INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Online ISSN 1841-9844, ISSN-L 1841-9836, Volume: 20, Issue: 4, Month: August, Year: 2025
Article Number: 6847, https://doi.org/10.15837/ijccc.2025.4.6847

CCC Publications

Multiple-UAV Path Planning in Obstacle Environments with Evolutionary Computation

Bui Trong Dung, Trinh Van Chien, Van Son Nguyen, Minh T. Hoang, Phuong Nhung Do, Hoang Anh Dang

Bui Trong Dung and Trinh Van Chien
School of Information and Communication Technology
Hanoi University of Science and Technology, Vietnam
No.1, Dai Co Viet Street, Hai Ba Trung, Hanoi, Vietnam
dung.bt207594@sis.hust.edu.vn and chientv@soict.hust.edu.vn

Van Son Nguyen*
Faculty of Electrical and Electronic Engineering
Hanoi Open University, Vietnam
B101 Nguyen Hien Street, Hai Ba Trung, Hanoi, Vietnam
*Corresponding author: sonnv@hou.edu.vn

Trong Minh Hoang
Telecommunication Faculty No1
Posts and Telecommunications Institute of Technology, Vietnam
Hanoi, Vietnam
hoangtrongminh@ptit.edu.vn

Phuong Nhung Do and Hoang Anh Dang
Faculty of Electrical and Electronic Engineering
Hanoi Open University, Vietnam
B101 Nguyen Hien Street, Hai Ba Trung, Hanoi, Vietnam
dpnhung@hou.edu.vn and anhdh.dttt@hou.edu.vn

Abstract

Broadcasting to various devices is a critical requirement when applying 5G and B5G networks. Unmanned aerial vehicles (UAVs) are expected to facilitate wireless 5G connection to remote areas with high data transmission speed. However, optimizing the flight paths of multiple UAVs is a significant challenge. The primary concerns include ensuring the UAVs avoid collisions with each other and with obstacles in the flying environment while maintaining efficient end-to-end routes. To address the challenge, this research proposes effective multiple-UAV route planning techniques. Two multiple-UAV approaches based on genetic algorithm (GA) and particle swarm optimization (PSO) are utilized in this paper. In complex scenarios, the proposed methods effectively determine the optimal UAV routes while satisfying various constraints. Simulation results indicate that GA is faster to output final paths but PSO has better flight paths.

Keywords: unmanned aerial vehicles, 5g networks, route planning, genetic algorithm, particle swarm optimization.

1 Introduction

The rapid development of wireless communication technology [21, 41] has made the fifth generation (5G) and beyond 5G (B5G) networks superior to previous generations [37]. Unique features of 5G such as high data transmission, low latency, and real-time connection to multiple devices [4, 8, 28, 31, 35] facilitate high performance and reliable applications [7, 43]. While 4G/LTE has improved significantly in transmission speed and connection quality [17], 5G/B5G networks still offer outstanding enhancements with larger bandwidth [18]. These features enhance user experience [2] and create new opportunities for the telecommunication industry, particularly the IoT industry [39, 45]. 5G technology facilitates smart homes, a common application of the IoT, by enabling high-speed communication. The network enables homeowners to remotely access and operate their devices, including lights, thermostats, and security systems, with little latency. The low latency connectivity guarantees that commands are performed almost immediately, making controlling home environments quicker and more efficient. In smart city applications, 5G is the foundation for autonomous traffic management, real-time weather updates, energy-saving technologies, and smart lighting. It also facilitates effective water resource management, crowd control, and emergency response systems. Hence, smart city users benefit from the high-speed and reliable connectivity that 5G provides. Unmanned aerial vehicles (UAVs) are important components of modern applications in 5G/B5G networks [34]. In remote areas, UAVs can provide temporary connection [5, 27], perform security surveillance operations [24], and support emergencies [13, 14]. Additionally, they can act as aerial base stations to enhance 5G wireless networks, therefore, the quality of service is improved and their coverage areas are broadened [23, 25]. A way to utilize UAVs is by planning UAV routes and optimizing the UAV path-finding process. It offers numerous benefits, including minimizing travel time, saving energy, and avoiding danger zones or obstacles [1, 3, 20, 32, 33, 44, 46, 49, 54, 56]. UAVs operate in a three-dimensional space with dynamic elements and their planning process is a complex challenge [6, 36]. To pilot through the dimensions and avoid possible collisions successfully, a path-planning algorithm has to consider several factors. One of the challenges that have to be acknowledged when planning UAV flight routes is the environmental complexity [1, 9, 50].

Although many path-planning techniques exist, this research focuses on multi-UAV approaches and their suitability for complex, obstacle-rich environments. While effective in smaller problems, traditional shortest path algorithms, including the Dijkstra algorithm, struggle with scale and complexity in large-scale applications [42]. We explore alternative methods such as PSO (Particle Swarm Optimization) and GA (Genetic Algorithm), which can manage complicated search spaces [10, 12, 16, 26, 30, 47, 55]. Additionally, these techniques can simultaneously handle the multiple-UAV issue [15, 52]. Particle Swarm Optimization (PSO) offers its simplicity [22, 40, 48], fast convergence [10], and ability to handle complex search spaces [22] which make it well-suited for UAV path planning. In addition to PSO and GA, there are also other options. While sampling-based approaches can handle diverse information from previous planning iterations, they often get stuck in local minima and cause long exploration time [38, 51]. In dynamic environments, reinforcement learning shows promises [11] but formulating reward functions for situations can be challenging [29]. Therefore, this research explores the potential of utilizing PSO and GA in the multiple-UAV scenario. We simulate to evaluate their performance and decide which method provides a more effective solution for UAV route planning. While several studies have focused on using Particle Swarm Optimization (PSO) in multi-UAV path planning in obstacle environments [19], this paper introduces the Genetic Algorithm (GA) as a novel approach. Additionally, we provide a detailed computational complexity comparison between GA and PSO, which has not been seen in solving the multi-UAV path planning problem in complex environments using hybrid PSO.

