Comparative Analysis of Algorithms and Applications in Unmanned Aerial Vehicle Path Planning in Recent Years

Dongsheng Hu[1, *], Yu Kuang[2] and Taiyao Zhang[3]

[1]Leicester International Institute, Dalian University of Technology, Dalian 200093, China
[2]Sino-German College, Shanghai University of Science and Technology, Shanghai 116081 China
[3]Hainan Vocational University of Science and Technology, Haikou 571126, China
*Corresponding email: 1969230391@mail.dlut.edu.cn

Abstract. The research of UAVs(Unmanned Aerial Vehicle) has received a lot of attention in recent years, and more and more UAV manufacturers have emerged in China and abroad, such as DJI, 3D Robotics, Parrot and so on. The advantage of UAVs over manned vehicles is that they can perform missions in dangerous and complex environments, such as fires, battlefields, earthquake debris, etc. With the increase in automation and intelligence of UAVs, UAV technology has been developed as never before, and it will play an increasingly important role in the future military and civilian sectors. On the military side, UAVs can perform military tasks such as reconnaissance, attack, and electronic countermeasures. On the civilian side, UAVs can be used for weather detection, pesticide spraying, geological surveys, disaster relief and many other areas. The path planning system plays a crucial role when UAVs perform these tasks. Due to the fuel limitation of UAVs, how to avoid obstacles and choose the proper route in a reasonable and safe manner, and even how to deal with unexpected situations, become the focus of UAV path planning. In recent years, many new path planning algorithms have been proposed to accomplish these increasingly complex tasks. Then, the advantages and disadvantages of GA(Genetic Algorithm), ACO(Ant Colony Optimization), and PSO(Particle Swarm Optimization)are discussed, and the future development direction and trend of UAV path planning is outlined.

Keywords: UAV, Path planning, GA, ACO, PSO

1 Introduction

P. Kar et al. (eds.), *Proceedings of the 2023 International Conference on Image, Algorithms and Artificial Intelligence (ICIAAI 2023)*, Advances in Computer Science Research 108,
https://doi.org/10.2991/978-94-6463-300-9_44

In the past few years, optimization algorithms such as GA, ACO, and PSO have been widely used in different fields to solve various complex problems. Because of their ability to find near-optimal solutions in a short time, these algorithms have shown promising results in optimization tasks.

Previous research has extensively explored and modified these optimization algorithms to suit specific application domains, but some challenges still persist. the existing research needs to address issues such as algorithmic efficiency, convergence speed, and accuracy in producing optimal solutions. in addition, researchers have not yet fully explored the potential and limits of these optimization algorithms in certain application domains.

This paper will discuss and analyze the effectiveness, limitations, and challenges of ga, aco, and pso. furthermore, this study aims to investigate how combining these algorithms can lead to better optimization performance than using any one algorithm on its own. specifically, we will focus on the application of these algorithms in solving the traveling salesman problem, which is a well-known combinatorial optimization problem with great importance in logistics and transportation.

The significance of this research includes discovering new methods for solving real-world problems faster and with greater accuracy. by using hybrid algorithms and studying their performance on tsp, they also present a new perspective on the use of optimization approaches in other domains [1, 2].

2 Related works

2.1 Genetic Algorithms

Background of the algorithm. Genetic algorithms (GA) are a widely used method in various areas, including optimization and prediction. GA is inspired by the process of natural selection and genetics, simulating darwin's survival of the fittest theory. the algorithm maintains a population of individuals represented as chromosomes that can exchange genetic material through crossover and mutation until the fittest individual, or solution, is found.

Recent studies have shown the effectiveness and versatility of GA in various applications. For example, utilized ga to identify genes associated with systemic lupus erythematosus (SLE). in their study, they conducted a genome-wide association study

and processed the data using ga to filter out irrelevant genes and improve the accuracy of their findings [3].

Additionally, Xia et al. (2020) developed a GA-based predictive model for predicting the presence of prostate cancer in a given patient. their approach showed high accuracy compared to traditional models and could provide a valuable tool for cancer screening and diagnosis.

In conclusion, genetic algorithms are a powerful tool in many fields such as biology, engineering, and computer science, among others. as demonstrated by the studies mentioned above, ga has the potential to enhance predictions, discoveries, and optimizations in various domains [4].

