

Article

Energy-Efficient Internet of Drones Path-Planning Study Using Meta-Heuristic Algorithms

Gamil Ahmed ¹, Tarek Sheltami ¹, Mustafa Ghaleb ², Mosab Hamdan ², Ashraf Mahmoud ^{1,*}
and Ansar Yasar ³

- ¹ Computer Engineering Department, Interdisciplinary Research Center of Smart Mobility and Logistics, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia; gamil.ahmed@kfupm.edu.sa (G.A.); tarek@kfupm.edu.sa (T.S.)
- ² Interdisciplinary Research Center for Intelligent Secure Systems, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia; mustafa.ghaleb@kfupm.edu.sa (M.G.); mosab.mohamed@kfupm.edu.sa (M.H.)
- ³ Transportation Research Institute (IMOB), Hasselt University, 3500 Hasselt, Belgium; ansar.yasar@uhasselt.be
- * Correspondence: ashraf@kfupm.edu.sa

Abstract: The increasing popularity of unmanned aerial vehicles (UAVs), commonly known as drones, in various fields is primarily due to their agility, quick deployment, flexibility, and excellent mobility. Particularly, the Internet of Drones (IoD)—a networked UAV system—has gained broad-spectrum attention for its potential applications. However, threat-prone environments, characterized by obstacles, pose a challenge to the safety of drones. One of the key challenges in IoD formation is path planning, which involves determining optimal paths for all UAVs while avoiding obstacles and other constraints. Limited battery life is another challenge that limits the operation time of UAVs. To address these issues, drones require efficient collision avoidance and energy-efficient strategies for effective path planning. This study focuses on using meta-heuristic algorithms, recognized for their robust global optimization capabilities, to solve the UAV path-planning problem. We model the path-planning problem as an optimization problem that aims to minimize energy consumption while considering the threats posed by obstacles. Through extensive simulations, this research compares the effectiveness of particle swarm optimization (PSO), improved PSO (IPSO), comprehensively improved PSO (CIPSO), the artificial bee colony (ABC), and the genetic algorithm (GA) in optimizing the IoD's path planning in obstacle-dense environments. Different performance metrics have been considered, such as path optimality, energy consumption, straight line rate (SLR), and relative percentage deviation (RPD). Moreover, a nondeterministic test is applied, and a one-way ANOVA test is obtained to validate the results for different algorithms. Results indicate IPSO's superior performance in terms of IoD formation stability, convergence speed, and path length efficiency, albeit with a longer run time compared to PSO and ABC.

Keywords: Internet of Drones; obstacle avoidance; path planning; energy consumption; UAV; convergence speed; PSO; CIPSO; genetic algorithm; ABC



Citation: Ahmed, G.; Sheltami, T.; Ghaleb, M.; Hamdan, M.; Mahmoud, A.; Yasar, A. Energy-Efficient Internet of Drones Path-Planning Study Using Meta-Heuristic Algorithms. *Appl. Sci.* **2024**, *14*, 2418. <https://doi.org/10.3390/app14062418>

Academic Editor: Luis Javier Garcia Villalba

Received: 29 January 2024
Revised: 23 February 2024
Accepted: 24 February 2024
Published: 13 March 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Recently, UAVs have attracted much research attention due to recent technology advancement. Unlike terrestrial nodes, UAV operation has fewer constraints, which enables a more flexible mobility model. The communication devices integrated with a UAV, including connectivity attributes and network functionality, can effectively establish connections with a variety of urban devices. Notably, UAVs employ a flexible network of nodes whose actions should be meticulously controlled [1–5]. Nowadays, in different fields, tremendous attention is given to multi-agent systems for complex problem solutions that individuals cannot achieve [6–9].

To achieve UAV communication, the Internet of Drones (IoD), a swarm of UAVs, offers several benefits and applications. Firstly, it provides faster and more flexible deployment in harsh and complex environments. Secondly, the IoD enables effective coordination among UAVs for common missions. Thirdly, it supports various use cases in wireless networks, including enhancing wireless capacity and coverage at hotspots or temporary outdoor events [10–13]. Fourthly, the IoD can be utilized in disaster relief activities and public safety scenarios when traditional networks are damaged [3,14]. Fifthly, they are valuable for data gathering applications, providing real-time topological information and object location [15,16]. Lastly, the IoD is widely used in environment monitoring, including search and rescue operations and surveillance, thanks to their equipped cameras and sensors.

Generating an obstacle-free path for formations of the IoD is the primary and most difficult objective in environments with dense swarms and various constraints [17–22]. Furthermore, employing deterministic methods to construct paths for IoD formations necessitates significant storage capacity and extended execution time [23–26]. This is due to the fact that the optimization criteria encompass various factors, such as minimizing path length, avoiding obstacles, completing missions in shorter time duration, and minimizing energy consumption, which need to be jointly optimized [27,28].

Despite ample UAV applications, onboard energy has fundamentally limited UAV performance and endurance, which is practically finite [29–31]. As a result, most applications cannot reach their full potential. Furthermore, in the case of a lengthy path, where the drone is required to carry out its mission in a large area, it becomes crucial to minimize the energy consumption of the UAV in order to generate a safe path and successfully reach its destination. To provide a clearer perspective, enhancing the energy efficiency of the UAV enables it to accomplish more tasks within a shorter time frame. Conversely, if the drone lacks an efficient energy mechanism, it would need to undergo more recharge cycles, thereby prolonging the mission duration [32]. Thus, reducing energy consumption during drone missions poses a significant challenge.

To address this problem, powerful optimization techniques, such as game theory [33], convex optimization [34], stochastic optimization, and transport theory [35], have been applied. Although several non-linear approaches have shown satisfactory results, they are typically confirmed in narrow and primary habitats, making them impossible to apply in large-scale and complex environments [11,29].

In the past few years, popular meta-heuristic algorithms have been seen as a powerful global optimization approach for solving UAV trajectory problems. Examples include the genetic algorithm (GA) for mobile robotic path planning [36,37], particle swarm optimization (PSO) [38], and the artificial bee colony (ABC) [39]. However, existing research in UAV path planning within the IoD often falls short in effectively navigating complex environments, prioritizing energy efficiency, and thoroughly comparing meta-heuristic optimization techniques. Furthermore, there is a notable lack of extensive, rigorously validated simulation studies to test various path-planning approaches.

Although meta-heuristic algorithms have been implemented for path planning in the literature, there is limited consideration of IoD formation by researchers. Furthermore, when the IoD scenarios are considered, researchers typically employ simple scenarios and sparse environments. Specifically, the drones are assigned tasks in diverse environmental conditions, including dense environments that require collaborative efforts to accomplish their missions. The objective of this analysis is to establish a baseline performance study for solving drone path planning in highly dense environments as well as to evaluate the limitations and strengths of these algorithms. In accomplishing this, we aim to gain insights into the suitability and challenges associated with employing these algorithms in such scenarios.

In this work, we model the path planning of the IoD as a single objective optimization problem where the paths of the IoD are constrained to avoid collision with obstacles and with other UAVs in highly dense environments. We design an objective function involving two important objectives, namely path length and energy consumption. The path can

be optimized either for minimum path length or minimum energy consumption. Using simulations, we consider energy consumption as an objective to be minimized. Moreover, the performance of different meta-heuristic approaches is evaluated and compared. The aim is to study the efficiency and the behavior of these algorithms under different conditions. The optimization techniques covered in this work include PSO, IPSO, CIPSO [40], GA, and ABC, which are extensively used in several optimization problems. Extensive simulation experiments have been conducted to investigate the performance of these algorithms.

The contributions of this work are summarized as follows.

We conduct extensive simulations to evaluate and compare the performance of five meta-heuristic approaches: IPSO, CIPSO, PSO, GA, and ABC, to solve the IoD path-planning problem in highly dense environment scenarios. The ANOVA test is applied to provide a statistically robust analysis of the differences among them, ensuring an effective comparison. This helps to gain a deeper understanding of the performance of the respective approach under various conditions and scenarios, and to provide valuable insights for practical implementation. Furthermore, the comparative analysis sheds light on the suitability of each algorithm for different application scenarios, aiding researchers in selecting the most appropriate algorithm for their specific IoD network requirements. Moreover, this study opens an opportunity for future works to design proper algorithms based on the outlined limitations and strengthens each of the considered approaches.

The rest of the paper is organized as follows. Section 2 presents the state of the art in the field of UAV path planning. Section 3 defines the energy model for path planning, the IoD path-planning problem, objectives and constraints, and the meta-heuristic algorithms applied in this study. Results and discussion are illustrated in Section 4. Section 5 presents the discussion of weaknesses and strengths of the analyzed algorithms. Finally, conclusions are drawn in Section 6.

2. Literature Review

Several works have been developed in the field of UAV path-planning algorithms, highlighting different applications, optimization approaches, and advancements in the field.

The author in [41] discussed the development of a path-planning technique for UAVs in target coverage problems. The paper presented an algorithm that combined a genetic algorithm with three different approaches during the initial population phase to address altitude limitations and ensure safe paths. Another framework presented in [42], called Sparrow Particle Swarm Algorithm (SPSA), was proposed to solve path planning in UAVs. To reduce blind search, this framework enhances initialization and updated positions, and gives importance to the start–end line. The target reach in deadlocked areas is improved by utilizing adaptive variable speed escapes. Simplification and interpolation techniques using cubic splines are employed to enhance path smoothness for real flight trajectories.

In a survey presented in [43], an overview of 3D path-planning algorithms for UAVs was provided, categorizing them into five categories: node-based algorithms, bio-inspired algorithms, mathematical model-based algorithms, sampling-based algorithms, and multi-fusion-based algorithms. Another survey in [44] focused on computational intelligence (CI) methods for UAV path planning, specifically exploring time domain-based methods in both 2D and 3D environments. Additionally, the work in [45] provided a comprehensive survey of UAV path-planning algorithms, with a particular emphasis on applications, such as disaster management, communication networks, and general scenarios. The survey highlighted that 67% of UAV path-planning algorithms focus on general applications, 22% on disaster management, and only 11% focus on communication networks. Additionally, researchers proposed an integrated IoT-powered UAV-based smart city management system to address earthquakes, disasters, and bushfires.

In [46], a multi-frequency vibrational genetic algorithm (mVGA) was introduced for UAV path planning in city and sinusoidal environmental models. The algorithm incorporated a new mutation application strategy and emphasized both local random and global random diversity. The initial population phase of the algorithm integrated concepts

from the Voronoi diagram and clustering methods. The mVGA algorithm demonstrated reduced computational time compared to three other genetic algorithms.

The PSO algorithm has played an important role in numerous UAV path-planning works due to its effectiveness in optimization problems [47]. Furthermore, PSO offers several advantages that make it suitable for real-world applications, relative to other path-planning algorithms. These advantages are fast convergence speed, the capability to search for optimal paths in large search spaces, robustness, and the ability to handle dynamic environments. Additionally, PSO is easy to implement and can be adapted to different path-planning problems by adjusting the fitness function or the acceleration coefficients. Inspired by these advantages, ref. [48] proposed a spherical vector-based particle swarm optimization (SPSO) approach. The approach formulated a cost function based on path length to convert path-planning into an optimization problem that incorporated all constraints for safe and feasible UAV operation.

In [49], an improved PSO algorithm was presented for real-time UAV path planning. A chaos strategy was utilized to prevent particles from converging to local optima and the path quality was enhanced using the Dijkstra algorithm. The work in [50] introduced a path-planning approach that considered the dynamic properties of the UAV and the complexity of real 3D environments. It combined two non-deterministic algorithms, the genetic algorithm (GA) and PSO, to address complexity and reduce computation time. Furthermore, the GBPSO, an enhanced version of PSO, was proposed in [51] to improve the performance of 3D path planning for UAVs. This algorithm incorporated a competition strategy into the standard PSO framework to optimize the global best solution.