This paper can be utilized in various critical scenarios, especially in disaster response and emergencies. In disaster areas like fire in forests with high tree density, UAVs are ideal for retrieving emergency information from remote regions [53]. These environments have significant challenges, such as navigating through tree canopy, encountering weather conditions, and avoiding natural obstacles like waterfalls. Additionally, UAVs sometimes have to navigate around restricted flight regions, protected areas, and signal interference. Optimized path planning methods proposed in this research help to ensure that UAVs can be directed efficiently and allow crucial messages to be transmitted quickly

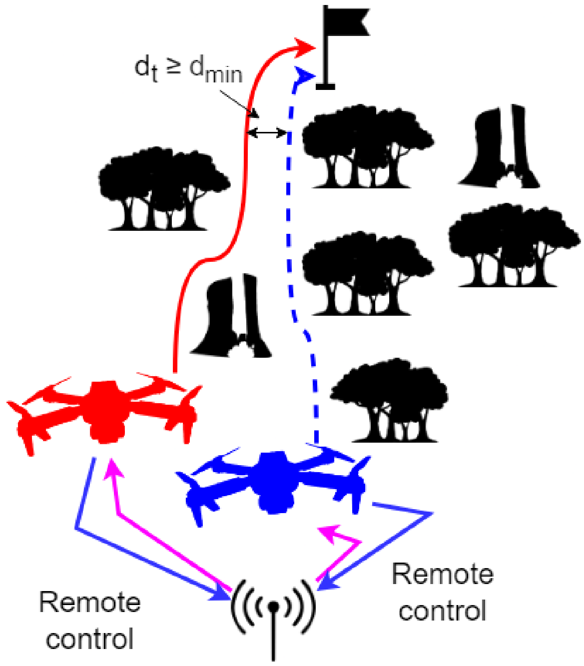

Figure 1: The considered system model

to the information collection base station or directly to rescue teams.

2 System Model & Optimization Problem

As shown in Figure 1, the map illustrates the starting points of both UAVs, their destination, and the locations of obstacles. The established routes must avoid obstacles and prevent collisions with another UAV. Moreover, the total travel time needs to be as short as possible. Additionally, if both UAVs need to pass through the same point on the map, only one can cross at a time, as they must maintain a safe distance from each other at all times.

2.1 UAV Path Representation and Segment Calculation

The UAV's path is represented as a sequence of redirect points, each point is defined by its coordinates (x_i, y_i) and corresponding UAV speed v_i. The segment between (x_i, y_i, v_i) and $(x_{i+1}, y_{i+1}, v_{i+1})$ redirect points is defined as the path segment S_i. To determine the coordinates of a UAV at any given time t along a path segment S_i, we use the following formulas

$$x(t) = \frac{(t - t_i) \cdot (x_{i+1} - x_i)}{t_{i+1} - t_i} + x_i, \tag{1}$$

$$y(t) = \frac{(t - t_i) \cdot (y_{i+1} - y_i)}{t_{i+1} - t_i} + y_i, \tag{2}$$

where t_i represents the time when the UAV reaches point (x_i, y_i, v_i), where $t_i \leq t < t_{i+1}$.

2.2 Formulated Optimization Problem

The cost function evaluates UAV flight paths based on three key factors: time cost, danger zone cost, and collision cost. The time cost for UAV j, denoted as C_{time_j}, represents the time it takes for UAV j to complete its task. The total time cost for both UAVs can be calculated as

$$C_{time} = \sum_j C_{time\_j}. \tag{3}$$

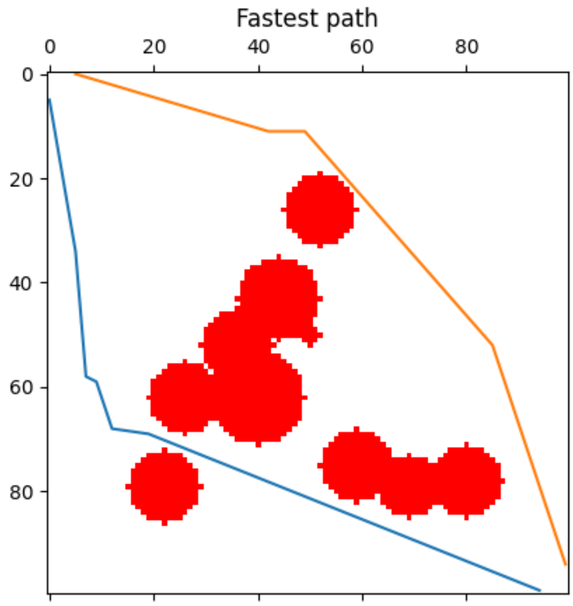

Figure 2: Genetic algorithm output paths

The danger zone cost is computed by summing the costs incurred during each time interval when a UAV is inside a dangerous zone. The total danger zone cost can be expressed as

$$C_{\text{danger\_zones}} = \sum_t C_{\text{danger\_zones\_t}}. \tag{4}$$

Here, $C_{\text{danger\_zones\_t}}$ stands for danger zone cost at time interval t. The cost at each time interval is determined by distance between the UAV and center of a danger zone, its speed, and a penalty coefficient $k1$. Mathematically, the danger zone cost for the UAV at time interval t is expressed as

$$C_{\text{danger\_zones\_t}} = \begin{cases} 0 & \text{if } d_t \geq d_{\min}, \\ \frac{v_t \times k1}{d_t} & \text{if } d_t < d_{\min}, \end{cases} \tag{5}$$

where d_t represents the distance between the UAV and the danger zone center at time t , v_t denotes its speed at time t, and $k1$ is the penalty coefficient. The collision cost is incurred when the distance between two UAVs falls below the distance threshold for detecting a UAV collision. The total collision cost can be expressed as

$$C_{\text{collision}} = \sum_t C_{\text{collision\_t}}. \tag{6}$$

Here, $C_{\text{collision\_t}}$ stands for collision cost at time interval t. The cost at each time interval is determined by the distance between the two UAVs, velocities, relative velocity, and penalty coefficient $k2$. The collision cost for both UAVs at time intervals t is expressed as

$$C_{\text{collision\_t}} = \begin{cases} 0 & \text{if } d_t \geq d_{\min}, \\ \dfrac{(v_1 + v_2 + \delta v)k2}{d_t} & \text{if } d_t < d_{\min}. \end{cases} \tag{7}$$

Here, at each interval time t, d_t represents the distance between two UAVs, v_1 and v_2 stand for the velocities of UAV 1 and UAV 2, respectively, δv is the relative velocity between those UAVs and $k2$ is the penalty coefficient for UAV collisions.