Algorithm Logic. The main reason for this is that the solution to the problem is represented by a sequence of numbers, and the genetic operator works directly on this sequence. Coding can be divided into binary and real coding. If an individual is represented by a binary coding, the decoding formula can be used to convert a binary number into a decimal number.

$$F(b_{i1}, b_{i2}, \ldots, b_{il}) = R_i + \frac{T_i - R_i}{2^l - 1} \sum_{j=1}^{l} b_{ij} 2^{j-1} \quad (1)$$

Algorithm Flow. (1) Initialisation: a population is selected, i.e., a set of strings or individuals is selected. This initial group is also the set of problem hypotheses. This set of strings or individuals is usually generated randomly. The optimal solution to the problem is found by evolving the solution to these initial hypotheses [5].

(2) Selection: Selection is the selection of good individuals from a population and the elimination of bad individuals. It is based on an assessment of fitness. The higher the fitness, the higher the probability that an individual will be selected and its number of "offspring" in the next generation will be greater, and the selected individuals are included in a pool of pairs. The most commonly used selection methods are roulette, retention of the best individuals, expectation, ranked selection, competition, and linear normalization.

(3) Crossover: The process of replacing and recombining parts of the structure of two parent individuals to create a new individual. It is an important tool in genetic algorithms for generating good individuals. Crossover involves randomly selecting two individuals from a matching pool based on some crossover probability, where the crossover position is also random, and the crossover probability is typically large, between 0.6 and 0.9. For the next generation of individuals selected for propagation,

two individuals are randomly selected at the same location, and the exchange occurs at the selected location according to the crossover probability P. The crossover probability is usually large, and the crossover probability is usually small. The Figure 1 showed the crossing of two parent individuals.

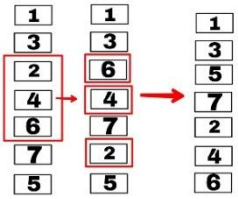

Figure 1. The crossing of two parent individuals

(4) Mutation: According to the principle of genetic variation in biological inheritance, the probability of mutation in certain individuals is Pm. The mutation probability Pm is consistent with the fact that biological variability is very low, so the value of Pm is very low, usually between 0.01 and 0.2. The Figure 2 showed the mutation of an individual.

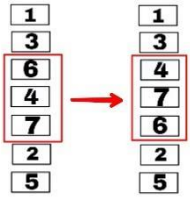

Figure 2. The mutation of an individual

(5) In this case, an optimal solution is found by iteratively searching for an optimal solution, which is either an optimal solution or an optimal population. Otherwise, the previous generation is replaced by a new iterative gene obtained by cross selection, and the cycle continues, returning to the stepwise selection process (2). The flowchart of the GA was shown in the Figure 3.

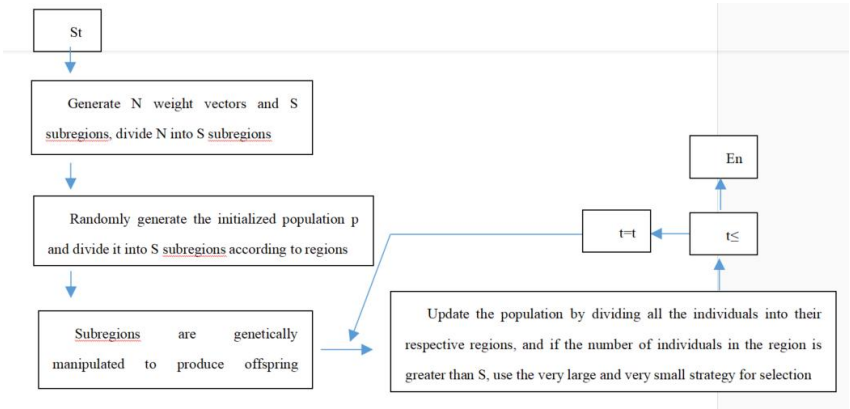

Figure 3. The flowchart of the genetic algorithm

Genetic algorithms (GA) have been widely used in unmanned aerial vehicle (UAV) path planning due to their effectiveness and robustness. In the past three years, several studies have used GA to optimize UAV paths in different scenarios.