In [52], a method was presented to address the path-planning problem for UAVs operating in adversarial environments with radar-guided surface-to-air missiles (SAMs) and unknown threats. An improved PSO algorithm is utilized, along with preprocessing steps involving Voronoi diagram generation and the use of the Dijkstra algorithm to obtain an initial path for the multi-swarm PSO algorithm. The multi-swarm PSO algorithm employed multiple swarms with sub-swarms to balance exploration and exploitation. Researchers in [40] proposed a three-dimensional path-planning algorithm for UAV formation using an enhanced PSO approach. The algorithm incorporated a chaos-based logistic map to improve the initial distribution of particles and replaced constant acceleration coefficients and maximum velocity with adaptive linearly varying values, leading to an improved optimal solution.

In spite of the improvements associated with the PSO algorithm in the literature, it still encounters the issue of premature convergence and the risk of becoming trapped in local minima when applied to complex problems. To highlight the limitations and strengths of PSO and other meta-heuristic algorithms, such as GA, ABC, CIPSO, and IPSO, we conduct a thorough performance evaluation and consider different metrics. Furthermore, we demonstrate the differences among these algorithms by analyzing the collected data and employing a statistical test, specifically the one-way ANOVA test. This study effectively guides researchers in enhancing the current approach to overcome the challenges related to local optima and slow convergence issues.

3. Meta-Heuristic Algorithms for Energy-Efficient IoD Path-Planning Description

In this section, we provide a comprehensive explanation of the energy model, the IoD path planning, and the heuristic approaches used in the context of the IoD path planning.

3.1. Energy Model for UAV Path Planning

The energy model plays a crucial role in understanding the energy consumption of drones during their flight. Notably, despite the significant features of UAVs, such as their versatility and agility, their flight time is still limited by the onboard battery capacity, which often leads to incomplete missions [21]. To cope with this limitation, the UAV path planning is formulated as a minimization optimization problem in which the energy consumption is taken into consideration when planning the UAV trajectory. In this regard, the path

is optimized such that the energy consumption is minimum. This can be achieved by considering the energy as an objective to be minimized. The energy consumption of the generated path is calculated using the drone's energy model, as outlined by Di et al. in [53]. This model takes into account various maneuvering activities, including horizontal and vertical movements as well as hovering. In the context of path planning, the emphasis is on understanding how different movements, such as hovering, horizontal movement, and vertical movement, impact energy consumption. Therefore, the energy consumption for each operation is determined according to the methodology presented in [53]. It has to be stressed that the speeds of all UAVs are assumed to be constant in this study.

The climbing energy for a distance Δh and speed v_{clmb} is given by

$$E_{clmb}(\Delta h) = P_{clmb} \frac{\Delta h}{v_{clmb}}, \quad (1)$$

where P_{clmb} is the power required for climbing.

The descent energy for a distance of Δh and velocity v_{desce} is given by

$$E_{desce}(\Delta h) = P_{desce} \frac{\Delta h}{v_{desce}}, \quad (2)$$

where P_{desce} is the power required for descending.

The hovering-in-place energy for a duration of time from $t1$ to $t2$ is given by

$$E_{hov} = P_{hov}(t2 - t1), \quad (3)$$

where P_{hov} is the power required for hovering.

Flying horizontally is given as a function of speed and can be expressed as follows:

$$E_{horz} = P_v \frac{h_{dis}}{v}, \quad (4)$$

where h_{dis} is the horizontal distance, P_v is the power required for flying horizontally, and v is the horizontal speed.

Therefore, the energy can be written as follow:

$$E = E_{clmb} + E_{desce} + E_{hov} + E_{horz}. \quad (5)$$

3.2. IoD Path Representation and Obstacle Modeling

3.2.1. IoD Path Representation

In this study, we assume that the information about the environment is known in advance. In specific, the locations and sizes of obstacles are precisely known. The UAVs fly from starting positions (SPs) to the target positions (TPs) as safely as possible. The goal is to find the optimal paths for all drones in the IoD from SPs to TPs. To accomplish this, several factors should be considered, such as the formation of the IoD and terrain obstacles. The IoD formation involves multiple drones flying together in an environment where each drone should be equipped with an efficient path-planning technique to ensure that drones arrive at their destinations safely. The 3D path of the drone is represented by a set of waypoints that are connected to form a path from the starting point to the destination.

Let M be the number of waypoints ranging from 1 to M ; then, the path can be represented as follows:

$$Path = (P_1(x_1, y_1, z_1), P_2(x_2, y_2, z_2), \dots, P_M(x_m, y_m, z_m)). \quad (6)$$

3.2.2. Obstacle Model

In this paper, the obstacle is modeled as a half sphere as in [32], where the q^{th} obstacle is represented as

$$Obs_q = (x_q, y_q, z_q, r_q), \quad (7)$$

where (x_q, y_q, z_q) is the coordinate of the obstacle center and r_q is the radius of the obstacle. For the k^{th} obstacle, a 3D coordinate can be expressed as follows:

$$x_k = r_k \cos(\theta) \sin(\phi) + x_{k0}, \tag{8}$$

$$y_k = r_k \sin(\theta) \sin(\phi) + y_{k0}, \tag{9}$$

$$z_k = r_k \cos(\phi) + z_{k0}, \tag{10}$$

where (x_{k0}, y_{k0}, z_{k0}) is the coordinate of the k^{th} obstacle's center, while (x_k, y_k, z_k) is the coordinates of the k^{th} obstacle. r_k corresponds to the radius of the obstacle, with $\theta \in [0, 2\pi]$, and $\phi \in [0, \frac{\pi}{2}]$.

3.3. Objective Function and Constraints

The primary source of UAV power is its onboard battery, which supports all activities, including processing, flight, and communication. However, the battery capacity is not adequate for extended flight missions. To cope with this problem, the UAV trajectory is formulated as an optimization problem, where the energy consumption is the objective to be minimized. Thus, the optimal trajectory is the collision-free path that requires minimum energy. To plan the path of the UAV, the sequence of waypoints is chosen to satisfy all constraints and achieve the minimum energy consumption for the case of energy optimization or achieve minimum path length for the case of path length optimization.

It has to be stressed that trying all possibilities to select the best sequence of waypoints is considered a nondeterministic polynomial (NP)-hard problem that requires lots of resources and unbounded execution time. That is, this problem is difficult to solve using a deterministic approach in a reasonable time. The problem is formulated as a single objective optimization problem, and the total energy for each path is calculated using Equation (11).

$$F_{Energy} = \sum_{i=1}^{N_w-1} E_{S_i}, \tag{11}$$

where N_w is the number of path waypoints, and E_{S_i} is the energy of segment i .

Another important objective is the path length. In some missions, a longer path is not desirable because of the drone's limited power. Thus, the shortest path is preferable. The path length can be calculated as follows.

$$F_{pl} = \sum_{i=1}^{N_w-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2}, \tag{12}$$

where N_w is the number of waypoints, and (x_i, y_i, z_i) is the i^{th} path waypoint.

In this study, we consider both objectives: energy consumption and path length. Since the problem is formulated as a single objective optimization problem, we study each objective individually. Moreover, in a threat-prone environment, the path is constrained to avoid colliding with surrounding obstacles, and with other drones in the formation. To avoid colliding with obstacles, the distance from all waypoints to the center of any obstacle should be greater than the size of the obstacle. If even one waypoint goes through the obstacle, the path is considered not feasible and is penalized with high cost, η . Furthermore, the drones in the IoD formation should not fly to the same location at the same time. To accomplish this, a safety distance between drones should be maintained.

In light of the above description, the objective function for energy consumption and path length is a minimization optimization problem and can be expressed as in Equation (13). It has to be stressed that the proposed objective function either minimizes the energy consumption or the path length, but not both objectives at the same time.

$$F_{Obj} = \begin{cases} F_{Energy} + \lambda(F_{Oc} + F_{Mc}), & \text{for energy,} \\ F_{pl} + \lambda(F_{Oc} + F_{Mc}), & \text{for path length,} \end{cases} \tag{13}$$

subject to the following constraints:

$$D_{T_0}(i, j) = \sqrt{(x(i) - x_o(j))^2 + (y(i) - y_o(j))^2 + (z(i) - z_o(j))^2} \geq R_j, \quad (14)$$

$$D_{T_u}(i, j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \geq SD, \quad (15)$$

where $D_{T_0}(i, j)$ is the distance between $drone_i$ and $obstacle_j$ and should be greater than the size of $obstacle_j(R_j)$ for $i = 1, 2, \dots, N_U$, and $j = 1, 2, \dots, N_{obs}$. $D_{T_u}(i, j)$ for $i, j = 1, 2, \dots, N_U, i \neq j$ is the distance between UAVs and should be greater than the safety distance, denoted by SD . F_{Energy} , F_{Oc} , and F_{Mc} are functions that account for energy consumption, the likelihood of colliding, and member collision, respectively. The coordinate points (x_i, y_i, z_i) are the waypoints of $path_i$, while (x_o, y_o, z_o) are the centers of obstacles.

F_{Oc} and F_{Mc} can be expressed as follows:

$$F_{Oc} = \begin{cases} \eta & \text{if the path goes through an obstacle,} \\ 0 & \text{Otherwise.} \end{cases}$$

$$F_{Mc} = \begin{cases} \eta & \text{if the drones fly to the same location at the same time,} \\ 0 & \text{Otherwise.} \end{cases}$$

where η is a large penalty.

The functions F_{Oc} and F_{Mc} represent the collision with environment obstacles and members in the formation, respectively. These functions are penalized with a large value, denoted by η , when the path goes through obstacles or includes collision with other drones. This penalty significantly increases the cost of the path, resulting in its removal from the desired path.

3.4. Meta-Heuristic Algorithms

In this subsection, we will briefly explain some meta-heuristic approaches that will be utilized to solve the IoD path-planning problem. In terms of prospects, each algorithm has potential applications in the context of the IoD. PSO, IPSO, GA, and ABC can be applied to optimize path planning for the IoD in various scenarios, including disaster relief, surveillance operations, and environment monitoring. These algorithms can contribute to enhancing the efficiency and sustainability of UAV missions within the IoD formation.

3.4.1. Genetic Algorithm (GA)

GA [37] is one of the evolutionary algorithms utilized to solve NP-hard optimization problems that cannot be solved by deterministic methods in polynomial time. The notion of natural selection is used by the GA and applied to a population of potential solutions to the respective problem. The operation of the GA is shown in Figure 1.

First, the initial population is randomly generated and then evaluated by the fitness function. The selection strategy is utilized to select the parents and produce new children through crossover and mutation. Each parent in UAV path planning represents a complete path from the start point to the end point.

The selection method is used to determine which individuals are more likely to generate the best offspring. In this method, the most suitable individuals win the competition of reproduction, while the least fit are excluded, which improves overall fitness.

