APPLICATION OF THE DIFFERENTIAL EVOLUTION ALGORITHM IN UNMANNED AERIAL VEHICLE (UAV) PATHFINDING

Lappeenranta–Lahti University of Technology LUT

Bachelor’s Programme in Science (Technology), Bachelor's thesis

2024

Hanze Yu

Examiner(s): Yunhe Wang, D.Sc. (Tech.)

Associate Professor Damian Kedziora

ABSTRACT

Lappeenranta–Lahti University of Technology LUT

LUT School of Engineering Science

Software Engineering

Hanze Yu

Application of the Differential Evolution Algorithm in Unmanned Aerial Vehicle (UAV) Pathfinding

Bachelor's thesis

2024

35 pages, 11 figures, 6 tables and 0 appendices

Examiner(s): Yunhe Wang, D.Sc. (Tech.) and Associate Professor Damian Kedziora

Keywords: Genetic algorithm; UAV; Pathfinding algorithm; Differential Evolution Algorithm

This paper studies the differential evolution (DE) algorithm for UAV path planning in a three-dimensional complex environment. By simulating the mutation, crossover and selection operations in the natural evolution process, the flight path of the UAV is gradually optimized. The experiment compares the performance of the DE algorithm with other path planning algorithms under the same starting point, end point and obstacle settings, focusing on the path length and calculation time. The experiment shows that the path optimization quality of the DE algorithm is significantly improved, and it can effectively avoid obstacles, which is suitable for application scenarios that require high-precision path planning. The SHADE algorithm is also used to experiment with the multi-path patrol conflict problem.

Table of contents

1. Introduction

Unmanned aerial vehicles (UAVs) are extensively utilized in both military and civilian domains because of their compact size, ease of use, and robust survival, particularly in hazardous conditions and at low altitudes. In this case, planning a collision-free and shortest path is the key to ensure that the UAV can successfully complete its mission. Due to the uncertainty and dynamics in the real environment, UAVs face many uncontrollable factors when performing tasks, so the study of UAV path planning algorithms is particularly important (Nikolos et al., 2003, p. 899).

This paper focuses on the application of differential evolution (DE) algorithm in UAV path planning to solve the complex challenges faced by UAVs when operating in dynamic and unpredictable environments. With the continuous expansion of the application scope of UAVs, a powerful and adaptive navigation strategy is urgently needed. This research not only has direct practical significance, but also helps to promote the development of autonomous navigation technology (Floreano and Wood, 2015, p. 121).

Traditional UAV path planning algorithms perform well in structured environments, but often perform poorly under complex or changing conditions. Although earlier studies, such as those by Beard et al. and Mettler et al., have demonstrated the effectiveness of these methods, they have generally ignored highly dynamic scenarios (Beard and McLain, 2002, p. 762; Mettler et al., 2003, p. 104). In contrast, the differential evolution algorithm proposed by Storn and Price is more conducive to the implementation of UAV pathfinding algorithms due to its robustness and flexibility, and can better cope with the complex situations faced by UAVs (Storn and Price, 1996, p. 346).

Although there is evident potential for enhancement in differential evolution algorithms, these algorithms are now insufficiently resilient to thoroughly assess the numerous unforeseen situations encountered in UAV operations. This research and experimental investigation seeks to examine and evaluate the effectiveness of differential evolution algorithms in comparison to classic path planning algorithms (Chakraborty, 2008, p. 56).

The study is being conducted to address the lack of prior research on optimizing the computational efficiency and path length of UAVs in complicated situations. This is an important aspect that needs continual improvement. Despite the limited data available due to constraints in test scenarios and UAV models, this experiment offers valuable data and information for the advancement of differential evolution algorithms in UAV path planning (Qin and Suganthan, 2005, p. 125).

The work will adhere to a systematic structure, commencing with a comprehensive assessment of existing literature. Following this, there will be an extensive overview of the DE algorithm and three experiments. These experiments will include thorough explanations of the methodologies employed and the outcomes achieved. Subsequently, the outcomes of the three studies will be examined. Lastly, the report will analyze the consequences of the research and provide potential avenues for further advancement.

2. literature review

Unmanned aerial vehicles (UAVs) have garnered significant interest in path planning, particularly in military and civilian applications. The current path planning algorithms consist mostly of classical methods and evolutionary algorithms. The A* method and the Dijkstra algorithm are effective in structured settings but are less efficient in high-dimensional and dynamic environments (Nikolos et al., 2003, p. 899). Typically, these algorithms are incapable of efficiently managing intricate three-dimensional settings and moving impediments in real-time (Mettler et al. 2003, p. 104).