For example, a study by Liu et al. (2018) proposed a ga-based algorithm for uav surveillance missions, where the objective is to cover as much area as possible while avoiding obstacles. the algorithm optimizes the uav's trajectory by considering both coverage and obstacle avoidance, resulting in efficient and safe surveillance [6].

Another study by Chen et al. (2019) applied ga to uav delivery routes optimization, aiming to minimize the delivery time and cost. the algorithm takes into account various factors such as wind speed, payload weight, and altitude, and generates the optimal route accordingly [7].

Furthermore, a recent study by Li et al. (2020) proposed an adaptive ga-based algorithm for uav search and rescue operations. the algorithm adapts its parameters according to the search environment, which improves the efficiency and accuracy of the search process [8].

In conclusion, ga has proven to be a powerful tool in UAV path planning and optimization, and its applications continue to expand in various fields.

2.2 Ant Colony Optimization (ACO)

Background. It mimics the decision-making process of ants searching for Mood by following their scent trails to find the shortest path to food. Researchers have used this algorithm to solve a variety of problems, including the traveling salesman problem, the vehicle routing problem, and the resource allocation problem. In addition,

integrating the algorithm with other meta-heuristic methods, such as genetic algorithms and particle swarm optimization, has been successful in many real-world applications.

Recent research has focused on improving the ant colony algorithm's performance and efficiency. One study by guo et al. (2020) proposed a modified ant colony algorithm using local search methods and dynamic clustering of ants to enhance the algorithm's ability to explore different solutions. another study by mohammadzadeh et al. (2019) utilized an improved ant colony algorithm with a self-adaptive approach to enhance the algorithm's stability and convergence rate [9, 10].

In conclusion, the ant colony algorithm continues to be a promising optimization methodology that can address various real-world problems effectively. ongoing research will undoubtedly lead to further improvements and innovation in this field.

Algorithm Logic. The ant colony algorithm refers to an iterative algorithm in which in each iteration, each ant walks from one point to another to build a route but cannot return again to the point it has already reached. In each path, the ants will determine the next point to be reached by more pheromones, or randomly if it is a city not reached by other ants. At the end of the iteration, the final best path is determined by modifying the values of the pheromones in it based on the paths constructed by the ants [11].

Algorithm flow:

Assume that ants have the following characteristics

1) Ants do not visit the same city in another journey (see Figure 4).

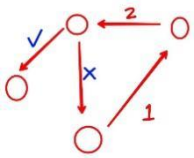

Figure 4. Not visit the same city in another journey

2) Ants can know the distance between cities and will prefer to go to the nearest city if other conditions are equal (see Figure 5).

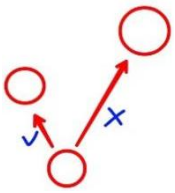

Figure 5. Chosen to nearest city

3 Ants will release pheromones during their journey and will prefer the road with higher concentration of pheromones if the distance is the same (see Figure 6).

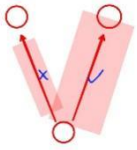

Figure 6. Chosen to higher concentration of pheromones

(The area of the light red part= concentration of pheromones)

The probability that ant k chooses city j as its next destination from city i is determined by the following equation:

$$P_k^{ij} = \frac{(\tau_{ij}^{\alpha})(\eta_{ij}^{\beta})}{\sum_{z \in allowedx} (\tau_{ij}^{\alpha})(\eta_{ij}^{\beta})} \tag{2}$$

Where (τ_{ij}^{α}) is the value of flomon concentration and his evolution equation:

$$\tau_{ij}(t+1) = \rho * \tau_{ij} + \triangle \tau \tag{3}$$

That is determined by the volatility coefficient rho of pheromone and the change value of pheromone after the last cycle $\triangle\tau$, the volatility coefficient is a constant Δ number, the $\triangle \tau$ value of path ij is determined by the systematic constant Q and the total distance traveled by this ant in the last cycle, the shorter the total distance, the larger the delta tao will be.

If a total of m ants traveled through ij two cities in the last period, each ant will bring an increment of delta tao's pheromone concentration, according to which we can update the pheromone concentration of ij path tao:

$$\triangle \tau_k^{ij} = \sum_{k=1}^{m} \triangle \tau_k^{ij} \tag{4}$$

(n_{ij}^{β}) is the visibility value, it is the reciprocal of the distance between cities ij, the smaller the distance between two cities, the larger the eta value.