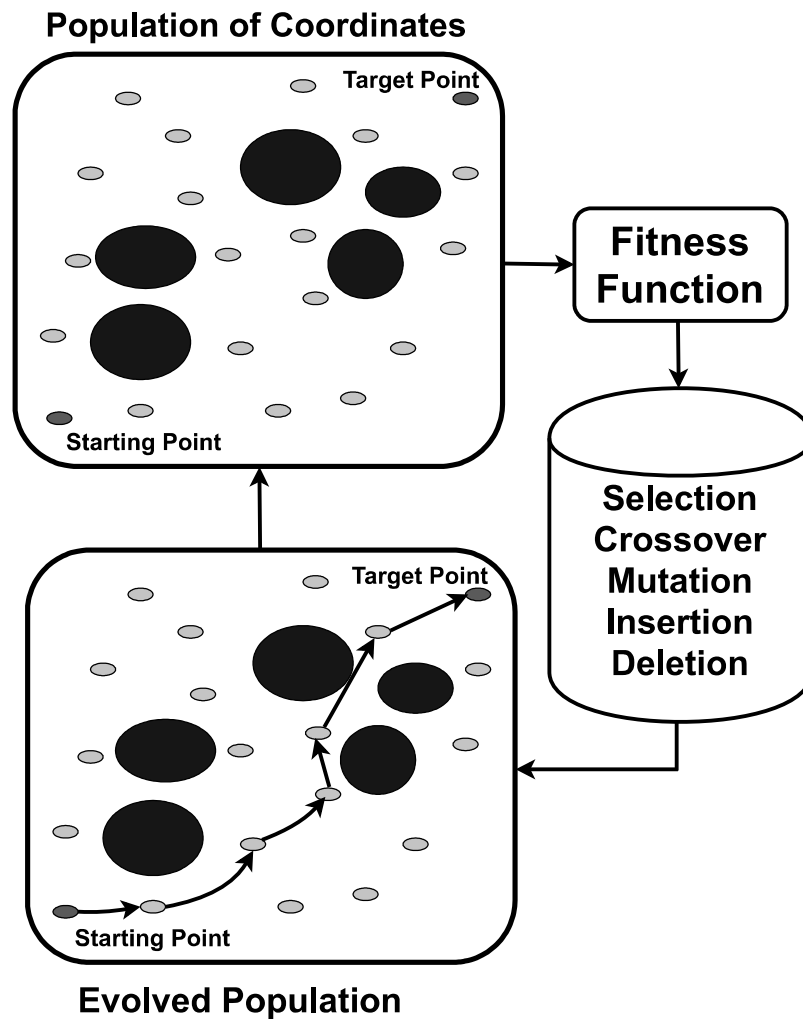


Figure 1. Genetic algorithm for UAV path planning: blue dots represent obstacles, green dots represent source/destination, and gray dots represent the waypoints.

In this part, we briefly explain the process of GA to generate a new solution, including selection, crossover, and mutation.

- **Selection techniques:** There are several selection techniques. The rank selection method always chooses individuals with the best fitness. If n individuals make up the population, the applied selection is carried out by ranking the solution based on its fitness. The selection method will select the k best individuals. For each individual, the probability of being selected is proportional to the corresponding fitness. The tournament selection technique is another selection method that uses proportional selection on pairs of individuals, then chooses from among these pairs the individual with the highest fitness score with a probability $P \in [0.5, 1]$. This process is repeated n times to obtain the n individuals of populations that will serve as parents. The uniform selection method selects individuals uniformly and without reliance on the fitness value. Each individual has $1/N$ probability of being selected, where N is the total number of individuals in the population.
- **Crossover mechanism:** When two chromosomes have been selected, a crossover is made. A crossover is a mechanism used to generate new children from two individuals. The new individuals (children) inherit certain features from their parents. The genes are selected from two parents and then children are generated. The crossover probability represents the frequency with which the crossovers are applied. If there is no crossover, i.e., probability of crossover is 0%, then children are an exact copy of the parents. If there is crossover, the children are partially similar to their par-

ents. However, if the probability is set to 100%, then all the children are generated by crossover.

- Mutation strategies: Mutations are performed on a small proportion of randomly selected individuals. This process generates a new population. The process is repeated many times to imitate the principle of evolution, which only takes its meaning over a large number of generations. The process can be stopped after an arbitrary number of generations or when a solution has a sufficiently satisfactory score. In mutation strategy, one gene can be substituted in another chromosome randomly. A mutation rate is generally chosen between 0.001 and 0.01. It is necessary to choose a relatively low value to avoid random search and keep the principle of selection and evolution. The mutation serves to avoid premature convergence of the algorithm. For example, during an optimum search, the mutation serves to avoid convergence towards local optima.

3.4.2. Particle Swarm Optimization (PSO)

PSO [38], introduced in 1995, is a population-based algorithm and inspired by the movement of individuals in a swarm, such as a bird flock. The movement of particles is based on their current locations and speeds. The neighboring particles, known as informants, share their information about the surrounding environment. Each particle is represented as a vector of position and velocity. In UAV path planning, the particle represents a complete path from the starting location to the end location, i.e., a sequence of waypoints.

The velocity of the particle is updated based on the current position $x(t)$ and current velocity $v(t)$. In specific, the updated velocity $v(t+1)$ and the updated location $x(t+1)$ are computed as follows:

$$v_{t+1} = \omega v_t + c_1 r_1 (pBest_t - x_t) + c_2 r_2 (gBest_t - x_t) \quad (16)$$

$$x(t+1) = x(t) + v(t+1) \quad (17)$$

where ω is the inertia coefficient, (c_1, c_2) are the acceleration coefficients, and (r_1, r_2) are uniform random numbers drawn from $[0, 1]$. The update in Equation (16) utilizes the optimal local solution of the swarm denoted by $pBest_t$, and the optimal global solution of the swarm, denoted by $gBest_t$. Subsequently, the generated solution is assessed and $pBest_{t+1}$ and $gBest_{t+1}$ are computed for the next iteration. The parameters for PSO are explained thoroughly in [32]. The process is repeated until the maximum iteration or maximum run time is reached. The pseudo-code of PSO for UAV path planning is illustrated in Algorithm 1.

Figure 2 shows the graphical representation of the evolution of particles in PSO. The figure depicts the positions and velocities of the i^{th} particle, denoted by X_i and V_i , respectively, at times $t-1$, t , and $t+1$. The figure illustrates how the particle interacts within the swarm and showcases its behavior. It is important to note that each particle in the PSO algorithm possesses its own information as well as neighbor information and swarm information. This knowledge affects the decision-making process regarding the particle's movement direction and speed. By considering (1) its personal best position, (2) the best position found by its neighbors, and (3) the global best position found by the entire swarm, a particle determines its next movement. The combination of these three key pieces of knowledge guides the particle's behavior, allowing it to explore and exploit the search space effectively.

Algorithm 1: Pseudo-code of standard PSO algorithm

```

begin
  Max_Iteration: maximum iteration
  INITIALIZATION;;
  initialize PSO parameters
  initialize position and velocity
  begin
    iter=0
    Calculate fitness value and obtain gBest and pBest
    while Iter < Max_Iteration do
      for i ← 1 to PopSize do
        Update the velocity of particle using Equation (16)
        Update the position of particle using Equation (17)
        Calculate fitness of generated particle;
        Update gBest and pBest
      end
      Increment Iter
    end
    Repeat until ideal best is obtained or run out of time
  end
  Return Optimal Paths for IoD
end

```

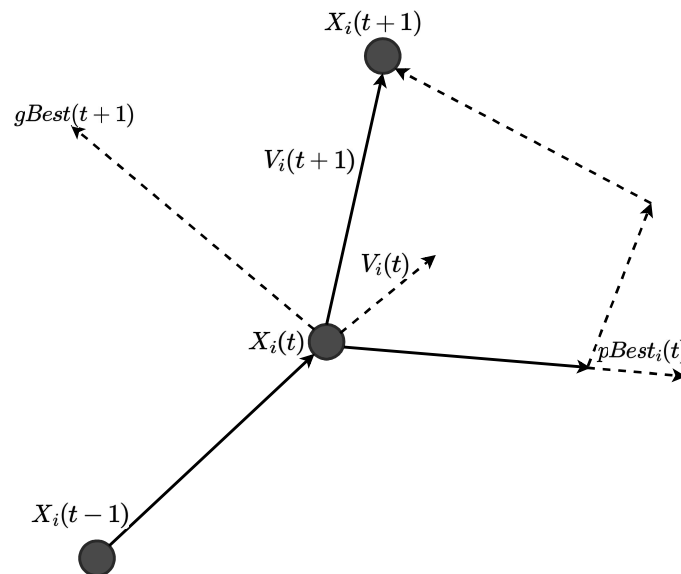


Figure 2. Graphical representation of PSO evolution.

3.4.3. Improved Particle Swarm Optimization

Authors in [32] improved the standard PSO to enhance the optimality and speed of convergence. Improvements include (1) improved initialization stage of swarm particles using chaos-based particle initialization as opposed to the conventional random initialization. This step helps to provide diverse swarm particles and thus speeds up the convergence. To accomplish this, the simplest logistic map for the IoD formation initialization is applied as presented in [32].

$$X_{n+1} = \mu X_n(1 - X_n), \quad (18)$$

where X_n represents the n^{th} chaotic variable, and μ is a bifurcation coefficient. Other improvements include (2) an improved update strategy by replacing inactive particles,

where particles lose their ability to search locally or globally, which may cause premature convergence, and (3) adaptive mutation of parameters such that their values change with time. In specific, the inertia parameter ω and a newly introduced parameter, denoted by ϵ , are computed as follows:

$$\omega(t) = \omega_{min} + \frac{MaxIt - t}{MaxIt} * (\omega_{max} - \omega_{min}), \quad (19)$$

where $MaxIt$ is the maximum simulation time, t is the current simulation time, and $(\omega_{min}, \omega_{max})$ are minimum and maximum values of inertia, respectively.

$$\epsilon = \epsilon_{max} - \frac{(\epsilon_{max} - \epsilon_{min})t}{MaxIt} \quad (20)$$

where ϵ_{max} and ϵ_{min} are constants, where $\epsilon_{max} > \epsilon_{min}$.

$$x(t + 1) = x(t) + \epsilon v(t + 1) \quad (21)$$

IPSO for the IoD path planning is explained in Algorithm 2.

Algorithm 2: Improved PSO algorithm

```

begin
  Limit: Threshold of trail
  Max_Iteration: maximum iteration
  INITIALIZATION;;
  initialize c1,c2
  initialize position and velocity with logistic map by Equation (18)
  begin
    iter=0
    Calculate fitness value and obtain gBest and pBest
    while Iter<Max_Iteration do
      Update  $\omega$  by Equation (19)
      Update  $\epsilon$  by Equation (20)
      for  $i \leftarrow 1$  to PopSize do
        Update the velocity of particle using Equation (16)
        Update the position of particle using Equation (21)
        Calculate fitness of generated particle;
        if particle is inactive then
          | Increment its trail
        else
          | Clear its trail
          | Update gBest and pBest
        end
      end
      for  $i = 1$  : to PopSize do
        if trail(i)>limit then
          | Replace inactive particle by new fresh one.
          | Clear its trail
        end
      end
      Increment Iter
    end
    Repeat until ideal best is obtained or run out of time
  end
  Return Optimal Paths for IoD
end

```

3.4.4. Artificial Bee Colony (ABC)

Artificial bee colony (ABC), presented in [54], is one of the most recently defined algorithms by Dervis Karaboga in 2005, motivated by the intelligent behavior of honey bees. Similar to PSO and differential evolution (DE) algorithms, ABC employs common control parameters, such as the size of the colony and the maximum number of cycles. ABC operates as a population-based search algorithm, where individuals, referred to as food positions, are iteratively modified by artificial bees over time. The primary goal of the bees is to locate food sources with high nectar levels, ultimately aiming to identify the one with the highest nectar content. In this system, artificial bees navigate a multidimensional search space, with employed and onlooker bees selecting food sources based on their experience and the experiences of other bees within their nest. They adjust their positions accordingly. In contrast, scouts randomly explore and select food sources without relying on previous experience. Whenever a new food source offers a higher nectar amount than the one stored in their memory, bees update their position accordingly, discarding the previous source. As a result, the ABC algorithm combines local search methods, performed by employed and onlooker bees, with global search methods, managed by onlookers and scouts, to balance the exploration and exploitation processes. ABC for the IoD path planning is explained in Algorithm 3.