Evolutionary algorithms offer a very adaptable and effective approach to path design. The Differential Evolution (DE) method, introduced by Storn and Price in 1996, is widely recognized as a significant breakthrough in the domain of global optimization (Storn and Price 1997, p. 346). The Differential Evolution (DE) method improves the individuals in the population by performing mutation, crossover, and selection processes, progressively converging towards the ideal solution. The DE method is highly effective in handling intricate path planning issues due to its strong resilience and straightforward implementation (Storn and Price 1997, p. 346).

The adaptive differential evolution (ADE) method enhances the search capability and convergence speed of the classic DE algorithm by dynamically modifying the mutation and crossover probabilities (Storn and Price 1997, p. 350). The ADE algorithm has the ability to adaptively modify the mutation factor and crossover probability, which enables it to display various search methods at different search stages. This improves both the route search capability in the global space and the route search efficiency in the local space (Brintaki and Nikolos 2005, p. 1012).

Although the differential evolution algorithm performs well in path planning, it encounters some obstacles in real-world scenarios. The classic DE algorithm may converge prematurely in complex spaces and converge to a local optimal solution before finding the global optimal solution (Ghambari et al., 2020, p. 789). In addition, in a dynamic environment, the DE algorithm must constantly modify parameters to adapt to changes in the environment. However, the current adaptive strategy is not good enough, resulting in insufficient flexibility of the algorithm in practical applications (Nikolos et al., 2003, p. 900).

In order to enhance the effectiveness and resilience of path planning, scholars have suggested hybrid algorithms that integrate the DE algorithm with other conventional algorithms. For instance, the DE algorithm is integrated with the A* algorithm, with the A* method being employed for overall path planning, and subsequently, the DE algorithm is utilized for local optimization to handle the presence of dynamic barriers (Ghambari et al., 2020, p. 793). This hybrid approach demonstrates excellent performance in terms of both computational efficiency and the quality of the generated paths.

Collaborative route planning remains an unsolved issue. In multi-UAV missions, the coordination of numerous UAVs is necessary to ensure simultaneous arrival at the target site and prevent collisions. This places more demands on the path planning algorithm. The current study mostly concentrates on the route planning of a solitary UAV and lacks comprehensive investigation into the collaborative path planning of numerous UAVs (Beard et al. 2002, p. 762).

Storn and Price's work on UAV route planning is largely considered the benchmark in the industry. The differential evolution method they presented has gained significance as a solution to the global optimization issue due to its straightforward and efficient optimization mechanism (Storn and Price 1997, p. 346). Their study established the groundwork for the later advancement of adaptive differential evolution algorithms and facilitated the utilization of evolutionary algorithms in the domain of path planning.

The research conducted by Beard et al. (2002) on the collaborative path planning of numerous UAVs is widely regarded as a significant achievement in this particular area of study. The authors suggested a system for allocating and intercepting targets in a coordinated manner. This strategy enhanced the efficiency and safety of completing missions by optimizing the routes of several UAVs (Beard et al. 2002, p. 762). This study serves as a crucial point of reference for the path planning of multi-UAV systems.

Machine learning approaches have been used to make more progress in UAV route planning. Reinforcement learning has been utilized in route planning issues, specifically in the context of Unmanned Aerial Vehicles (UAVs). In this approach, UAVs learn the most efficient courses by iteratively interacting with the environment and adjusting their behaviors based on trial and error. (Sutton and Barto 2018, p. 198). This technique enables Unmanned Aerial Vehicles (UAVs) to more efficiently adjust to changing situations in comparison to conventional algorithms.

Furthermore, the incorporation of sensor data and the capacity to process it in real-time has improved the flexibility of UAVs in dynamic contexts. Scientists have created algorithms that use sensor fusion techniques to merge information from several sensors. This allows for a thorough comprehension of the surroundings and improves the accuracy of path planning (Thrun et al. 2005, p. 236).

Recent research have investigated the utilization of bio-inspired algorithms for UAV route planning, alongside hybrid and machine learning methods. Algorithms that draw inspiration from natural processes, such as ant colony optimization (ACO) and particle swarm optimization (PSO), have shown encouraging outcomes in navigating intricate settings and evading obstacles (Dorigo et al. 1996, p. 210; Kennedy and Eberhart 1995, p. 1942). These algorithms, inspired by biological systems, utilize the combined actions of basic agents to accomplish reliable and effective path planning.

Moreover, progress in hardware and processing capacity has facilitated the integration of increasingly intricate algorithms into real-time systems. Advancements in onboard processors and communication networks enable UAVs to do intricate computations and synchronize with one another during flights (Ghamry et al. 2016, p. 75).