α and β beta are the controlling coefficients of the Fluoron concentration and the visibility weight, respectively. When β is 0, the ants make their choice based solely on the distance to the city. When beta is equal to 0, the ants make their decision based solely on the concentration of the Flomont.

A recent application of ant colony optimization (ACO) algorithms is in the field of unmanned aerial vehicle (UAV) path planning. ACO algorithms simulate the behavior of real ant colonies to find the most optimal path between multiple points. these algorithms have been used in UAVs for tasks such as search and rescue, surveillance, and monitoring.

A study by Zhang et al. (2019) proposed an improved ACO algorithm for obstacle avoidance in UAV path planning, which uses dynamic weight coefficients to balance exploration and exploitation during the optimization process. The results showed that the algorithm was able to efficiently plan paths for UAVs while avoiding obstacles [12].

Similarly, Thakur et al. (2018) used a modified ACO algorithm for path planning of a swarm of UAVs. the algorithm was able to simultaneously optimize the paths of multiple UAVs to improve overall efficiency and reduce collision risks. the authors concluded that the algorithm could potentially be useful for applications such as border surveillance and disaster management [13].

In conclusion, ACO algorithms have shown great potential for UAV path planning and have been applied in various fields. these algorithms are capable of efficiently finding the most optimal paths for uavs while taking into account obstacles and other factors.

2.3 Particle Swarm Optimization (PSO)

Algorithm Background. First introduced in 1995 by Kennedy and Eberhart, Particle Swarm Optimization (PSO) is a well-known population-based stochastic optimization algorithm. In the process, each individual particle moves in the direction of the nearest neighbor, and each individual particle moves in the direction of the nearest neighbor.

Due to its simplicity and effectiveness, PSO is widely used in various fields such as engineering design, robotics, data mining, and finance.With the rapid development of big data and artificial intelligence technologies, PSO has become an increasingly powerful tool for solving complex optimization problems.

Recent studies have focused on improving the performance and efficiency of PSOs algorithms. One approach is to increase the diversity of particles by introducing mutation operators or combining PSOs with other optimization techniques.Another method is to modify the core mathematical operations in PSOs, such as the social learning mechanism, to improve its convergence speed and reduce the likelihood of premature convergence [15].

Algorithm Logic. Suppose there are m particles forming a population in a D-dimensional target search space, where the i-th particle is represented as a D-dimensional vector X1= (xi1, xi2, ----, axed) (i= (1, 2, m). The position of the i-th particle in the D-dimensional search space is xi. In other words, the position of each particle is a potential solution to the problem. Put another way, the position of each particle is a potential solution to solving the problem. Fitting xito an objective function calculates the goodness of fit, and the goodness of xican be measured by the magnitude of the goodness of fit. The "flying" velocity of the ith particle is also a D-dimensional vector, denoted as vi=(vi1, vi2...viD). The first PSO algorithm proposed by Kennedy and Eberhart operates on particles using the following formula:

$$v_{id} = v_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{gd}) \quad (5)$$

$$x_{id} = x_{id} + v_{id} \quad (6)$$

with i = 1, 2, ---, m, d = 1, 2, ---, D; the learning factors c1 and c2 are not negative; r1and r2are random numbers between [16]. Depending on a specific problem, this condition is either a maximum number of iterations or (and) a minimum adaptation threshold. The above PSO algorithm is also called the global version of PSO because pg is the optimal position of the entire particle population.The optimal position searched by the ith particle's neighbours can also be taken as pg, and the above method is also called the local version of PSO. The local version of PSO converges slower but is relatively less likely to fall into a local optimum.