Algorithm 3: ABC algorithm for IoD path planning

```

Limit: Threshold of trail, Max_Iteration: maximum iteration
INITIALIZATION:;
Position← initialize positions;
begin
  iter=0
  Calculate fitness value and obtain BestCost, BestPath
  while Iter<Max_Iteration do
    for i ← 1 to PopSize do
      NewPosition(i)← new solutions in the neighbourhood of position(i) for the
      employed
      Calculate fitness value and obtain BestCost, BestPath
      if solution is inactive then
        | Increment its trail
      else
        | Clear its trail, Update BestCost, BestPath
      end
    end
    for i = 1 : to PopSize do
      if trail(i)>limit then
        | Replace inactive solution by new fresh one,
        | Clear its trail
      end
    end
    Increment Iter
  end
  Repeat until ideal best is obtained or run out of time
end

```

3.5. Research Process of Path-Planning Problem

In this part, we summarize the research process of the path planning. Figure 3 shows the flowchart of the conducted research. The first step is problem formulation. The UAV path-planning problem is formulated as a minimization optimization problem. In this study, we consider path energy consumption as the main objective to be minimized. The energy model is explained in Section 3.1, which includes different maneuvering activities, such as vertical, horizontal, and hovering flight. In our study, the drones fly at a fixed

altitude, simplifying the problem into a 2D path-planning problem. Therefore, we consider the horizontal and turn energy.

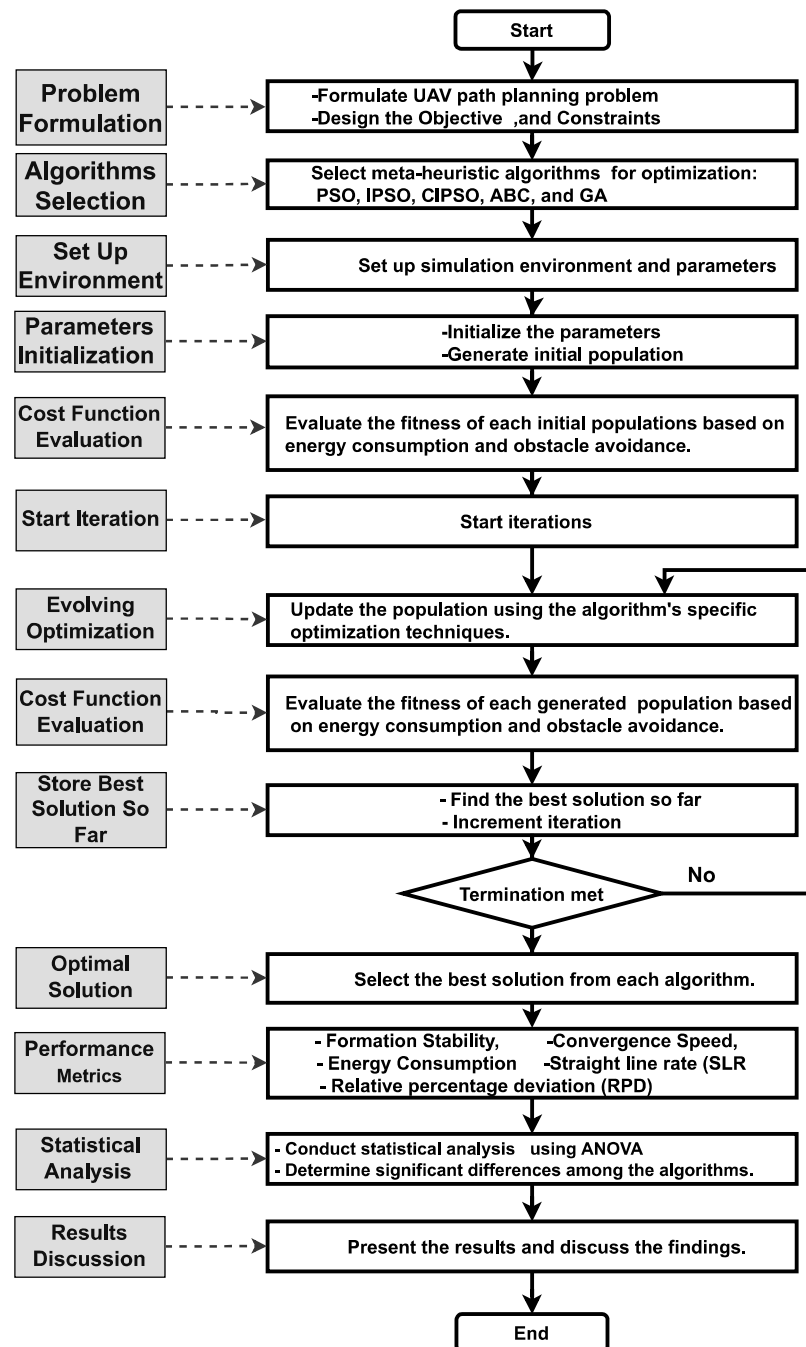


Figure 3. Flowchart of the study process.

Moving on, in each subsequent step, we apply meta-heuristic approaches. This involves setting the environment boundaries, obstacle locations, and their sizes. Next, we set the parameters of the selected algorithm and initialize its phase by initializing the parameters and populations. The population is then evaluated using the objective function. Following that, the iteration starts, and for each iteration, the population is updated based on the algorithm’s evolving process. New populations are generated and evaluated, and the best path so far is stored. The iteration continues until the maximum iteration is reached, at which point the algorithm ends, and the optimal paths for all drones are generated. These steps are repeated for all algorithms.

After collecting the results for all algorithms, we select metrics to evaluate their performance, including formation stability, convergence speed, energy consumption, straight line rate, and relative percentage deviation. Finally, we conduct a statistical analysis applying one-way ANOVA to determine significant differences between these algorithms, which are visualized in figures.

4. Result Discussion and Analysis

For fair comparison, we implement and evaluate all algorithms in Matlab on a computer featuring Core i7 CPU, @1.9 GHz (8 CPUs), 16 GB RAM. In the following subsections, we explain the relevant parameters and discuss the results.

4.1. Parameter Settings

In this subsection, we explore the parameter settings of our model. These configurations are crucial in determining the model's behavior and performance.

Formation of the IoD: Five drones navigate from their initial locations to designated destinations, as shown in Table 1.

Environment: 12 obstacles are randomly distributed over $2000\text{ m} \times 2000\text{ m}$.

The population size is configured to 250 particles, with a maximum iteration limit of 500. The number of waypoints is set to 20, and a safety distance of 20 meters is established. The probabilities of mutation (Pm) and crossover (Pc) are set to 0.005 and 0.8, respectively. In the case of the artificial bee colony (ABC) algorithm, there are 300 employed bees, 500 onlooker bees, and 300 scout bees, with a limit of 20% of maximum iteration. The parameter settings are summarized in Table 2. It is worth noting that the selection of parameters involves experimentation and fine-tuning to attain optimal performance.

Table 1. UAVs start and end points.

Drones	Start Point (x_s, y_s, z_s)	Destination Point (x_d, y_d, z_d)
Drone 1 (x, y, z)	(00, 200, 50)	(2000, 400, 50)
Drone 2 (x, y, z)	(00, 600, 50)	(2000, 800, 50)
Drone 3 (x, y, z)	(00, 1000, 50)	(2000, 12800, 50)
Drone 4 (x, y, z)	(00, 1400, 50)	(2000, 1600, 50)
Drone 5 (x, y, z)	(00, 1600, 50)	(2000, 2000, 50)

Table 2. Parameters setting.

Parameter	Vlaue
No. waypoints	20 waypoints
Popsiz	250 population
Max Iteration	500 iterations
No obstacles	12 obstacles
PSO parameters	$c_1 = 0.0596; c_2 = 2.596, w = 0.5$
CIPSO parameters	$c_{max} = 3.5, c_{min} = 0.5, w_{max} = 0.9, w_{min} = 0.4, \mu = 2, \alpha = 2$
GA parameters	$P_m = 0.005, P_C = 0.8$
ABC parameters	employed bees = 300, onlooker bees = 500, scout bees = 300
SD	20 m
No runs	30 runs

4.2. Results and Discussion

In this subsection, we discuss the simulation results, which includes analysis of the IoD path planning, the IoD path optimality, energy consumption, and execution time for the selected algorithms.

4.2.1. IoD Path Planning

The IoD is composed of multiple drones, each facing unique environmental constraints. As a result, evaluating the entire formation proves more beneficial than planning paths for individual drones. This study assesses the IoD paths using optimization algorithms. Figure 4 presents a visualization of the 2D UAV path-planning results. It illustrates the paths generated by the optimization algorithms, plotted on a 2D map to depict the planned trajectories for the drones within the formation.

The outcomes of Figure 4 illustrate that all algorithms effectively generate feasible paths for the IoD formation. Moreover, the study highlights that the IPSO algorithm yields smoother and straighter paths compared to others. This characteristic is advantageous in terms of energy efficiency and time optimization.

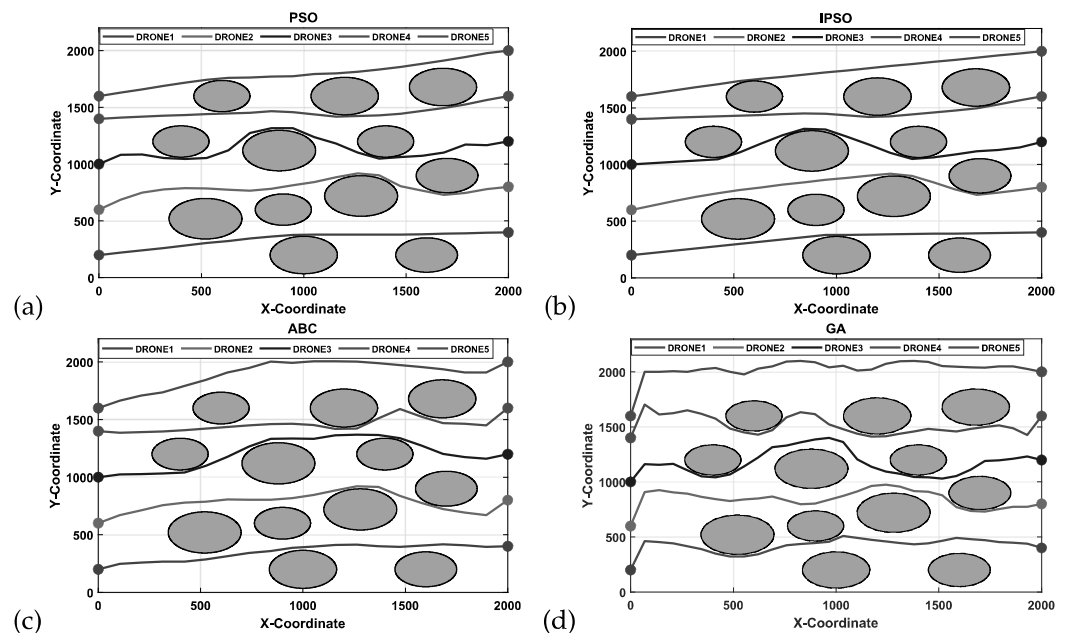


Figure 4. Two-dimensional view for path planning for the four approaches: (a) PSO, (b) IPSO, (c) ABC, and (d) GA. The drones fly from starting locations to their destinations.

4.2.2. IoD Path Optimality

A path optimality is an indication of safety and cost-effectiveness for UAV operation in the flying environment. A convergence curve, reflecting solution optimality, provides insight into solution quality. The results of convergence curves for all algorithms, averaged over 30 independent replications, are depicted in Figure 5. We observe varying energy levels for different drones, attributable to distinct environmental constraints. For instance, UAV1 and UAV5 exhibit the lowest fitness value due to light obstacles along their paths, enabling them to fly relatively far from obstacles. Conversely, UAV2, UAV3, and UAV4 encounter obstacles between their start and end points, requiring avoidance of obstacle-laden paths and incurring high costs. This illustrates that IPSO generally yields lower energy values across most UAV paths.

