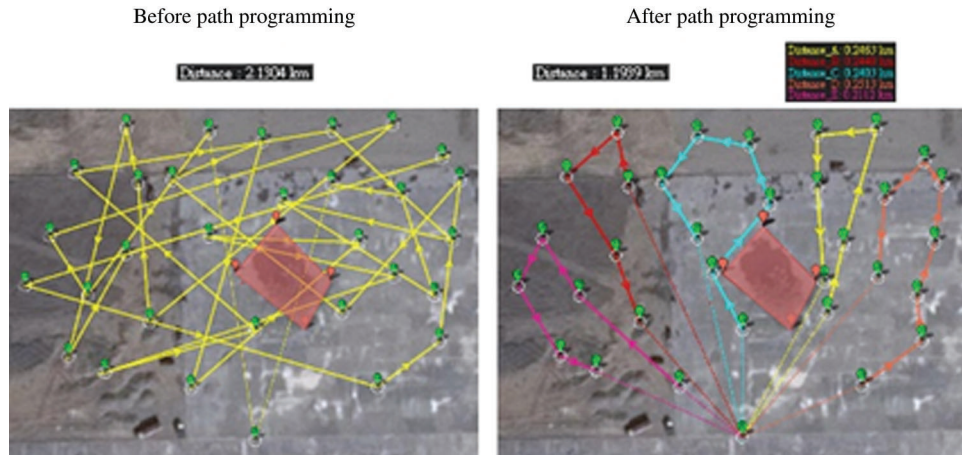


**Figure 10:** Real flight path of three copters with no-fly zone

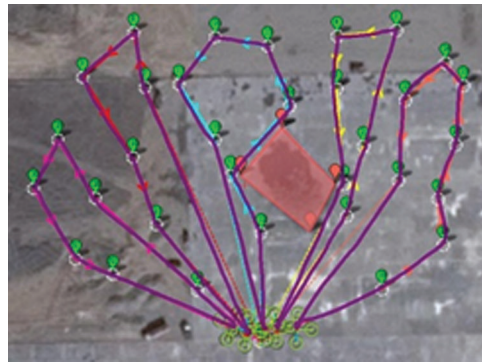


**Figure 11:** Comparison of 1-copter and 3-copter set performance

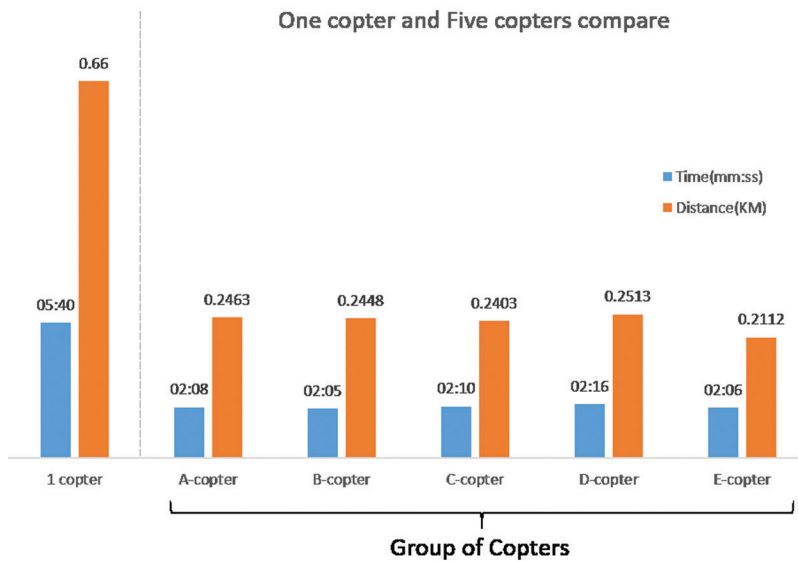
The flight distance of each of the five copters was shorter than the single copter's path, while the time consumed was reduced by 50%. Meanwhile, it is shown that our multiple-vehicle algorithm is able to dispatch a balanced work load to each vehicle of the swarm in a very efficient way. A comparison of time consumed and distance traveled by two different sets is presented in Fig. 14.