We now formulate the constraints and objective function for our optimization problem. Our objective is to minimize the total cost, denoted as C, which includes three individual costs, which are

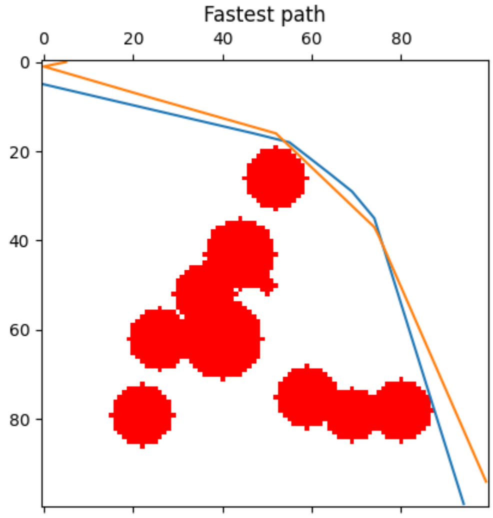

Figure 3: Particle swarm optimization output paths

time cost, danger zone cost, and collision cost. The objective function is expressed as

$$\text{minimize} \quad C = C_{time} + C_{danger\ zones} + C_{collision} \tag{8}$$

$$\text{subject to} \quad (x_1, y_1) = (x_{start}, y_{start}) \tag{9}$$

$$(x_n, y_n) = (x_{end}, y_{end}) \tag{10}$$

$$0 \leq x_i \leq x_{\max} \tag{11}$$

$$0 \leq y_i \leq y_{\max} \tag{12}$$

$$v_{\min} \leq v_i \leq v_{\max}. \tag{13}$$

In addition to minimizing the total cost, the constraints are designed to ensure that the UAV will fly along the expected flight path, from the source to the destination location which is shown in (9) and (10), through redirect points, in permitted areas, which is demonstrated in (11) and (12). At redirect points, they can rotate when flying, and those intermediate locations' coordinates can be updated by algorithms. To ensure flight safety but still fly effectively, the UAVs' velocity has to be limited within a speed range in (13).The sensitivity to weather conditions such as wind, rain or other environmental factors has significant impacts on specific UAVs due to the configurations. To keep a general framework, we have not considered these environmental factors in this work.

3 Proposed Solution

3.1 Methodological Approach

The Particle Swarm Optimization (PSO) approach is outlined in 1. Initially, a swarm of particles is generated randomly, where each particle represents a potential pair of end-to-end UAV paths. These particles are evaluated using a cost function defined in (8), the function can approximate the quality of each pair by considering travel time, penalties for entering danger zones, and potential collisions. The approximation helps guide the optimization process. To improve the quality of these paths, each particle adjusts its velocity and coordinates of rotation points based on its own best position (*Pbest*) and the best position of the entire swarm (*Gbest*). Particles use inertia weight to accelerate towards *Pbest* and *Gbest*. This dynamic adjustment allows particles to explore the search space and converge towards optimal paths effectively. It minimizes travel time while avoiding danger zones and UAV collisions. Throughout the optimization process, the algorithm continuously tracks the best positions of each particle (*Pbest*) and the best position founded by the entire swarm (*Gbest*). As better solutions

Input : number of patience iteration as n_patience_iteration; inertia weight range as omega_min_max
Procedure: `Find final path by using PSO approach`
begin
    path_pairs, gbest_pair ← initialize();
    pbest_pairs ← path_pairs;
    while *no convergence* **do**
        if *cost(gbest_pair) stable for n_patience_iteration* **then**
            path_pairs.partial_reset();
        end
        foreach *p ∈ path_pairs.len* **do**
            inertia_weight.update(omega_min_max);
            p_delta ← pbest_pairs[p] - path_pairs[p];
            g_delta ← gbest_pair - path_pairs[p];
            rand_coef = rand.uniform(0,1);
            path_pairs[p].update(p_delta, g_delta, inertia_weight, rand_coef);
            pbest_pair.compare_update(path_pairs[p]);
            gbest_pair.compare_update(path_pairs[p]);
        end
    end
end
Output: Final path for the problem

Algorithm 1: PSO Algorithm

are discovered, these two values are updated. To prevent the algorithm from getting stuck in local minima, when the global best cost remains stable for a predefined patience threshold, a mechanism is applied to reset a subset of the path solutions partially. This introduces new diversity into the population, helping the algorithm explore alternative solutions and improve convergence efficiency. After several iterations that lack significant improvement, the stopping condition is considered satisfied, then the algorithm outputs the best pair of paths, representing the optimal flight paths for the two UAVs. Significant improvement is implicitly defined by multiple factors. Firstly, whether the average fitness has not plateaued over a recent window. Secondly, the average rate of improvement over a period of iterations is above an adaptive threshold. The algorithm is considered to have "significant improvement" when these conditions are met, ensuring a balance between finding a good solution and avoiding excessive computation. If the algorithm continues to provide a significant improvement over a long period and has not yet exceeded the maximum number of iterations, it continues running to refine the solution further. Besides adjusting parameters and partially resetting a subset of solutions to further guarantee convergence within a short time frame, the algorithm runs for at least 20 percent of the maximum allowed iterations before early termination is considered. This mechanism helps to prevent premature stopping and ensures the search process thoroughly explores the solution space before converging. These routes can avoid danger zones, prevent UAV collisions, and minimize travel time.

Besides the PSO approach, we propose a Genetic Algorithm (GA) method to solve this problem, outlined in 2. In the initial phase, we generate individuals, each representing a potential pair of paths for two UAVs. These paths are initialized randomly, meaning that each redirect point for each UAV is selected randomly within the defined map boundaries and speed limits. This randomness ensures that the initial population is diverse. Once the population is initialized, we evaluate the fitness of each individual. The fitness function is designed to check how well each path satisfies the UAV routing problem, it calculates the total travel time for two UAVs, adds penalties for entering danger zones, and includes penalties for potential collisions between two UAVs, directly relating to the cost function defined in (8). The selection phase is critical, for our UAV path optimization problem, we use the tournament selection method. First, it chooses randomly a subset of individuals from the population. From this subset, the individual with the highest fitness is selected as a parent. This

```
Input    : crossover_rate as crossover rate; mutation_rate as mutation rate; n_redirect_point
           as number of redirect points
Procedure: Find final path by using GA approach
begin
    path_pairs, global_best_pair ← initialize();
    while no convergence do
        offspring ← empty array;
        foreach individual ∈ path_pairs do
            parents ← tournament(path_pairs);
            if get_random_rate() < crossover_rate then
                cross_index ← random(1,n_rotate_point);
                child ← crossover(parents,cross_index);
            end
            else
                child ← random_select(parents);
            end
            foreach redirect_points ∈ child do
                if get_random_rate() < mutation_rate then
                    redirect_points.mutate_coordinate();
                    redirect_points.mutate_velocity();
                end
            end
            offspring.append(child);
        end
        path_pairs ← elitism(path_pairs, offspring);
        local_best_cost ← path_pairs.min_fitness_value();
        if local_best_cost < global_best_cost then
            global_best_pair ← path_pairs.best_pair();
            global_best_cost ← local_best_cost;
        end
    end
end
Output: Final path for the problem
```