3. Algorithm and Experiment Introduction

3.1 Algorithm Introduction

The Differential Evolution (DE) method was introduced by Storn and Price in 1997. The technique described is a global optimization method that is well-suited for tackling optimization problems in continuous spaces (Storn and Price, 1997, p. 346). The Differential Evolution (DE) method iteratively improves individuals in the population by applying mutation, crossover, and selection processes, progressively bringing them closer to the ideal solution. The method possesses the benefits of high resilience, straightforward implementation, and compatibility with parallel computing (Storn and Price, 1997, p. 346).

The core steps of the DE algorithm include:

Initialize the population: Randomly generate an initial population containing *N* candidate solutions, each of which is a *D* dimensional vector representing a point in the search space.

Mutation operation: For each candidate solution *x* (called the target vector), select three different individuals in the population except *x* and generate a mutation vector:

$$\mathbf{v}_i = \mathbf{x}_r 1 + F \cdot (\mathbf{x}_r 2 - \mathbf{x}_r 3) \tag{1}$$

Where F is the factor of variation, usually 0.5.

Crossover operation: The mutation vector is crossed with the target vector to generate the test vector *u*. The crossover operation uses a uniform crossover method and is controlled by the crossover probability *CR*:

$$u_{i,j} = \begin{cases} v_{i,j} & \text{if } rand_j(0,1) \leq CR \text{ or } j = j_{rand} \\ x_{i,j} & \text{otherwise} \end{cases} \quad (2)$$

Selection: By comparing the fitness values of the test vector and the target vector, the vector with better fitness is selected to enter the next generation population:

$$x_i^{(t+1)} = \begin{cases} u_i & \text{if } f(u_i) \leq f(x_i) \\ x_i & \text{otherwise} \end{cases} \quad (3)$$

Based on the original DE algorithm, many derivative algorithms have been generated to optimize and solve other problems. Other DE derivative algorithms are introduced below:

Adaptive Differential Evolution (ADE) enhances the search capability and speed of convergence of the algorithm by dynamically modifying the mutation and crossover probabilities, using the classic DE as a basis (Storn and Price, 1997, p. 350). The ADE algorithm has the capability to adaptively modify the mutation factor and crossover probability, enabling it to exhibit various search strategies throughout different search phases. This leads to an enhancement in both the global search capability and the local search efficiency of the algorithm (Brintaki and Nikolos, 2005, p. 1012).

The JADE algorithm enhances the ADE method by incorporating an external archive and an adaptive adjustment mechanism. The external archive is a repository that preserves high-quality solutions that have been deleted in order to improve the variety of the population and prevent premature convergence. The JADE algorithm incorporates successful historical knowledge to dynamically alter and enhance the mutation factor (F) and crossover probability (CR), hence increasing the method's adaptability and robustness (Zhang and Sanderson, 2009, p. 1023).

The SHADE method, which stands for Success-History based Adaptive Differential Evolution, is a variation of the Differential Evolution (DE) algorithm. It incorporates a success history mechanism to dynamically modify the mutation factor (F) and crossover probability (CR). The SHADE algorithm utilizes the success history data from each generation to adjust the parameters F and CR. This process enhances the algorithm's capability for global exploration and local exploitation. The SHADE algorithm utilizes external archives to preserve discarded solutions, hence enhancing the variety of the population (Tanabe and Fukunaga, 2013, p. 76).

3.2 Experimental introduction and analysis

All experimental codes can be viewed through the Github link:

https://github.com/Littlefish0925/DE-algorithm-in-UAV-pathfinding-experimental-code

3.2.1 Comparison of DE algorithm pathfinding algorithms in three-dimensional environments

This experiment aims to compare the performance of multiple differential evolution (DE) algorithms in pathfinding in a three-dimensional environment, including the basic differential evolution (DE) algorithm, the adaptive differential evolution (ADE) algorithm, the JADE algorithm, and the SHADE algorithm. And the number of iterations of each algorithm is the same, all 500 times. The experimental environment is designed as follows:

Starting point and end point: set a fixed starting point and end point.

Starting point: [0, 0, 0]

End point: [10, 10, 10]

Obstacles: set multiple obstacles of different shapes and positions, including cuboids and cylinders. The obstacles are randomly distributed between the paths, which increases the complexity of path planning.

Algorithm parameters:

DE algorithm: mutation factor F = 0.5, crossover probability CR = 0.9.

ADE algorithm: adaptively adjust the mutation factor and crossover probability.

JADE algorithm: parameters $\mu F = 0.5$, $\mu CR = 0.5$, adapt to the adaptive learning rate.