By borrowing the idea of genetic algorithm, the concept of hybrid PSO algorithm was first proposed. It was further proposed as the HPSO algorithm with reproduction and subpopulation. Particles in the swarm are given a hybridization probability, which is user-defined and independent of the particles' fitness value. Based on the hybridization probability, a certain number of particles are selected and placed in a pool at each iteration. To keep the number of particles in the population constant, the particles in the pool randomly mate two by two to produce the same number of child particles and replace the parent particles with child particles.The positions of the child

particles are calculated by the arithmetic weighted sum of the positions of the parent particles [17].

$$child_1(x) = p * parent_1(x) + (1-p) * parent_2(x) \quad (7)$$

$$child_2(x) = p * parent_2(x) + (1-p) * parent_1(x) \quad (8)$$

$$child_1(v) = \frac{parent_1(v)+parent_2(v)}{|parent_1(v)+parent_2(v)|}|parent_1(v)| \quad (9)$$

$$child_2(v) = \frac{parent_1(v)+parent_2(v)}{|parent_1(v)+parent_2(v)|}|parent_2(v)| \quad (10)$$

For the local version of the PSO algorithm, it is equivalent to dividing several subpopulations within a population, so that hybridization operations can be performed either within the same subpopulation or optionally between different subpopulations.

3 Discussion

3.1 Genetic Algorithms

Advantages of genetic algorithms:

1) No concern with the problem domain fast random search capability.

2) The search starts from a population and is potentially parallel, allowing simultaneous comparisons of multiple individuals, ROBUST.

3) Search uses evaluation function inspired, simple process

4) Iterative using probabilistic mechanisms, stochastic in nature

5) Is scalable and can be easily combined with other algorithms.

3.2 Disadvantages of Genetic Algorithms

1) Since it is necessary to first encode the problem and then decode it after finding the optimal solution, the programming implementation of the genetic algorithm is complex.

2) In contrast, implementation of other three operators involve many parameters, e.g., cross-over and variance rates, and choice of these parameters can significantly affect solution quality.

3) The search algorithm is slow, and it takes more time to train the algorithm to have a more accurate solution.

4) The algorithm can be improved by a combination of several heuristic algorithms, but it has a certain dependence on the choice of the initial population.

5) The potential of the parallel mechanism of the algorithm is not used to the full extent. This is also a hot area of current research in genetics.

6) In this paper, Genetic Algorithm (proposed in 1972) is no longer a good solution for large computation problem, and it easily becomes "precocious". Instead, Hybrid Genetic Algorithms and Cooperative Evolutionary Algorithms are often used. They are all derived from GA.

3.3 Ant Colony Optimization

(1) The solution speed will be very slow, and the solution result will be very unreasonable if the values of α and β are set incorrectly.

(2) The ant colony algorithm is particularly computationally intensive and takes a long time to solve.

(3) In reality, it is difficult to achieve a limited amount of computation, because the ant colony algorithm ultimately requires all the ants to take the best path as determined by the algorithm.

(4) In practice, such as in image-based road planning, only one ant needs to find the best path, not all ants need to find the best path, it should be because if all ants are involved in the computation, it will greatly affect the efficiency of the computation and consume more time.

(5) The convergence speed of the ant colony algorithm is very slow, and it is easy to fall into the local optimum. In addition, it is difficult to compute the optimal path in experiments because the ant colony algorithm has no initial pheromone.

(6) The Ant Colony Algorithm is very complex, which can be reflected in the time needed to search.

(7) The ant colony algorithm has a tendency to become stagnant. In the algorithm, the end result is that all the ants follow the same path, which is not conducive to the search for a better path.

3.4 Genetic Algorithms

(1) The advantage of PSO is the simplicity of the algorithm and its ease of implementation.

(2) There are not many parameters to adjust, and no gradient information is required.

(3) It is an effective tool for solving nonlinear continuous, combinatorial, and mixed-integer nonlinear optimization problems.

Disadvantages

(4) The encoding of network weights and the selection of genetic operators are sometimes troublesome.

(5) It is usually necessary to use period improvement methods to avoid falling into local optima because discrete optimization problems are not well handled and easily fall into local optima. Table 1 showed a comparison of the advantages and disadvantages of the three algorithms.