Algorithm 2: Genetic Algorithm Approach

process is repeated to select another parent. After selecting parents, the crossover phase generates new offspring. For the UAV path-finding problem, we use a single-point crossover technique. Here, a random crossover point along the sequence of redirect points is selected. The segments before and after this point are exchanged between the two parents to create two new offspring. This technique ensures that the offspring inherit from both parents. The mutation is applied to some offspring with a certain probability. In the context of UAV routing, mutation updates the coordinates of a redirect point or the UAV speed at certain points. This randomness helps maintain diversity in the population, preventing premature convergence to local optima solutions. To ensure continual improvement in the quality of output solutions, we have implemented an elitism strategy for the UAV path optimization problem. It preserves the top-performing individuals from the current generation. Specifically, a part of individuals with the highest fitness scores, representing the most potential UAV paths are directly carried over to the next generation. By keeping these elite individuals, we maintain a baseline level of quality in the population. The remaining slots in the population can be filled with next-generation offspring by crossover and mutation operations. The stopping criteria are designed to ensure convergence while preventing excessive computation, meeting the real-time demands of the application. First, the average fitness has plateaued over a recent window, indicating minimal improvement. Second, the average rate of improvement over a period of iterations is under an adaptive threshold. Finally, the number of current iterations exceeds the maximum number of allowed iterations. If any of these criteria are met, the stopping condition is satisfied, ensuring a balance between finding an optimal solution and computational efficiency.[1]

3.2 Computational Complexity Analysis

The fitness evaluation is crucial because it directly impacts the output of both algorithms in finding optimal solutions. For each segment, we calculate its slope and its intercept, which has a complexity of $\mathcal{O}(1)$. Besides, the presence or absence of collisions between the UAV flight path and obstacles must also be considered. Because we have to check along a flight segment, that makes the complexity of inspection of whether a UAV collided with an obstacle is $\mathcal{O}(\sqrt{2} \times map\_dimension)$, where map_dimension is denoted for the length of the square side of the flight area. In addition, we have to evaluate n_redirect_point redirect points of each UAV, which makes the computation complexity for 2 UAVs equals $\mathcal{O}(\sqrt{2} \times map\_dimension \times n\_redirect\_point \times 2) = \mathcal{O}(map\_dimension \times n\_redirect\_point)$. On the other hand, checking if two UAVs collide with each other has a huge impact on the fitness function. We take the approach of checking every second for the entire flight whether two UAVs will collide with each other, which leads to its complexity, which is $\mathcal{O}(\sqrt{2} \times map\_dimension/v\_min \times n\_redirect\_point) = \mathcal{O}(map\_dimension/v\_min \times n\_redirect\_point)$, equivalent to the longest time a UAV can fly. Here, the slowest flight speed is v_min. Finally, for each redirect point, the cost function has to consider end-to-end travel time, with the complexity of $\mathcal{O}(n\_redirect\_point \times 2)$. Because checking obstacle collision has the highest times complexity, and because three evaluating steps, including UAV collision checking, object collision checking, and time-consuming calculation, are used in order, the fitness evaluation cost can be described as

$$\mathcal{O}(map\_dimension \times n\_redirect\_point). \tag{14}$$

To approximate the complexity of the PSO approach, in 1, we examine each component and determine its complexity. In the beginning, we consider the population initialization process of n_particle particles, each particle has 2 UAV, each UAV has n_redirect_point redirect points, and each point has 3 attributes, which are horizontal coordinates, vertical coordinates, and velocity values. Therefore, its complexity is

$$\mathcal{O}(n\_particle \times n\_redirect\_point). \tag{15}$$

The primary steps involved in each iteration of the PSO algorithm, which runs for a maximum of n_max_iter iterations, include updating the speed, coordinates corresponding to redirect points, and the fitness value of the examining particle. Since these updates are performed on redirect points, the total complexity for these updates in each iteration is $\mathcal{O}(n\_particle \times n\_redirect\_point)$. After updating velocities and coordinates, each particle fitness value is evaluated, and its complexity is defined in (14), therefore, evaluating all of the particles has a complexity of $\mathcal{O}(n\_particle \times map\_dimension \times n\_redirect\_point)$. Since the complexity of each iteration is dominated by the fitness calculation step, the complexity of one iteration in the PSO approach can be formulated as follows

$$\mathcal{O}(n\_particle \times map\_dimension \times n\_redirect\_point). \tag{16}$$

In the genetic algorithm, i.e., Algorithm 2, the initialization step has different parameters, compared to the PSO algorithm, from n_particle of particles to population_size of individuals, but the mechanism remains the same, therefore, its complexity is $\mathcal{O}(population\_size \times n\_redirect\_point)$.

Next, the child is mutated, where the complexity is $\mathcal{O}(n\_redirect\_point)$ because the mutation function iterates over each redirect point. Because of that, parsing through all individuals costs total $\mathcal{O}(population\_size \times n\_redirect\_point)$. After generating the offspring through crossover and mutation, the fitness values of the offspring are evaluated using function (8). Since it iterating through each individual in the offspring and current population, the complexity of evaluating the fitness values step is $\mathcal{O}(population\_size \times n\_redirect\_point \times map\_dimension)$. Finally, the elitism strategy is

[1]Previous works have demonstrated that genetic algorithm performs better than ant colony optimization in many practical applications. Meanwhile, particle swarm optimization converges fast to a fixed point solution. Hence, we have selected the two algorithms for demonstrating the system performance. Since deep reinforcement learning is aligned with data-driven approaches, it is not included for comparison.

applied to combine the best individuals, so it forms the next generation, the elitism process has the complexity of $\mathcal{O}(population\_size \times population\_size)$. Overall, the complexity of calculating fitness value dominates other complexity in each iteration, hence, the complexity of each iteration in the genetic algorithm is

$$\mathcal{O}(population\_size \times n\_redirect\_point \times map\_dimension). \tag{17}$$