**Figure 12:** Path of five copters with no-fly zone



**Figure 13:** Real flight path of five copters with no-fly zone



**Figure 14:** Comparison of 1-copter and 5-copters set performance

## 5 Conclusions

In this research, a MAS was successfully combined with the A\* search and the two-phased Tabu designed for multi-vehicle path programming. The programmed paths were able to bypass the no-travel zone via the A\* search and reprogram the path with the two-phased TS. Any viable solution could be gradually converged by the improved TS, even if the initial solution exceeded the balance distance. Through the experiment, time and energy spent on task execution were reduced remarkably by the new design.

According to the experiment's results, it is certain that copters can follow the waypoints and path and finish the mission exactly as the algorithm indicates. That means the efficient and balanced flight-path UAV swarm system created by this research is feasible. This research could benefit the promising application of drone delivery.

**Acknowledgement:** We would like to thank all the lab. members for helping us to conduct the flight tests during function validation.

**Funding Statement:** Ministry of Science and Technology of R.O.C. supports this work under the contract number of MOST 109-2221-E-150-007, M. Lee, <https://www.most.gov.tw/?l=en>

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

- [1] M. Chuang, "Development of aerial robot swarm with 4G LTE communication system," M.S. thesis. National Formosa University, Yunlin, Taiwan, 2018.
- [2] M. Lee, B. Chen and Y. Lai, "A hybrid tabu search and 2-opt path programming for mission route planning of multiple robots under range limitations," *Electronics*, vol. 9, no. 3, pp. 534, 2020.
- [3] M. Lee, B. Chen and W. Lu, "Failure-Robot path complementation for robot swarm mission planning," *Applied Sciences*, vol. 9, no. 18, pp. 3756, 2019.
- [4] D. S. Maity and S. Goswami, "Multipath data transmission with minimization of congestion using ant colony optimization for MTSP and total queue length," *International Journal of Latest Research In Science and Technology*, vol. 2, no. 2, pp. 109–114, 2013.
- [5] S. Sariel-Talay, T. R. Balch and N. Erdogan, "Multiple traveling robot problem: A solution based on dynamic task selection and robust execution," *Institute of Electrical and Electronics Engineers/American Society of Mechanical Engineers Transactions on Mechatronics*, vol. 14, no. 2, pp. 198–206, 2009.
- [6] M. Yousefikhoshbakht, F. Didehvar and F. Rahmati, "Modification of the ant colony optimization for solving the multiple traveling salesman problem," *Romanian Journal of Information Science and Technology*, vol. 16, no. 1, pp. 65–80, 2013.
- [7] R. Necula, M. Breaban and M. Raschip, "Performance evaluation of ant colony system for the single-depot multiple traveling salesman problem," in *Proc. 10th International Conference on Hybrid Artificial Intelligence Systems*, Bilbao, Spain, pp. 257–268, 2015.
- [8] M. Xu, S. Li and J. Guo, "Optimization of multiple traveling salesman problem based on simulated annealing genetic algorithm," in *Proceedings MATEC Web of Conferences 100*, Zhengzhou, China, pp. 8, 2017.
- [9] J. F. Côté and J. Y. Potvin, "A Tabu search heuristic for the vehicle routing problem with private fleet and common carrier," *European Journal of Operational Research*, vol. 198, no. 2, pp. 464–469, 2009.
- [10] T. Liao and W. Huang, "A study of dynamic logistics based on two-phased method," *International Journal of Advanced Information Technology*, vol. 2, no. 1, pp. 76–94, 2008.
- [11] T. Bektas, "The multiple traveling salesman problem: An overview of formulations and solution procedures," *Omega*, vol. 34, no. 3, pp. 209–219, 2006.

- [12] B. Cai, *Path Programming for Autonomous Unmanned Vehicle System, Bachelor monograph*. Yunlin, Taiwan: National Formosa University, 2015.
- [13] F. Hung, *Vehicle Routing Problem of Integrated Supply Medical Materials in Strategic Alliance Hospitals: A Study of One Medical Center, Bachelor Monograph*. Yunlin, Taiwan: National Yunlin University of Science and Technology, 2004.
- [14] S. Lin, "Computer solutions of the traveling salesman problem," *Bell System Technical Journal*, vol. 44, no. 10, pp. 2245–2269, 1965.