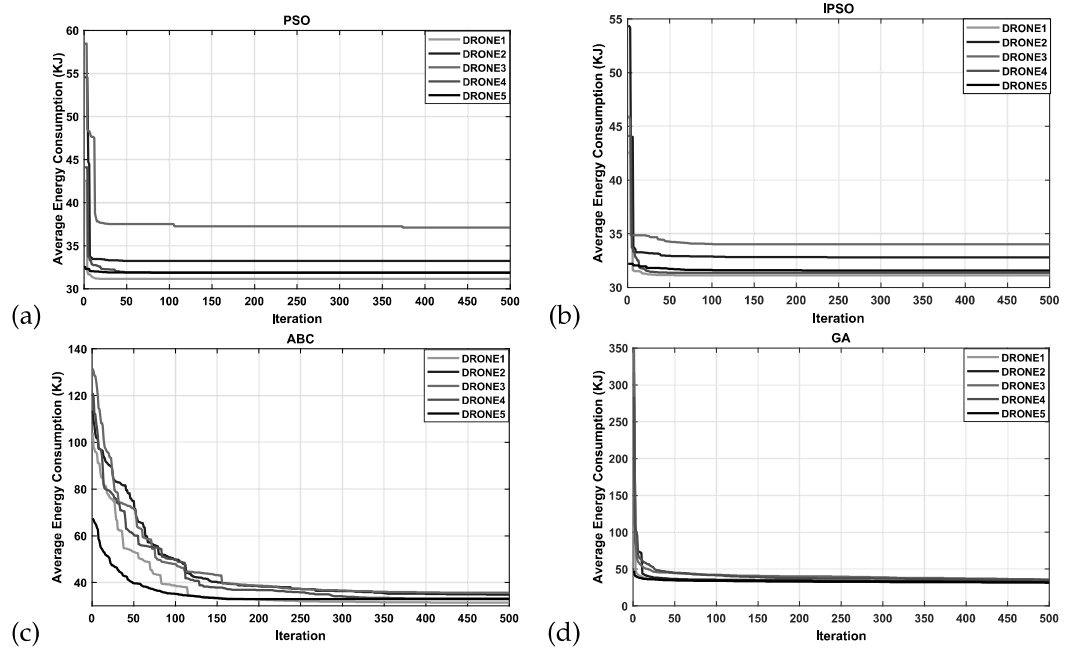


Figure 5. Comparison of the IoD path optimality and convergence curves.

4.2.3. Convergence Speed

The convergence speed is a significant factor in assessing the effectiveness of an algorithm. As we can see in Figure 5, IPSO outperforms other competitors by demonstrating faster convergence at an early stage of process evolution. This proves IPSO’s capability to generate more accurate and quicker safety paths. While GA shows faster convergence speed than ABC, it gets stuck in local optima, and the generated path tends to have higher energy consumption.

It is worth mentioning that PSO is susceptible to getting stuck in local optima and experiencing premature convergence issues. However, IPSO mitigates these problems through adaptive mutation of solutions, which strikes a balance between local and global search. This adaptive mutation approach accelerates convergence. Furthermore, if a particle becomes inactive and fails to improve the solution, it is replaced by a new particle. This replacement process enhances the diversity of the search and guides the solution towards the global optimum.

4.2.4. Energy Consumption Evaluation

The experimental results are summarized in Table 3, showcasing the worst, best, mean, and standard deviation of the average energy consumption for all algorithms. The task involves drones flying from their starting location to the destination while avoiding collision with terrain obstacles. These results represent the average of 30 independent runs, with the best value (minimum energy consumption) highlighted in bold. Observing Table 3, it is evident that IPSO outperforms other competitors across various metrics (best, mean, average, SD), demonstrating superior results.

Table 3. The worst, best, average, and standard deviation of the four heuristic approaches.

Drone	Algorithm	Best	Worst	Mean	STD
UAV1	PSO	2017.13	2026.48	2017.82	1.80282
	IPSO	2014.5	2018.32	2014.81	1.0032
	CIPSO	2015.29	2020.77	2015.71	1.24308
	ABC	2016.41	2056.93	2032.06	13.1891
	GA	2045.672	2288.015	2109.42	69.81839

Table 3. Cont.

Drone	Algorithm	Best	Worst	Mean	STD
UAV2	PSO	2014.5	2591.82	2120.04	101.653
	IPSO	2014.5	2487.12	2172.91	72.2923
	CIPSO	2014.5	2529	2151.76	84.0366
	ABC	2016.41	2977.13	2256.55	193.813
	GA	2045.672	2455.225	2226.042	83.33722
UAV3	PSO	2218.07	4356	2390.31	424.777
	IPSO	2056.41	2642.45	2200.61	86.7288
	CIPSO	2104.9	3156.52	2257.52	188.143
	ABC	2061.43	2671.54	2309.05	135.134
	GA	2096.605	2719.034	2428.615	169.0806
UAV4	PSO	2042.52	3467.21	2117.91	256.92
	IPSO	2011.98	2046.18	2023.54	10.3423
	CIPSO	1818.57	2203.93	1845.25	72.1855
	ABC	2030.07	2586.42	2154.89	161.564
	GA	2071.561	3054.177	2384.51	264.2965
UAV5	PSO	2050.73	2171.21	2067.81	26.4614
	IPSO	2040.87	2148.3	2045.79	19.8377
	CIPSO	2014.5	2518.53	2157.05	81.1005
	ABC	2043.28	2241.74	2133.88	93.6581
	GA	2051.807	2172.429	2094.52	38.80558

4.2.5. Straight Line Rate (SLR) Evaluation

For further investigation of the performance of the mentioned algorithms, we study straight line rate (SLR), which is defined by the length of the generated path over the shortest path. The shortest path is defined as the Euclidean distance from the starting point to the destination, regardless of existing obstacles.

The straight line rate can be explained as follows:

$$SLR = \frac{\sum_{i=1}^{N_w-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2}}{\sqrt{(x_{N_w} - x_1)^2 + (y_{N_w} - y_1)^2 + (z_{N_w} - z_1)^2}}, \tag{22}$$

where (x_1, y_1, z_1) and $(x_{N_w}, y_{N_w}, z_{N_w})$ are the coordinates of the starting and ending points of the path, respectively.

The lower the value of SLR, the better the performance. We use a design of experiment tool to analyze the SLR, and Minitab software is utilized to run the ANOVA test. SLR shows the indicator of the shortest path length. It is defined as the ratio of the generated path to the shortest path, as explained in Equation (22).

To further evaluate and compare the performance of algorithms, complete random design is applied to the generated data. The aim is to show if these algorithms are significantly different at a significance level of 0.05. We test the normality of data using the Anderson–Darling tool, and the results are illustrated in Figure 6.

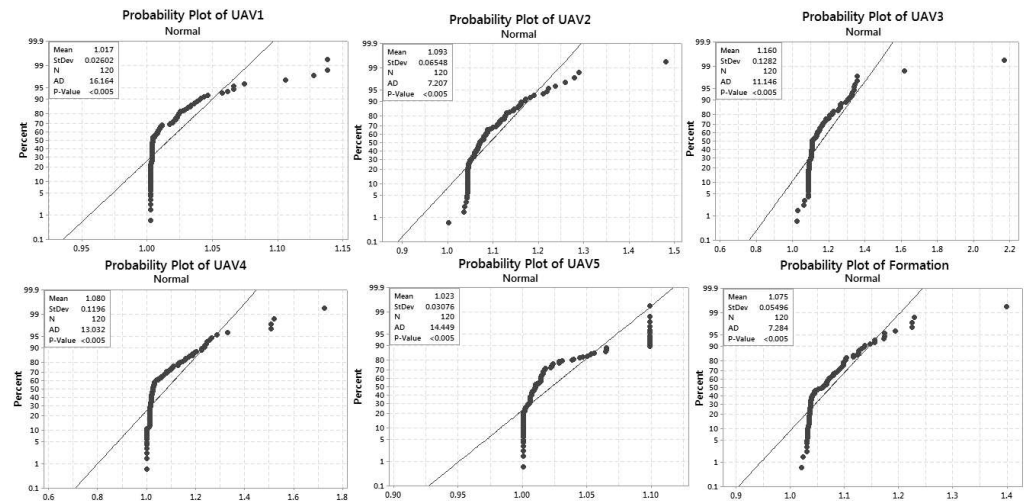


Figure 6. SLR normality test.

The p -value is less than $0.005 < \alpha$, which indicates that the data are not normal. Hence, we apply a non-parametric test, Kruskal–Wallis, with a significance level of 0.05.

To make the comparison more significant, we compare the 95% confidence intervals for all algorithms. Results are plotted in Figure 7 for all UAVs. The CIs illustrate that IPSO shows the smallest SLR, which provides the shortest path length. Thus, there is a difference between algorithms on SLR. This can be confirmed by applying the one-way ANOVA test to validate the results for different algorithms. We apply the Kruskal–Wallis test under two hypotheses. The null hypothesis assumes that there is no significant difference among algorithms, while the alternative hypothesis assumes that at least one algorithm differs from the others. The test statistic result has a p -value of zero for all UAVs, indicating that the null hypothesis is rejected at α level higher than 0.05 in a favor of the alternative hypothesis. Hence, there is a significant difference among algorithms on the SLR.

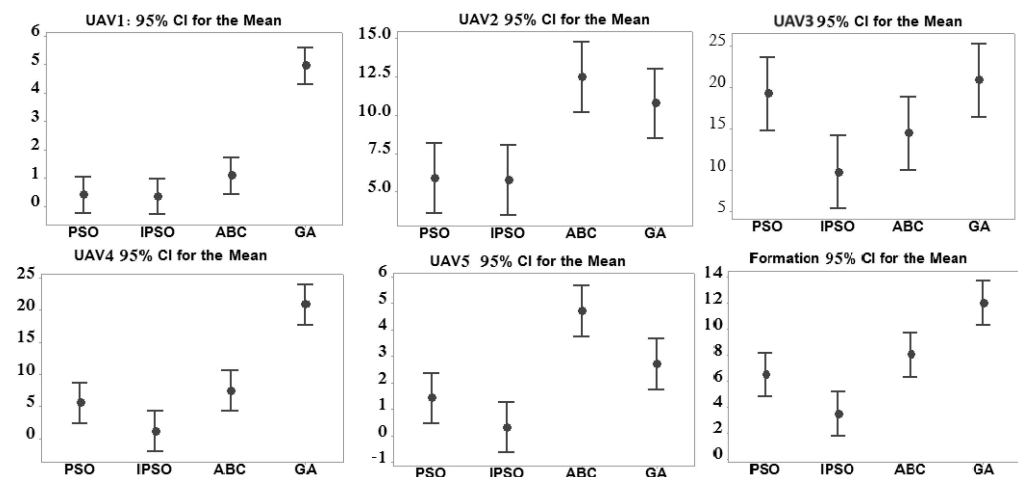


Figure 7. SLR confidence interval.

To find the difference between all algorithms, we apply a Tukey–Kramer pair-wise test to compare different algorithms. The results of different confident intervals (CIs) of the statistical test are illustrated in Figure 8.

It is worth mentioning that if different CIs include zero, it indicates that there is no significant difference between the two algorithms. Let us take UAV1 as an example. The different CIs of IPSO-PSO, IPSO-ABC, and PSO-ABC include zero, which indicates that there is no significant difference between them. However, the different CIs of GA-PSO,

GA-IPSO, and GA-ABC do not include zero, which indicates that there is a significant difference between GA and the other algorithms.