Table 1. A comparison of the advantages and disadvantages of the three algorithms

Algorithm	Characteristic	Advantage	Disadvantage
GA	Simulate the evolution process, global optimization and support parallelism through methods	Can handle multiple problems and find the global optimal solution	May fall into local optima with a long search time
ACO	Simulate ant behavior, global optimization, and deal with constraint problems	Find the global optimal solution and handle conditional constraint problems	Impressions about changes in the environment and long search times
PSO	Simulate bird swarm behavior, global optimization, deal with continuous problems,	Find global optimal solutions, and deal with continuity problems	May fall into local optima and be sensitive to parameters

3.5 Improvement Methods

GA. 1) Adjusting the probability of crossover and mutation. Given that crossover and mutation are the core operations of the GA algorithm, we can adjust the probability of crossover and mutation based on the characteristics of the problem to improve algorithm performance. A higher crossover probability will lead to faster population convergence, but it is also prone to premature phenomena, while a lower crossover probability will increase the algorithm's global search ability; Similarly, a higher mutation probability will increase the diversity of the algorithm but may also have a negative impact on accuracy. 2) Improve selection operations. Selection operation means that individuals with higher fitness are selected from the population for crossover or mutation, so selection operation is very important for the evolution of the population. The commonly used selection methods currently include tournaments, and

elite retention, among others. Among them, selecting the elite retention strategy can keep the best individuals while continuously compressing the population size, thereby avoiding the risk of local convergence. 3) Introducing new evolutionary mechanisms. In the existing ga algorithm, we need to score each individual according to the fitness function, and then carry out selection, crossover and mutation operations. If we could add some new evolutionary mechanisms, such as exploring evolution.

ACO. 1) Introducing local search: It is not enough for ants to only consider the best choices around them when moving. Some local search strategies can be used to broaden the search range of ants, such as introducing 2-opt or 3-opt methods. 2) Properly increase the volatilization speed of pheromone: pheromone refers to a kind of information released by ants on the path, which is used to attract other ants to choose the same path. Properly increasing the speed of pheromone volatilization can promote the fast convergence of the algorithm and prevent it from falling into the local optimal solution prematurely. 3) Introducing dynamic weight: the commonly used heuristic functions in aco algorithm are distance and pheromone concentration. However, when selecting these heuristic functions, it is necessary to manually determine the weights between them. To better balance the impact of these two heuristic functions, dynamic weights can be used to adjust the proportion of the two functions. 4) Improved pheromone update strategy: pheromone update strategy directly affects the convergence and stability of the algorithm. The pheromone update strategy can be optimized by introducing adaptive parameters and other means to make it more consistent with the actual application scenarios. 5) Polynomial and logarithmic distributions are introduced: the traditional aco algorithm uses continuous smooth distribution functions (such as normal distribution), and these continuous function are not easy to achieve discretization. Some logarithmic or polynomial based discretization methods can be introduced to improve the aco algorithm, and improve the efficiency and accuracy of the algorithm.

PSO. 1) Change parameter settings: When performing the PSO algorithm, it is necessary to set appropriate parameters, such as speed factor and learning factor. If reasonable parameter selection is made, the convergence speed and result accuracy of the PSO algorithm can be improved. 2) Using more complex dynamic models: Compared to standard PSOs, using more complex dynamic models can better utilize

historical information for learning and obtain optimal solutions faster. For example, adaptive pso, chaotic pso, etc. 3) Introducing multi-objective functions: Introducing multi-objective functions into the pso algorithm can help search for more comprehensive and reliable solutions. For example, multi-objective particle swarm optimization (mopso) algorithm. 4) Combining with other algorithms: Combining the PSO algorithm with other optimization algorithms, such as GA or DE, can also improve algorithm performance and achieve better results. 5) Non global optimal avoidance method: It is common to encounter situations where particles deviate or have local optima that cannot reach global optima. Some techniques to avoid deviation and local optima may be helpful, such as decreasing inertia weight. 6) Implement different search strategies: Support mutation strategies for different search behaviors to increase the overall search ability of the PSO. This method has resulted in a series of algorithms that improve PSO performance by continuously mutating and using multiple populations.

4 Conclusion

After analyzing and comparing the performance of three popular drone path planning algorithms in recent years: ga, aco, and pso, we can draw some useful conclusions. each algorithm has its unique features and drawbacks that can be utilized depending on the specific application scenario.

Firstly, the GA algorithm has a strong ability to search for optimal solutions, mainly based on genetic selection and crossover. This approach performs well in complex and dynamic environments where obstacle avoidance is required. Second, the ACO algorithm relies on ant colony optimization to find the shortest path towards the goal. It is suitable for solving problems with multiple start and endpoint nodes, and in cases where extensive exploration is needed. Finally, the PSO algorithm utilizes swarm intelligence to identify high-quality solutions while enabling rapid convergence. It is advantageous in dealing with tasks that require real-time responsiveness.