4 Numerical Results

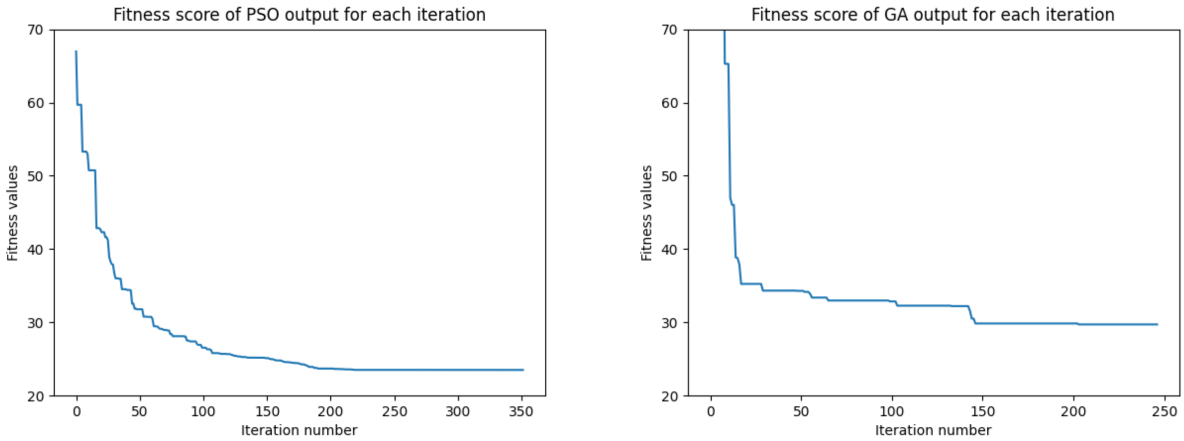

Figure 4: Fitness score of two algorithm output paths for each iteration

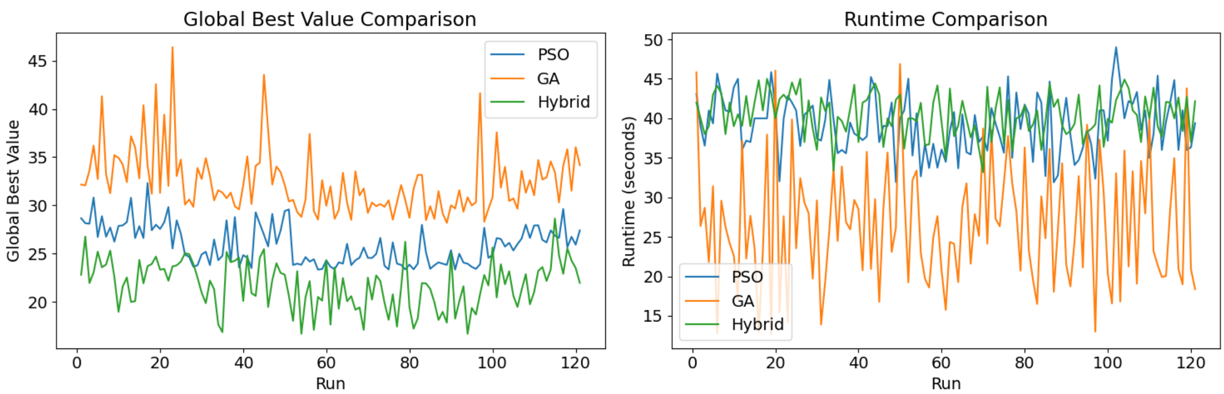

Figure 5: Comparison of valuation score and runtime for each algorithm output pair of paths

4.1 Parameter Settings

In our research on efficient multiple-UAV route planning in complex environments, we define a set of parameter settings to ensure the effectiveness and reliability of our algorithms. First, we set the distance threshold for detecting UAV collisions to 4 [m], while restricting the maximum velocity for UAVs to 10 [mps]. The map is designed as a square grid with dimensions of 100 [m], and we randomly place 10 obstacles within this space. Next, we designate 5 redirect points along the UAV route. Considering the start and destination positions for 2 UAVs, they start their journey from the coordinates (5,0) and (0,5), respectively, and reach the goal located at (100,95) and (95,100), respectively.