SHADE algorithm: historical memory parameter H = 10, combining the archive mechanism and adaptive adjustment strategy.

Evaluation indicators: path length, calculation time

Experimental results and analysis

The average path length and average calculation time of each algorithm under different operating conditions were recorded in the experiment. The following are detailed results and analysis. Please refer to Table 1, Table 2 and Figure 1, Figure 2 for data

Table 1: Data comparison of DE algorithm's pathfinding function in a 3D environment (Scenario 1)

Algorithm	Average distance (meters)	Average running time (seconds)
DE F = 0.5, CR = 0.9	22.679	17.524
ADE	21.889	17.153
JADE μCR = μF = 0.5	33.725	17.651
SHADE	19.986	17.105

Table 2: Data comparison of DE algorithm's pathfinding function in a 3D environment (Scenario 2)

Algorithm	Average distance (meters)	Average running time (seconds)
DE F = 0.5, CR = 0.9	20.847	17.065
ADE	21.036	16.453
JADE μCR = μF = 0.5	31.232	16.470
SHADE	20.539	16.746

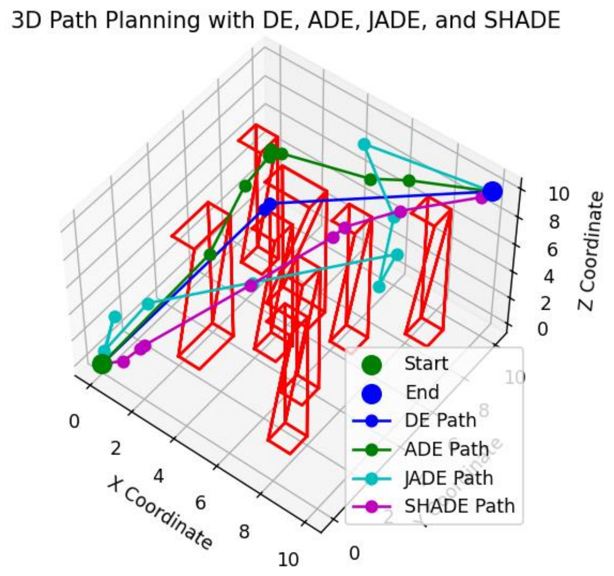

Figure 1: Route comparison of DE algorithm' s pathfinding function in a 3D environment (Scene 1)

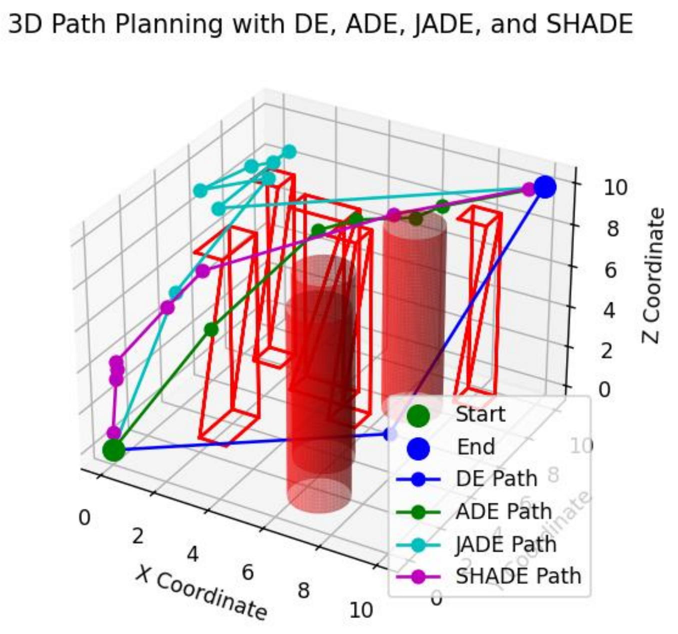

Figure 2: Route comparison of DE algorithm' s pathfinding function in a 3D environment (Scene 2)

Based on the facts provided above, we may derive the experimental findings:

TThe differential evolution (DE) method has very consistent performance, but its trajectory length is quite long. The path length of the ADE method is slightly shorter than that of the DE algorithm, indicating that adaptive adjustment parameters can improve the path to some extent. The trajectory of the JADE algorithm is much longer than that of the other algorithms, indicating that the current experimental parameters may not be able to discover the most effective path. The SHADE algorithm has the shortest path length, indicating its superior ability in path optimization.

The differential evolution (DE) algorithm has a higher convergence rate. The computation time of the ADE algorithm is slightly shorter than that of the DE method, indicating the effectiveness of the adaptive technique. Although the path of the JADE method is longer, the computation time required is comparable to that of the other algorithms. The computation time of the SHADE algorithm is the shortest, which further confirms its efficiency.