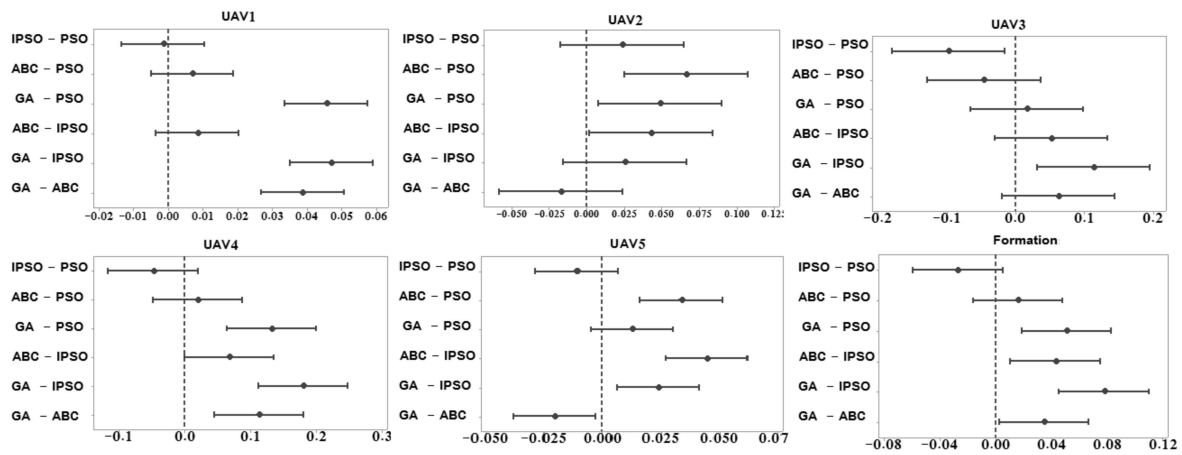


Figure 8. SLR Tukey pair-wise comparison.

4.2.6. Relative Percentage Deviation (RPD)

A relative percentage deviation represents the percentage deviation of a solution from the best one and can be defined by the following equation:

$$RPD = \frac{|E - E_{best}|}{E_{best}} \times 100 \tag{23}$$

where E is the energy required for the drone to follow the generated path from the starting point to the ending point by algorithms, and E_{best} is the energy required for the shortest path from the starting point to the ending point, regardless of obstacles. Results are obtained for 30 independent runs and RPD is calculated for all algorithms. We also apply a complete random design on the generated data. The aim is to show if these algorithms are significantly different at a significance level of 0.05. We test the normality of the data using the Anderson–Darling tool, and the results are illustrated in Figure 9.

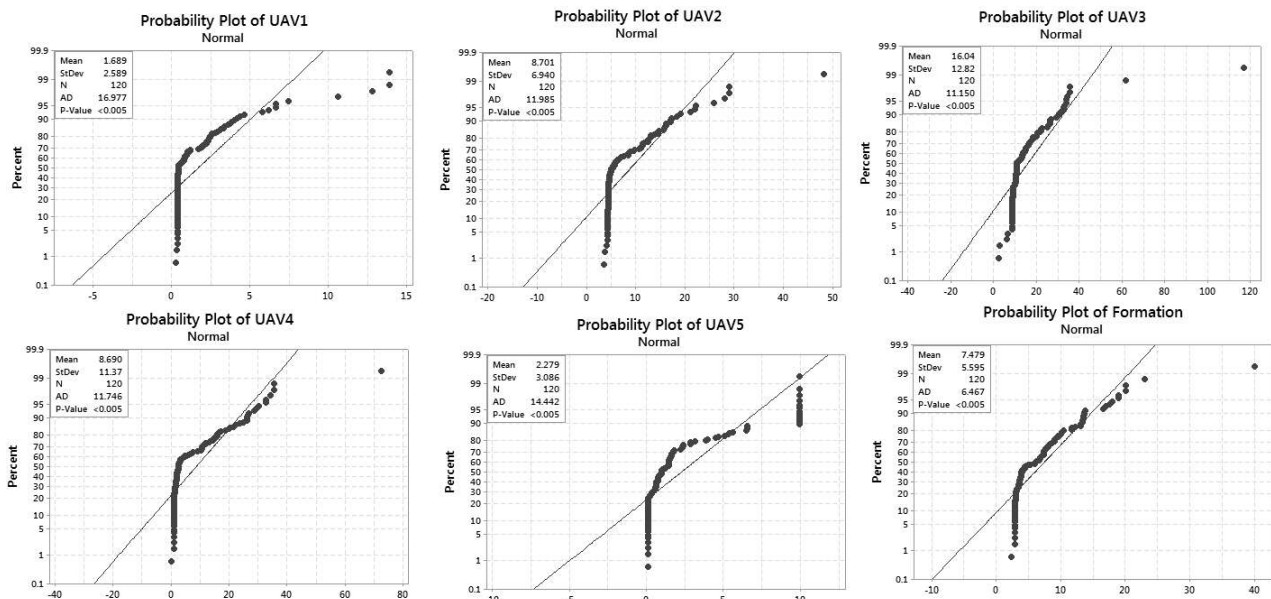


Figure 9. RPD normality test.

The p -value is less than $0.005 < \alpha$, which indicates that the data are not normal. Hence, we apply a non-parametric test, Kruskal–Wallis, with significance level of 0.05. We also compare the 95% confidence intervals for all algorithms. Results are plotted in Figure 10 for all UAVs. The CIs illustrate that IPSO shows the smallest RPD, which provides more accurate and diversity results. Thus, there is a difference between algorithms on RPD. This can be confirmed by applying the one-way ANOVA test to validate the results for different algorithms, and applying the Kruskal–Wallis test under two hypotheses. The null hypothesis assumes that there is no significant difference among the four algorithms, while the alternative hypothesis assumes that at least one algorithm differs from the others. The test statistic has a p -value of zero for all UAVs, indicating that the null hypothesis is rejected at α level higher than 0.05, in favor of the alternative hypothesis. Hence, there is a significant difference among algorithms on the RPD.

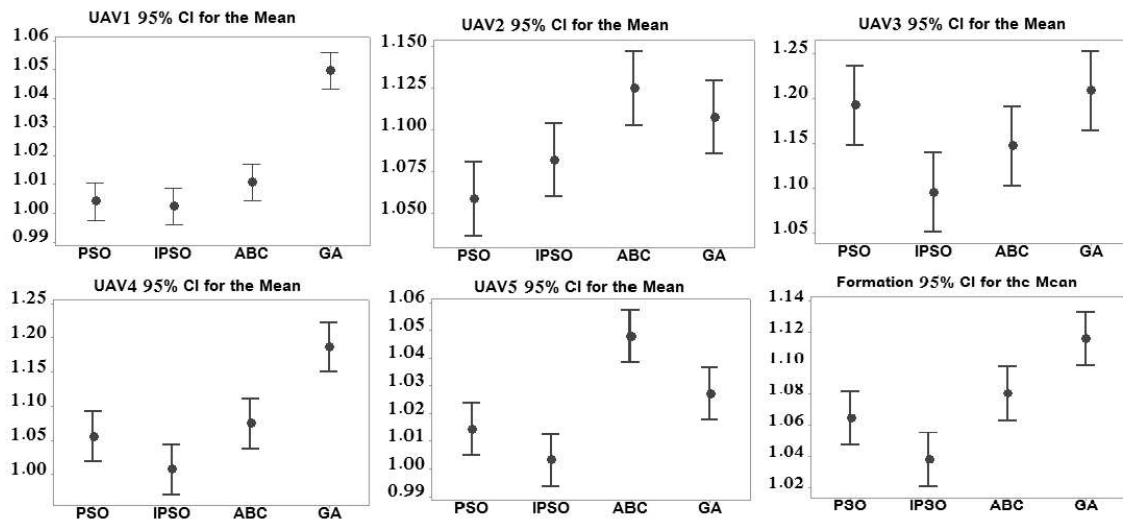


Figure 10. RPD confidence interval.

To compare different algorithms, we employ the Tukey–Kramer pair-wise test. The results of different confidence intervals (CIs) of the statistical test are illustrated in Figure 11. Taking UAV1 as an example, the different CIs of IPSO-PSO, IPSO-ABC, and PSO-ABC include zero, indicating that there is no significant difference between them. However, the different CIs of GA-PSO, GA-IPSO, and GA-ABC do not include zero, indicating a significant difference between GA and the other algorithms.

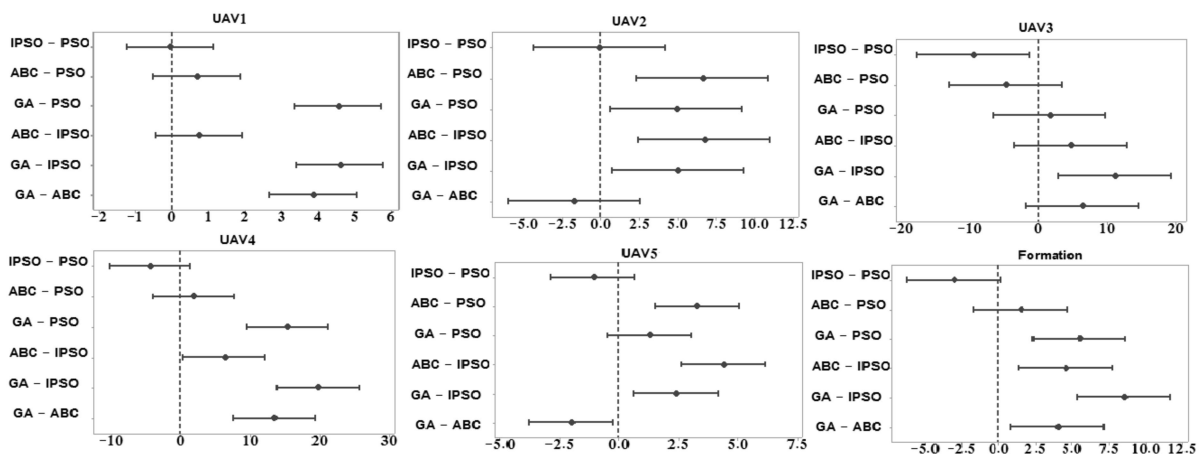


Figure 11. RPD Tukey pair-wise comparison.

It is evident that IPSO yields lower average RPD values than the other competitors for all UAVs, followed by PSO and ABC, while GA exhibits the highest average RPD values. Although IPSO and PSO closely compete in UAV1, IPSO generally demonstrates superior results.

4.2.7. IoD Formation Performance

Evaluating the IoD formation proves more effective, as it provides a comprehensive assessment of the entire group of drones within the IoD. The evaluation of the IoD formation involves calculating the average fitness value for the entire formation, which reflects the level of stability or cohesiveness achieved by the IoD formation.

Again, SLR and RPD CIs and the different CIs of the whole formation are obtained for 30 replications, and the results are shown in the sixth frame of Figures 7, 8, 10 and 11.

Furthermore, for SLR, the different CIs plotted in the sixth frame of Figure 8 indicate that IPSO and PSO are in close competition. They also indicate no significant difference between PSO and ABC. Thus, IPSO shows better SLR and provides shorter path length.

The confidence interval and difference confidence interval of RPD shown in the sixth frame of Figures 10 and 11 illustrate that IPSO outperforms the other algorithms, while PSO and ABC are in close competition.

To further investigate the performance of the presented algorithms, we compare the formation SLR and RPD for all algorithms, and the results are depicted in Figure 12.