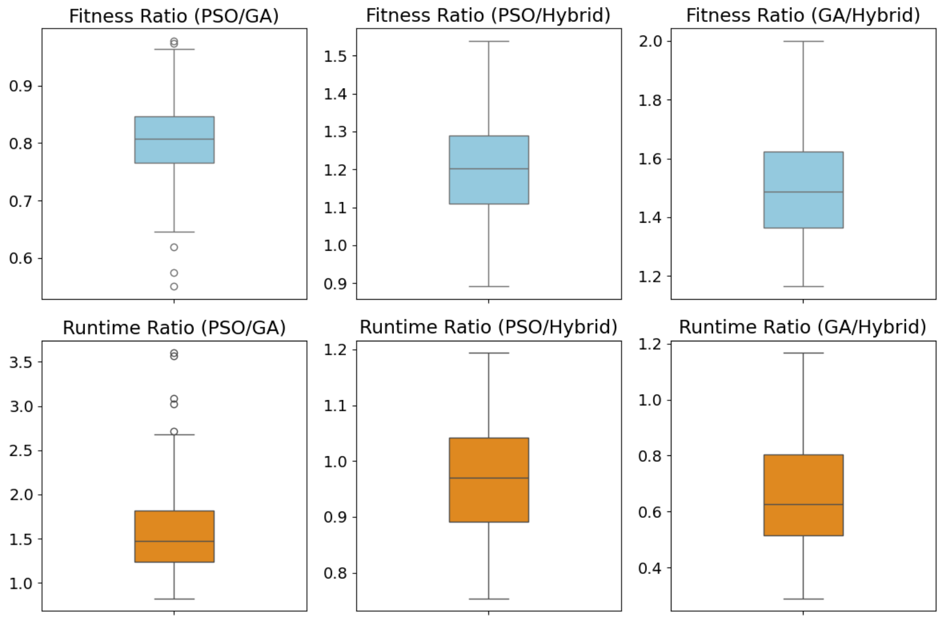

Figure 6: The box plot of benchmarks

To fine-tune our algorithms, we used Bayesian Hyperparameter Optimization, a strategic choice based on the continuous nature of our hyperparameters. This method facilitates efficient exploration of the hyperparameter space. For the PSO algorithm, this involved adjusting the number of particles, the cognitive and social parameters, and the inertia weights. While for the GA algorithm, we focused on the population size, crossover rate, and mutation rate. The search ranges for the number of particles and population size ranged from 20 to 200. In addition, the search space for the cognitive and social parameters from 0.1 to 10, the inertia weights from 0.1 to 2, the crossover rate from 0.5 to 1.0 and the mutation rate from 0.05 to 0.3. Through this optimization process, we determined suitable parameters for our algorithms. For the PSO algorithm, we configure the number of particles used in the Particle Swarm Optimization algorithm to 30 and execute with cognitive and social parameters set to 2.0 and 2.0, respectively. Additionally, we limit the inertia weights to between 0.3 and 0.8. For the GA algorithm, we initialize a population of 20 individuals and define a crossover rate of 0.9 and a mutation rate of 0.1. The algorithm runtime and end-to-end travel time are recorded to compare the performance of the two approaches.

4.2 Results and Discussions

Figure 4 shows the convergence behavior of PSO and GA, in one run among 120 experiment runs. Both algorithms demonstrate a rapid decrease in fitness scores, with elbow point around the 50^{th} iteration for PSO and the 15^{th} iteration for GA. Moreover, the first graph in Figure 5 and the first three box plots in Figure 6 compare the end-to-end travel times of the UAVs for GA, PSO and Hybrid PSO-GA. The results indicate that the PSO algorithm finds better routes with travel times ranging between 24 to 28 [s], while GA produces significantly longer travel times, between 29 to 34 [s]. The Hybrid PSO-GA approach, which initializes paths using GA and refines them with PSO, achieves the shortest travel times, typically ranging from 18 to 23 [s]. In addition, there is a run of the genetic algorithm that has more than 46 [s] in travel time, highlighting its inconsistency in path optimization. Furthermore, in the second graph of Figure 5 and the last three box plots in Figure 6 illustrate the runtime comparison for each algorithm. The GA method consistently shows shortest runtimes, completes the task within 20 [s] to 35 [s] to find optimal routes. While the PSO algorithm requires 35 [s] to more than 40 [s], which is 1.5× longer than GA algorithm as shown in the box plot. The Hybrid PSO-GA approach does not fully inherit GA's speed advantage, its runtime performance is closer to that of PSO, typically ranging from 39 to 42 [s]. The execution time is influenced primarily by the number of drones rather than the number of obstacles or unpredictable paths. The computational complexity is detailed in 16 and 17, showing a linear impact as the number of drones increases. Faster computation can be achieved by increasing server resources to meet real-time requirements. Overall, the results demonstrate that the PSO algorithm tends to have better output but consistently takes more time to execute.

5 Conclusion

In this paper, the study effectively employs Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) methodologies to enhance UAV routing within obstacle environments, proficiently tackling significant challenges such as the reduction of travel duration, circumventing hazardous zones, and averting collisions. By facilitating the proficient autonomous management and regulation of UAVs within 5G networks, these methodologies exhibit considerable potential for application in real-world scenarios such as disaster response, where effective UAV navigation is imperative. Moreover, the comprehensive examination of the computational complexity associated with both PSO and GA yields insightful perspectives regarding their efficacy and applicability in practical contexts, emphasizing their relevance for sophisticated UAVs.

Funding

This study was funded by project code MHN2023-01.01 from the Ha Noi Open University

Conflict of interest

The authors declare no conflict of interest.

References

[1] Aggarwal, Shubhani, Kumar, Neeraj (2020). Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges, *Computer communications*, 149, 270–299, 2020.

[2] Agiwal, Mamta, Kwon, Hyeyeon, Park, Seungkeun, Jin, Hu (2021). A survey on 4G-5G dual connectivity: Road to 5G implementation, *Ieee Access*, 9, 16193–16210, 2021.

[3] Ahmed, Shakil, Chowdhury, Mostafa Zaman, Jang, Yeong Min (2020). Energy-efficient UAV-to-user scheduling to maximize throughput in wireless networks, *IEEE Access*, 8, 21215–21225, 2020.

[4] Ahmed, Ramsha, Chen, Yueyun, Hassan, Bilal (2021). Deep learning-driven opportunistic spectrum access (OSA) framework for cognitive 5G and beyond 5G (B5G) networks, *Ad Hoc Networks*, 123, 102632, 2021.

[5] Bajracharya, Rojeena, Shrestha, Rakesh, Kim, Shiho, Jung, Haejoon (2022). 6G NR-U based wireless infrastructure UAV: Standardization, opportunities, challenges and future scopes, *IEEE Access*, 10, 30536–30555, 2022.

[6] Bayerlein, Harald, Theile, Mirco, Caccamo, Marco, Gesbert, David (2021). Multi-UAV path planning for wireless data harvesting with deep reinforcement learning, *IEEE Open Journal of the Communications Society*, 2, 1171–1187, 2021.

[7] Benzaïd, Chafika and Taleb, Tarik and Farooqi, Muhammad Zubair (2021). Trust in 5G and beyond networks, *IEEE Network*, 35, 212–222, 2021.

[8] Budhiraja, Ishan, Kumar, Neeraj, Tyagi, Sudhanshu, Tanwar, Sudeep, Han, Zhu, Piran, Md Jalil, Suh, Doug Young (2021). A systematic review on NOMA variants for 5G and beyond, *IEEE Access*, 9, 85573–85644, 2021.

[9] Chai, Xuzhao, Zheng, Zhishuai, Xiao, Junming, Yan, Li, Qu, Boyang, Wen, Pengwei, Wang, Haoyu, Zhou, You, Sun, Hang (2022). Multi-strategy fusion differential evolution algorithm for UAV path planning in complex environment, *Aerospace Science and Technology*, 121, 107287, 2022.

[10] Cheng, Shixin, Zhan, Hao, Yao, Huiqin, Fan, Huayu, Liu, Yan (2021). Large-scale many-objective particle swarm optimizer with fast convergence based on Alpha-stable mutation and Logistic function, *Applied Soft Computing*, 99, 106947, 2021.

[11] Cui, Zhengyang and Wang, Yong (2021). UAV path planning based on multi-layer reinforcement learning technique, *Ieee Access*, 9, 59486–59497, 2021.