Moreover, we found that these algorithms can be further improved through the integration of advanced methods such as deep learning and reinforcement learning. although there are still limitations and challenges with each algorithm, such as convergence speed or robustness, they have contributed significantly to the development of drone path planning techniques.

In conclusion, selecting the most appropriate algorithm depends on the specific requirements of the task, and a thorough comparison and analysis can guide decision-makers in this field. we expect that future studies will continue to explore more efficient and versatile drone path planning algorithms that could lead to breakthroughs in unmanned aerial vehicle applications.

Acknowledgment (Authors Contribution)

All authors are listed in alphabetical order, with equal contributions from all authors.

References

1. Wan, Z., & Jin, Y. Exploring algorithm portfolios for combinatorial optimization: a case study on travelling salesman problem. IEEE Transactions on Emerging Topics in Computational Intelligence, 4(1), 82-94 (2019).
2. Zhao, S., Liang, J., Zhang, Q. An adaptive differential evolution algorithm for the travelling salesman problem. Applied Soft Computing, 72, 527-544 (2018).
3. Song, Z., Chen, I., Hou, S., Shen, S., Yau, S., Qiu, S. Identification of genes associated with systemic lupus erythematosus using a genetic algorithm based on a support vector machine. Experimental and Therapeutic Medicine, 18(3), 2171-2178 (2019).
4. Xia, Y., Chen, G., Zhang, J., Shao, G., Yao, X., Liu, H. A genetic algorithm-based prediction model for prostate cancer. J. Healthcare Engineering, 2020, 1-13 (2020).
5. Chen, Y., Gao, H., Wang, J., Yu, Q. A genetic algorithm-based optimization method for uav wind energy conversion system. IEEE Access, 7, 160817-160826 (2019).
6. Yu, Yingying. Chen Y., Li T. Y.. An improved genetic algorithm for solving the travel quotient problem. Control and Decision Making, 29(8):1483-1488 (2014).
7. Li PY. UAV multi-objective path planning based on genetic algorithm. Agricultural Equipment and Vehicle Engineering, 57(1):4 (2019).
8. Li, X., Zhu, Y., Wu, J., Shao, S., Han, B. Adaptive genetic algorithm based on chaos search and obstacle suitability degree for UAV search and rescue path planning. IEEE Transactions on Systems, Man, and Cybernetics: systems, 50(1), 149-161 (2020).
9. Guo, Y., Shi, X., Wu, Z., Chen, W. A hybrid ant colony optimization algorithm for solving the traveling salesman problem. Mathematics, 8(4), 488 (2020).
10. Mohammadzadeh, E., Hedayatzadeh, R., Hosseini Ahmadi, J. A novel multi-objective energy management system for smart homes based on the improved ant colony. (2019).

11. Zhang, Y., Wang, X., Li, S., Zhang, I. Obstacle avoidance path planning for unmanned aerial vehicle based on improved ant colony algorithm. Chinese J. Aeronautics, 32(10), 2230-2242 (2019).
12. Thakur, J. K., Kumar, K., Singh, V.P., Srivastava, N. Exploring modified ant colony optimization for path planning of swarm of UAVs. Int. J. Aerospace Engineering, (2018).
13. Li, T. Liu, Y. Wu, Z. Efficient particle swarm optimization with adaptive mutation. IEEE Transactions on Cybernetics, 50(7), 3084-3097 (2020).
14. Cheng, J. Fu, X., Wang, I. A novel particle swarm optimization algorithm with adaptively adjusted inertia weight. Mathematical Problems in Engineering, 2019: 7690635 (2019).
15. Li B.Y., Xiao Y. A new algorithm for evolutionary computation--Particle swarm optimization algorithm. Computer Science, 2003(06):19-22.
16. Li A., Qin Z., Bao F. Particle swarm optimization algorithm. Computer Engineering and Applications, 2002(21):1-20.
17. Yang, W., Li, Z. A survey of particle swarm optimization algorithm. Chinese J. Engineering Science, 05, 87-94 (2004).