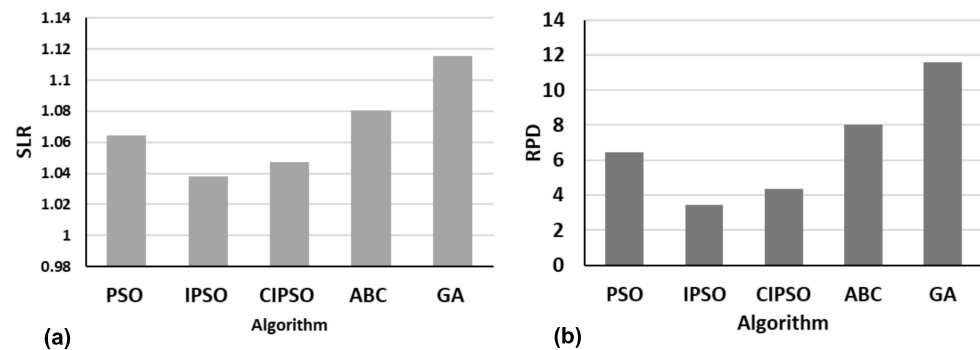


Figure 12. (a) SLR and (b) RPD of the IoD formation for all algorithms.

It can be observed that CIPSO outperforms PSO, ABC, and GA in terms of both SLR and RPD. However, when compared to IPSO, CIPSO shows slightly higher SLR and RPD. This observation suggests that IPSO-generated paths exhibit the least energy consumption, as the difference between these paths and straight lines is remarkably small, while GA shows the worst results.

Overall, the results of the IPSO algorithm consistently outperform those of the other algorithms, demonstrating superior performance for the IoD formation. This observation aligns with the findings recorded in Table 3.

4.2.8. Running Time Comparison

Running time is a crucial factor in evaluating the performance of different algorithms, as it directly impacts path generation and collision avoidance response. It represents the time required by an algorithm to generate a collision-free path, with lower execution times indicating better algorithm performance. The average execution times over 30 runs are recorded in Table 4 and depicted in Figure 13. The results indicate that PSO requires the shortest time to generate a collision-free path for the IoD, followed by ABC, CIPSO, and IPSO, while GA exhibits longer execution times.

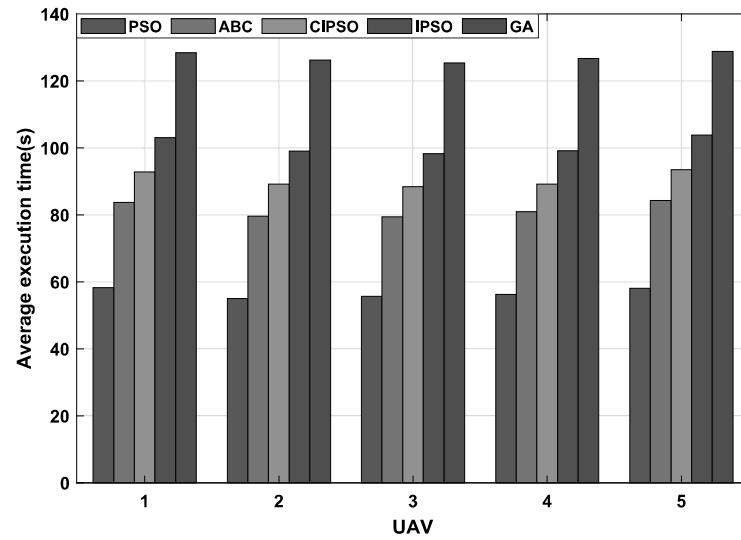


Figure 13. Average execution time.

Table 4. Average execution time.

Algorithm	PSO	ABC	CIPSO	IPSO	GA
UAV1	58.3129	83.7895	92.8117	103.124	128.492
UAV2	55.0654	79.6591	89.2066	99.1185	126.294
UAV3	55.7387	79.4361	88.4537	98.2819	125.378
UAV4	56.3328	80.9401	89.2281	99.1423	126.707
UAV5	58.1598	84.3046	93.4992	103.888	128.856

4.2.9. Scalability of IPSO in Multi-UAV Path-Planning

To investigate the scalability of IPSO, we conducted experiments in a large scenario by increasing the number of UAVs to 18 and the number of obstacles to 24. The objective is to assess how the performance of the algorithms is influenced as the scale of the problem increases. As shown in Figure 14, IPSO showcases its effectiveness in handling larger problem sizes efficiently.

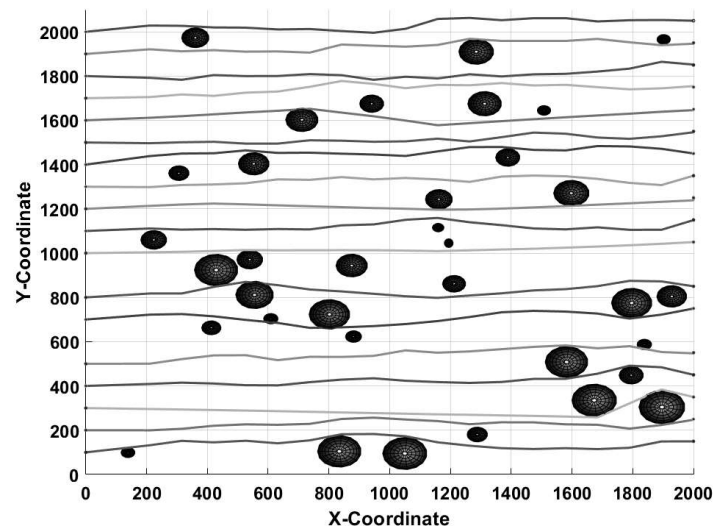


Figure 14. Two-dimensional view of paths for 18 UAVs navigating in environment filled with 24 obstacles.

5. Discussion and Remarks

This study highlights the importance of efficient collision avoidance and energy-efficient strategies for UAV path planning. The application of meta-heuristic algorithms offers a promising approach to address these challenges and optimize the performance of UAVs in obstacle-rich environments. The simulations focus on evaluating the effectiveness of these algorithms in terms of the IoD formation stability, convergence speed, path length efficiency, and energy consumption.

The results of the simulations reveal that IPSO outperforms the other algorithms in several key aspects. IPSO demonstrates superior IoD formation stability, ensuring that the networked UAV system maintains a stable and reliable configuration throughout the mission. This stability is crucial for maintaining effective communication, coordination, and cooperation among the drones. Additionally, IPSO exhibits faster convergence speed compared to PSO and ABC. Convergence speed refers to the rate at which the algorithm reaches the optimal solution. The faster convergence of IPSO implies that it requires fewer iterations to find the optimal or the near-optimal path for the IoD. This efficiency is vital for real-time applications where quick decision making and responsiveness are critical. However, it is worth noting that IPSO requires a longer run time compared to PSO and ABC. This trade-off between performance and computational time should be carefully considered when selecting the most suitable algorithm for a specific application.

While ABC has been successfully applied to various optimization problems, its exploration-focused nature and sensitivity to parameter settings are important considerations. To overcome these limitations associated with its exploration–exploitation balance, it is possible to carefully design and fine-tune the algorithm to ensure optimal performance and overcome the limitations.

One of the main limitations of the genetic algorithm is its susceptibility to premature convergence. GA's performance can be affected by the choice of its genetic operators and their parameter settings. Improper selection or tuning of crossover and mutation rates can lead to ineffective exploration or exploitation of the search space, hindering the algorithm's ability to find high-quality solutions.

The comparison and analysis of the results illustrated above provide valuable insights into the strengths and limitations of each algorithm for UAV path planning in obstacle-dense environments. These findings contribute to the existing body of knowledge and shed light on the selection and implementation of meta-heuristic algorithms for optimizing the path planning of UAVs in practical scenarios.

These algorithms can be improved to solve the IoD path-planning problem using different strategies, including enhancing the algorithms' parameters with adaptive learning for dynamic environments, exploring hybrid algorithms that balance path optimization with execution speed, and applying these algorithms in 3D path planning and real-world scenarios. Additionally, the hybridization of two algorithms, such as IPSO and ABC, combines the strengths of both algorithms.

The main limitation of the study is that it assumes that all drones are flying at a fixed altitude, and that the problem is simplified into 2D path planning, which may not fully represent real-world scenarios and lacks consideration of dynamic environments. Additionally, the study does not explore hybrid algorithms that balance path optimization with execution speed. Furthermore, the research does not address 3D path planning and real-world applications of the algorithms.

6. Conclusions

In this study, we employed five meta-heuristic approaches to obtain optimal paths for the IoD, conducting extensive simulations and analyzing the results for comparison. Our results showed that IPSO could generate smoother paths with minimum energy consumption, yielding best, worst, mean, and standard deviation values. These results signify a notable advancement in the operational efficiency and sustainability of UAV missions, highlighting the potential of IPSO in complex path-planning scenarios. Furthermore, the

exploration of the straight line rate and relative percentage deviation provided crucial insights into the path length and energy consumption optimization capabilities of these algorithms. Despite close competition between IPSO and PSO in certain scenarios, our rigorous statistical analysis, encompassing non-parametric tests and ANOVA, affirmed IPSO's dominance, particularly in the overall IoD formation performance. This underlines the algorithm's robustness in maintaining high stability within the IoD formations. Finally, the execution time for each algorithm was evaluated and compared. The results indicated that PSO was more time-efficient in generating the IoD paths compared to the others, whereas GA took a longer time to accomplish the same task.

In future research, we plan to enhance IPSO with adaptive learning mechanisms tailored for dynamic environments. Additionally, we aim to explore hybrid algorithms that strike a balance between path optimization and execution speed, as well as apply these algorithms in 3D path planning and real-world scenarios. These endeavors will further advance UAV navigation efficiency within the IoD context.

Author Contributions: Conceptualization, G.A. and T.S.; methodology and Software, M.G. and M.H.; writing—original draft preparation, G.A, T.S. and A.M. formal analysis, writing—review and editing A.Y.; investigation. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Project Number INML2104 under the interdisciplinary 619 center of smart mobility and logistics at King Fahd University of Petroleum and Minerals.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Acknowledgments: The authors would like to acknowledge the support of the Computer Engineering Department at King Fahd University of Petroleum and Minerals for the support of this work.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Sabino, S.; Grilo, A. Topology control of unmanned aerial vehicle (uav) mesh networks: A multi-objective evolutionary algorithm approach. In Proceedings of the 4th ACM Workshop on Micro Aerial Vehicle Networks, Systems, and Applications, New York, NY, USA, 10–15 June 2018; pp. 45–50.
- Sombolestan, S.; Rasooli, A.; Khodaygan, S. Optimal path-planning for mobile robots to find a hidden target in an unknown environment based on machine learning. *J. Ambient. Intell. Humaniz. Comput.* **2019**, *10*, 1841–1850. [CrossRef]
- Poudel, S.; Arafat, M.Y.; Moh, S. Bio-Inspired Optimization-Based Path Planning Algorithms in Unmanned Aerial Vehicles: A Survey. *Sensors* **2023**, *23*, 3051. [CrossRef]
- Ahmed, G.; Sheltami, T. A Safety System For Maximizing Operated UAVs Capacity Under Regulation Constraints. *IEEE Access* **2023**, *11*, 139069–139081. [CrossRef]
- Vo, T.M.N.; Wang, C.N.; Yang, F.C.; Singh, M. Internet of Things (IoT): Wireless Communications for Unmanned Aircraft System. *Eurasia Proc. Sci. Technol. Eng. Math.* **2023**, *23*, 388–399. [CrossRef]
- Chen, M.; Sharma, A.; Bhola, J.; Nguyen, T.V.; Truong, C.V. Multi-agent task planning and resource apportionment in a smart grid. *Int. J. Syst. Assur. Eng. Manag.* **2022**, *13*, 444–455. [CrossRef]
- Sheltami, T.; Ahmed, G.; Yasar, A. An Optimization Approach of IoD Deployment for Optimal Coverage Based on Radio Frequency Model. *CMES-Comput. Model. Eng. Sci.* **2024**, *139*, 2627–2647. [CrossRef]
- Abbas, N.; Abbas, Z.; Liu, X.; Khan, S.S.; Foster, E.D.; Larkin, S. A survey: Future smart cities based on advance control of Unmanned Aerial Vehicles (UAVs). *Appl. Sci.* **2023**, *13*, 9881. [CrossRef]
- Yan, Y.; Sun, Z.; Hou, Y.; Zhang, B.; Yuan, Z.; Zhang, G.; Wang, B.; Ma, X. UAV Swarm Mission Planning and Load Sensitivity Analysis Based on Clustering and Optimization Algorithms. *Appl. Sci.* **2023**, *13*, 12438. [CrossRef]
- Ghambari, S.; Lepagnot, J.; Jourdan, L.; Idoumghar, L. A comparative study of meta-heuristic algorithms for solving UAV path planning. In Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence (SSCI), Bangalore, India, 18–21 November 2018; pp. 174–181.
- Rezwan, S.; Choi, W. Artificial Intelligence Approaches for UAV Navigation: Recent Advances and Future Challenges. *IEEE Access* **2022**, *10*, 26320–26339. [CrossRef]
- Li, J.; Kang, F.; Chen, C.; Tong, S.; Jia, Y.; Zhang, C.; Wang, Y. The Improved A* Algorithm for Quadrotor UAVs under Forest Obstacle Avoidance Path Planning. *Appl. Sci.* **2023**, *13*, 4290. [CrossRef]