[12] D'Angelo, Gianni, Palmieri, Francesco (2021). GGA: A modified genetic algorithm with gradient-based local search for solving constrained optimization problems, *Information Sciences*, 547, 136–162, 2021.

[13] Do-Duy, Tan, Nguyen, Long D, Duong, Trung Q, Khosravirad, Saeed R, Claussen, Holger (2021). Joint optimisation of real-time deployment and resource allocation for UAV-aided disaster emergency communications, *IEEE Journal on Selected Areas in Communications*, 39, 3411–3424, 2021.

[14] Feng, Wanmei, Tang, Jie, Yu, Yu, Song, Jingru, Zhao, Nan, Chen, Gaojie, Wong, Kai-Kit, Chambers, Jonathon (2020). UAV-enabled SWIPT in IoT networks for emergency communications, *IEEE Wireless Communications*, 27, 140–147, 2020.

[15] Sahingoz, Ozgur Koray (2013). Flyable path planning for a multi-UAV system with Genetic Algorithms and Bezier curves, *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, 41–48, 2013.

[16] Gu, Zi-Min, Wang, Gai-Ge (2020). Improving NSGA-III algorithms with information feedback models for large-scale many-objective optimization, *Future Generation Computer Systems*, 107, 49–69, 2020.

[17] Hajlaoui, Emna, Zaier, Aida, Khlifi, Abdelhakim, Ghodhbane, Jihed, Hamed, Mouna Ben, Sbita, Lassâad (2020). 4G and 5G technologies: A Comparative Study, *2020 5th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, 1–6, 2020.

[18] Hao, Yunman (2021). Investigation and technological comparison of 4G and 5G networks, *Journal of Computer and Communications*, 9, 36–43, 2021.

[19] He, Wenjian and Qi, Xiaogang and Liu, Lifang (2021). A novel hybrid particle swarm optimization for multi-UAV cooperate path planning, *Applied Intelligence*, 51, 7350–7364, 2021.

[20] Hong, Dooyoung, Lee, Seonhoon, Cho, Young Hoo, Baek, Donkyu, Kim, Jaemin, Chang, Naehyuck (2020). Least-energy path planning with building accurate power consumption model of rotary unmanned aerial vehicle, *IEEE Transactions on Vehicular Technology*, 69, 14803–14817, 2020.

[21] Hong, Wei, Jiang, Zhi Hao, Yu, Chao, Hou, Debin, Wang, Haiming, Guo, Chong, Hu, Yun, Kuai, Le, Yu, Yingrui, Jiang, Zhengbo, others (2021). The role of millimeter-wave technologies in 5G/6G wireless communications, *IEEE Journal of Microwaves*, 1, 101–122, 2021.

[22] Houssein, Essam H, Gad, Ahmed G, Hussain, Kashif, Suganthan, Ponnuthurai Nagaratnam (2021). Major advances in particle swarm optimization: theory, analysis, and application, *Swarm and Evolutionary Computation*, 63, 100868, 2021.

[23] Huang, Yingqian, Cui, Miao, Zhang, Guangchi, Chen, Wei (2020). Bandwidth, power and trajectory optimization for UAV base station networks with backhaul and user QoS constraints, *IEEE Access*, 8, 67625–67634, 2020.

[24] Huang, Hailong, Savkin, Andrey V, Ni, Wei (2021). Online UAV trajectory planning for covert video surveillance of mobile targets, *IEEE Transactions on Automation Science and Engineering*, 19, 735–746, 2021.

[25] Huang, Hailong, Savkin, Andrey V (2022). Deployment of heterogeneous UAV base stations for optimal quality of coverage, *IEEE Internet of Things Journal*, 9, 16429–16437, 2022.

[26] Jiang, Jia-Jia, Wei, Wen-Xue, Shao, Wan-Lu, Liang, Yu-Feng, Qu, Yuan-Yuan (2021). Research on large-scale bi-level particle swarm optimization algorithm, *Ieee Access*, 9, 56364–56375, 2021.

[27] Jiang, Xu, Sheng, Min, Nan, ZHAO, Chengwen, XING, Weidang, LU, Xianbin, WANG (2022). Green UAV communications for 6G: A survey, *Chinese Journal of Aeronautics*, 35, 19–34, 2022.

[28] Khorov, Evgeny, Krasilov, Artem, Selnitskiy, Ilya, Akyildiz, Ian F (2020). A framework to maximize the capacity of 5G systems for ultra-reliable low-latency communications, *IEEE transactions on mobile computing*, 20, 2111–2123, 2020.

[29] Ladosz, Pawel, Weng, Lilian, Kim, Minwoo, Oh, Hyondong (2022). Exploration in deep reinforcement learning: A survey, *Information Fusion*, 85, 1–22, 2022.

[30] Lan, Rushi, Zhu, Yu, Lu, Huimin, Liu, Zhenbing, Luo, Xiaonan (2020). A two-phase learning-based swarm optimizer for large-scale optimization, *IEEE transactions on cybernetics*, 51, 6284–6293, 2020.

[31] Le, Trung-Kien, Salim, Umer, Kaltenberger, Florian (2020). An overview of physical layer design for ultra-reliable low-latency communications in 3GPP releases 15, 16, and 17, *IEEE access*, 9, 433–444, 2020.

[32] Li, Mushu, Cheng, Nan, Gao, Jie, Wang, Yinlu, Zhao, Lian, Shen, Xuemin (2020). Energy-efficient UAV-assisted mobile edge computing: Resource allocation and trajectory optimization, *IEEE Transactions on Vehicular Technology*, 69, 3424–3438, 2020.

[33] Lindqvist, Björn and Mansouri, Sina Sharif and Agha-mohammadi, Ali-akbar and Nikolakopoulos, George (2020). Nonlinear MPC for collision avoidance and control of UAVs with dynamic obstacles, *IEEE robotics and automation letters*, 5, 6001–6008, 2020.

[34] Marojevic, Vuk, Guvenc, Ismail, Dutta, Rudra, Sichitiu, Mihail L, Floyd, Brian A (2020). Advanced wireless for unmanned aerial systems: 5G standardization, research challenges, and AERPAW architecture, *IEEE Vehicular Technology Magazine*, 15, 22–30, 2020.