13. Gupta, M.; Varma, S. Optimal placement of UAVs of an aerial mesh network in an emergency situation. *J. Ambient. Intell. Humaniz. Comput.* **2021**, *12*, 343–358. [CrossRef]
14. Ahmed, G.A.; Sheltami, T.R.O.; Mahmoud, A.S.; Yasar, A. 3D Simulation Model for IoD-to-Vehicles Communication in IoD-assisted VANET. *Front. Built Environ.* **2023**, *9*, 1–18. [CrossRef]
15. Zhao, J.; Deng, C.; Yu, H.; Fei, H.; Li, D. Path planning of unmanned vehicles based on adaptive particle swarm optimization algorithm. *Comput. Commun.* **2024**, *216*, 112–129. [CrossRef]
16. Yahia, H.S.; Mohammed, A.S. Path planning optimization in unmanned aerial vehicles using meta-heuristic algorithms: A systematic review. *Environ. Monit. Assess.* **2023**, *195*, 30. [CrossRef]
17. Besada-Portas, E.; de la Torre, L.; Jesus, M.; de Andrés-Toro, B. Evolutionary trajectory planner for multiple UAVs in realistic scenarios. *IEEE Trans. Robot.* **2010**, *26*, 619–634. [CrossRef]
18. Zheng, C.; Li, L.; Xu, F.; Sun, F.; Ding, M. Evolutionary route planner for unmanned air vehicles. *IEEE Trans. Robot.* **2005**, *21*, 609–620. [CrossRef]
19. Belge, E.; Altan, A.; Hacıoğlu, R. Metaheuristic Optimization-Based Path Planning and Tracking of Quadcopter for Payload Hold-Release Mission. *Electronics* **2022**, *11*, 1208. [CrossRef]
20. Xu, Y.; Han, Y.; Sun, Z.; Gu, W.; Jin, Y.; Xue, X.; Lan, Y. Path Planning Optimization With Multiple Pesticide and Power Loading Bases Using Several Unmanned Aerial Systems on Segmented Agricultural Fields. *IEEE Trans. Syst. Man, Cybern. Syst.* **2022**, *3*, 1882–1894. [CrossRef]
21. Ahmed, G.; Sheltami, T.; Deriche, M.; Yasar, A. An energy efficient IoD static and dynamic collision avoidance approach based on gradient optimization. *Ad Hoc Netw.* **2021**, *118*, 102519. [CrossRef]
22. Qin, H.; Shao, S.; Wang, T.; Yu, X.; Jiang, Y.; Cao, Z. Review of autonomous path planning algorithms for mobile robots. *Drones* **2023**, *7*, 211. [CrossRef]
23. Ma, Y.; Zamirian, M.; Yang, Y.; Xu, Y.; Zhang, J. Path planning for mobile objects in four-dimension based on particle swarm optimization method with penalty function. *Math. Probl. Eng.* **2013**, *2013*, 613964. [CrossRef]
24. Ni, X.; Hu, W.; Fan, Q.; Cui, Y.; Qi, C. A Q-learning based multi-strategy integrated artificial bee colony algorithm with application in unmanned vehicle path planning. *Expert Syst. Appl.* **2024**, *236*, 121303. [CrossRef]
25. Lee, J.J.; Rathinam, S. A Meta-Heuristic Approach for an Aerial-Ground Vehicle Path Planning Problem. In Proceedings of the AIAA SCITECH 2024 Forum, Orlando, FL, USA, 8–12 January 2024; p. 0230.
26. Wang, H.; Song, S.; Guo, Q.; Xu, D.; Zhang, X.; Wang, P. Cooperative Motion Planning for Persistent 3D Visual Coverage with Multiple Quadrotor UAVs. *IEEE Trans. Autom. Sci. Eng.* **2023**. [CrossRef]
27. Ma, Y.; Hu, M.; Yan, X. Multi-objective path planning for unmanned surface vehicle with currents effects. *ISA Trans.* **2018**, *75*, 137–156. [CrossRef]
28. YongBo, C.; YueSong, M.; JianQiao, Y.; XiaoLong, S.; Nuo, X. Three-dimensional unmanned aerial vehicle path planning using modified wolf pack search algorithm. *Neurocomputing* **2017**, *266*, 445–457. [CrossRef]
29. Ahmed, G.; Sheltami, T.; Mahmoud, A.; Yasar, A. Energy-Efficient UAVs Coverage Path Planning Approach. *CMES-Comput. Model. Eng. Sci.* **2023**, *136*, 3239–3263. [CrossRef]
30. Li, M.; Cheng, N.; Gao, J.; Wang, Y.; Zhao, L.; Shen, X. Energy-efficient UAV-assisted mobile edge computing: Resource allocation and trajectory optimization. *IEEE Trans. Veh. Technol.* **2020**, *69*, 3424–3438. [CrossRef]
31. Ji, X.; Meng, X.; Wang, A.; Hua, Q.; Wang, F.; Chen, R.; Zhang, J.; Fang, D. E2PP: An Energy-Efficient Path Planning Method for UAV-Assisted Data Collection. *Secur. Commun. Netw.* **2020**, *2020*, 8850505. [CrossRef]
32. Ahmed, G.; Sheltami, T.; Mahmoud, A.; Yasar, A. IoD swarms collision avoidance via improved particle swarm optimization. *Transp. Res. Part A Policy Pract.* **2020**, *142*, 260–278. [CrossRef]
33. Yan, S.; Peng, M.; Cao, X. A game theory approach for joint access selection and resource allocation in UAV assisted IoT communication networks. *IEEE Internet Things J.* **2018**, *6*, 1663–1674. [CrossRef]
34. Li, B.; Fei, Z.; Zhang, Y. UAV communications for 5G and beyond: Recent advances and future trends. *IEEE Internet Things J.* **2018**, *6*, 2241–2263. [CrossRef]
35. Mozaffari, M.; Saad, W.; Bennis, M.; Debbah, M. Wireless communication using unmanned aerial vehicles (UAVs): Optimal transport theory for hover time optimization. *IEEE Trans. Wirel. Commun.* **2017**, *16*, 8052–8066. [CrossRef]
36. Hu, Y.; Yang, S.X. A knowledge based genetic algorithm for path planning of a mobile robot. In Proceedings of the IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004, New Orleans, LA, USA, 26 April–1 May 2004; Volume 5, pp. 4350–4355.
37. Ismail, A.; Sheta, A.; Al-Weshah, M. A mobile robot path planning using genetic algorithm in static environment. *J. Comput. Sci.* **2008**, *4*, 341–344.
38. Eberhart, R.; Kennedy, J. A new optimizer using particle swarm theory. In Proceedings of the MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 4–6 October 1995; pp. 39–43.
39. Coloni, A.; Dorigo, M.; Maniezzo, V. Distributed optimization by ant colonies. In Proceedings of the First European Conference on Artificial Life, Cambridge, MA, USA, 11–13 December 1991; Volume 142, pp. 134–142.
40. Shao, S.; Peng, Y.; He, C.; Du, Y. Efficient path planning for UAV formation via comprehensively improved particle swarm optimization. *ISA Trans.* **2020**, *97*, 415–430. [CrossRef] [PubMed]

41. Pehlivanoglu, Y.V.; Pehlivanoglu, P. An enhanced genetic algorithm for path planning of autonomous UAV in target coverage problems. *Appl. Soft Comput.* **2021**, *112*, 107796. [CrossRef]
42. Yu, W.; Liu, J.; Zhou, J. A novel sparrow particle swarm algorithm (SPSA) for unmanned aerial vehicle path planning. *Sci. Program.* **2021**, *2021*, 5158304. [CrossRef]
43. Yang, L.; Qi, J.; Xiao, J.; Yong, X. A literature review of UAV 3D path planning. In Proceedings of the 11th World Congress on Intelligent Control and Automation, Shenyang, China, 29 June–4 July 2014; pp. 2376–2381.
44. Zhao, Y.; Zheng, Z.; Liu, Y. Survey on computational-intelligence-based UAV path planning. *Knowl.-Based Syst.* **2018**, *158*, 54–64. [CrossRef]
45. Qadir, Z.; Ullah, F.; Munawar, H.S.; Al-Turjman, F. Addressing disasters in smart cities through UAVs path planning and 5G communications: A systematic review. *Comput. Commun.* **2021**, *168*, 114–135. [CrossRef]
46. Pehlivanoglu, Y.V. A new vibrational genetic algorithm enhanced with a Voronoi diagram for path planning of autonomous UAV. *Aerosp. Sci. Technol.* **2012**, *16*, 47–55. [CrossRef]
47. Chen, Y.; Yu, J.; Mei, Y.; Wang, Y.; Su, X. Modified central force optimization (MCFO) algorithm for 3D UAV path planning. *Neurocomputing* **2016**, *171*, 878–888. [CrossRef]
48. Phung, M.D.; Ha, Q.P. Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization. *Appl. Soft Comput.* **2021**, *107*, 107376. [CrossRef]
49. Ou, X.; Liu, Y.; Zhao, Y. PSO based UAV online path planning algorithms. In Proceedings of the 2017 International Conference on Automation, Control and Robots, Wuhan, China, 22–25 December 2017; pp. 41–45.
50. Roberge, V.; Tarbouchi, M.; Labonté, G. Fast genetic algorithm path planner for fixed-wing military UAV using GPU. *IEEE Trans. Aerosp. Electron. Syst.* **2018**, *54*, 2105–2117. [CrossRef]
51. Huang, C.; Fei, J. UAV path planning based on particle swarm optimization with global best path competition. *Int. J. Pattern Recognit. Artif. Intell.* **2018**, *32*, 1859008. [CrossRef]
52. Shin, J.J.; Bang, H. UAV path planning under dynamic threats using an improved PSO algorithm. *Int. J. Aerosp. Eng.* **2020**, *2020*, 8820284. [CrossRef]
53. Di Franco, C.; Buttazzo, G. Coverage path planning for UAVs photogrammetry with energy and resolution constraints. *J. Intell. Robot. Syst.* **2016**, *83*, 445–462. [CrossRef]
54. Karaboga, D. *An Idea Based on Honey Bee Swarm for Numerical Optimization*; Technical Report-tr06; Erciyes University: Kayseri, Turkey, 2005.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.