[35] Qian, Bo, Zhou, Haibo, Ma, Ting, Yu, Kai, Yu, Quan, Shen, Xuemin (2020). Multi-operator spectrum sharing for massive IoT coexisting in 5G/B5G wireless networks, *IEEE Journal on Selected Areas in Communications*, 39, 881–895, 2020.

[36] Rezwan, Sifat, Choi, Wooyeol (2022). Artificial intelligence approaches for UAV navigation: Recent advances and future challenges, *IEEE access*, 10, 26320–26339, 2022.

[37] Samdanis, Konstantinos, Taleb, Tarik (2020). The road beyond 5G: A vision and insight of the key technologies, *IEEE Network*, 34, 135–141, 2020.

[38] Schmid, Lukas, Pantic, Michael, Khanna, Raghav, Ott, Lionel, Siegwart, Roland, Nieto, Juan (2020). An efficient sampling-based method for online informative path planning in unknown environments, *IEEE Robotics and Automation Letters*, 5, 1500–1507, 2020.

[39] Shafique, Kinza, Khawaja, Bilal A, Sabir, Farah, Qazi, Sameer, Mustaqim, Muhammad (2020). Internet of things (IoT) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5G-IoT scenarios, *Ieee Access*, 8, 23022–23040, 2020.

[40] Shami, Tareq M, El-Saleh, Ayman A, Alswaitti, Mohammed, Al-Tashi, Qasem, Summakieh, Mhd Amen, Mirjalili, Seyedali (2022). Particle swarm optimization: A comprehensive survey, *Ieee Access*, 10, 10031–10061, 2022.

[41] Siddiqui, Maraj Uddin Ahmed, Qamar, Faizan, Ahmed, Faisal, Nguyen, Quang Ngoc, Hassan, Rosilah (2021). Interference management in 5G and beyond network: Requirements, challenges and future directions, *IEEE Access*, 9, 68932–68965, 2021.

[42] Soltani, AR, Fernando, T (2004). A fuzzy based multi-objective path planning of construction sites, *Automation in construction*, 13, 717–734, 2004.

[43] Tahir, Mohammad, Habaebi, Mohamed Hadi, Dabbagh, Mohammad, Mughees, Amna, Ahad, Abdul, Ahmed, Kazi Istiaque (2020). A review on application of blockchain in 5G and beyond networks: Taxonomy, field-trials, challenges and opportunities, *IEEE Access*, 8, 115876–115904, 2020.

[44] Tran, Dinh-Hieu and Vu, Thang X and Chatzinotas, Symeon and ShahbazPanahi, Shahram and Ottersten, Björn (2020). Coarse trajectory design for energy minimization in UAV-enabled, *IEEE Transactions on Vehicular Technology*, 69, 9483–9496, 2020.

[45] Vaezi, Mojtaba, Azari, Amin, Khosravirad, Saeed R, Shirvanimoghaddam, Mahyar, Azari, M Mahdi, Chasaki, Danai, Popovski, Petar (2022). Cellular, wide-area, and non-terrestrial IoT: A survey on 5G advances and the road toward 6G, *IEEE Communications Surveys & Tutorials*, 24, 1117–1174, 2022.

[46] Van Huynh, Dang, Do-Duy, Tan, Nguyen, Long D, Le, Minh-Tuan, Vo, Nguyen-Son, Duong, Trung Q (2021). Real-time optimized path planning and energy consumption for data collection in unmanned ariel vehicles-aided intelligent wireless sensing, *IEEE Transactions on Industrial Informatics*, 18, 2753–2761, 2021.

[47] Wang, Zi-Jia, Zhan, Zhi-Hui, Kwong, Sam, Jin, Hu, Zhang, Jun (2020). Adaptive granularity learning distributed particle swarm optimization for large-scale optimization, *IEEE transactions on cybernetics*, 51, 1175–1188, 2020.

[48] Wang, Feng, Zhang, Heng, Zhou, Aimin (2021). A particle swarm optimization algorithm for mixed-variable optimization problems, *Swarm and Evolutionary Computation*, 60, 100808, 2021.

[49] Xie, Junfei, Carrillo, Luis Rodolfo Garcia, Jin, Lei (2020). Path planning for UAV to cover multiple separated convex polygonal regions, *IEEE Access*, 8, 51770–51785, 2020.

[50] Xu, Cheng, Xu, Ming, Yin, Chanjuan (2020). Optimized multi-UAV cooperative path planning under the complex confrontation environment, *Computer Communications*, 162, 196–203, 2020.

[51] Xu, Zhefan, Deng, Di, Shimada, Kenji (2021). Autonomous UAV exploration of dynamic environments via incremental sampling and probabilistic roadmap, *IEEE Robotics and Automation Letters*, 6, 2729–2736, 2021.

[52] Yan, Ming and Yuan, Huimin and Xu, Jie and Yu, Ying and Jin, Libiao (2021). Task allocation and route planning of multiple UAVs in a marine environment based on an improved particle swarm optimization algorithm, *EURASIP Journal on Advances in Signal Processing*, 2021, 1–23, 2021.

[53] Yao, Huang, Qin, Rongjun, Chen, Xiaoyu (2019). Unmanned aerial vehicle for remote sensing applications—A review, *Remote Sensing*, 11, 1443, 2019.

[54] Yasin, Jawad N, Mohamed, Sherif AS, Haghbayan, Mohammad-Hashem, Heikkonen, Jukka, Tenhunen, Hannu, Plosila, Juha (2020). Unmanned aerial vehicles (uavs): Collision avoidance systems and approaches, *IEEE access*, 8, 105139–105155, 2020.

[55] Yi, Jiao-Hong, Xing, Li-Ning, Wang, Gai-Ge, Dong, Junyu, Vasilakos, Athanasios V, Alavi, Amir H, Wang, Ling (2020). Behavior of crossover operators in NSGA-III for large-scale optimization problems, *Information Sciences*, 509, 470–487, 2020.

[56] Zhang, Xiao, Duan, Lingjie (2020). Energy-saving deployment algorithms of UAV swarm for sustainable wireless coverage, *IEEE Transactions on Vehicular Technology*, 69, 10320–10335, 2020.

Cite this paper as:

Bui Trong Dung, Trinh Van Chien, Van Son Nguyen, Minh T. Hoang, Phuong Nhung Do, Hoang Anh Dang (2025). Multiple-UAV Path Planning in Obstacle Environments, *International Journal of Computers Communications & Control*, 20(4), 6847, 2025.
https://doi.org/10.15837/ijccc.2025.4.6847