The trajectories of the SHADE and ADE algorithms show a high degree of smoothness, indicating their strong ability in optimizing paths. The trajectory of the DE algorithm is generally smooth, but occasionally there are large deviations. The trajectory of the JADE algorithm is not smooth enough, and there are often sudden turns, resulting in longer distances and increased time consumption.

In the algorithm comparison, the SHADE method significantly outperformed the other methods in terms of path length and computation time. This shows that the SHADE algorithm is very efficient and reliable in finding paths in 3D environments. The DE and ADE algorithms performed well, especially in terms of obstacle avoidance and computational efficiency. The JADE method showed suboptimal performance in terms of path length, indicating that more parameter tuning may be needed to enhance its effectiveness.

The SHADE method performed well in this trial, and its adaptive parameter tuning and archiving mechanism performed well in both path optimization and processing efficiency. Future research will further investigate the SHADE method and compare it with other path-finding algorithms in various situations. In addition, experiments will be conducted to resolve multi-path conflicts.

3.2.2 Comparison of DE algorithm with other algorithms

The objective of this experiment is to evaluate and contrast the efficacy of four distinct path planning algorithms in a three-dimensional setting: the A* algorithm, the SHADE method, the genetic algorithm (GA), and the particle swarm optimization (PSO) algorithm. In order to maintain fairness, all algorithms were subjected to identical beginning point, end point, and obstacle configurations. Additionally, each algorithm underwent the same amount of iterations, precisely 200 times. The experiment assessed the path length, computation time, and collision count of the final path for each method, using a set number of iterations.

The following presents a comprehensive overview of all experimental algorithms:

The A* algorithm is a heuristic search technique that is commonly employed in path planning and graph search. The technique combines the benefits of the Dijkstra algorithm and directs the search using a heuristic function, often the estimated distance between the current node and the target node. This approach reduces the search area and enhances the efficiency of the search process (Hart et al., 1968, p. 262). The A* algorithm excels in grid-like environments and is capable of identifying the best route from the initial point to the final location. Nevertheless, in settings that are both high-dimensional and complex, the computing effort and time required by the system may see a substantial rise (Mettler et al., 2003, p. 104).

A Genetic Algorithm (GA) is an evolutionary algorithm that operates on the principles of natural selection and genetic mechanics. They systematically improve the solution by emulating the selection, crossover, and mutation procedures observed in biological evolution. The technique is commonly employed for solving global optimization problems and is known for its high level of resilience and flexibility (Holland, 1975, p. 187). Genetic algorithms (GA) are capable of successfully managing intricate environments in path planning. However, they may exhibit delayed convergence in high-dimensional spaces and are prone to quickly getting trapped in local optimum solutions (Goldberg, 1989, p. 123).

Particle Swarm Optimization (PSO) is an optimization system that utilizes swarm intelligence to imitate the collective behavior observed in bird flocks or fish schools. The ideal solution is achieved by particles progressively converging through the exchange of information and joint efforts. According to Kennedy and Eberhart (1995, p. 195), every particle in the system represents a possible solution and updates its location by considering its own experience as well as the experiences of its neighboring particles. The utilization of Particle Swarm Optimization (PSO) in the domain of path planning demonstrates its commendable capacity for global exploration. However, it is important to note that when confronted with dynamic situations, supplementary modifications and refinements may be necessary (Eberhart and Shi, 2001, p. 114).

Experimental scenario setting

Two experimental scenarios were set for testing

Starting point: [0, 0, 0]

End point: [10, 10, 10]

Obstacles: Multiple obstacles are set on the path, and obstacles exist in the straight path to increase the complexity of path planning.

Experimental results

The experimental results are shown in Table 3, Table 4 and Figures 3-8, which record the performance of the four algorithms in path length, calculation time and number of collisions. The data in the table are the average values obtained after 20 experiments.

Table 3: Performance comparison of different path planning algorithms (Scenario 1)

Algorithm	Average Path Length (meter)	Average Time (seconds)	Average Collisions
A*	30.0	11.183	0
SHADE	19.986	17.505	0
GA	67.171	17.073	2.0
PSO	44.342	17.232	0.6

Table 4: Performance comparison of different path planning algorithms (Scenario 2)

Algorithm	Average Path Length (meter)	Average Time (seconds)	Average Collisions
A*	30.0	10.136	0
SHADE	21.149	16.508	0
GA	61.798	16.292	3.6
PSO	61.693	16.496	0

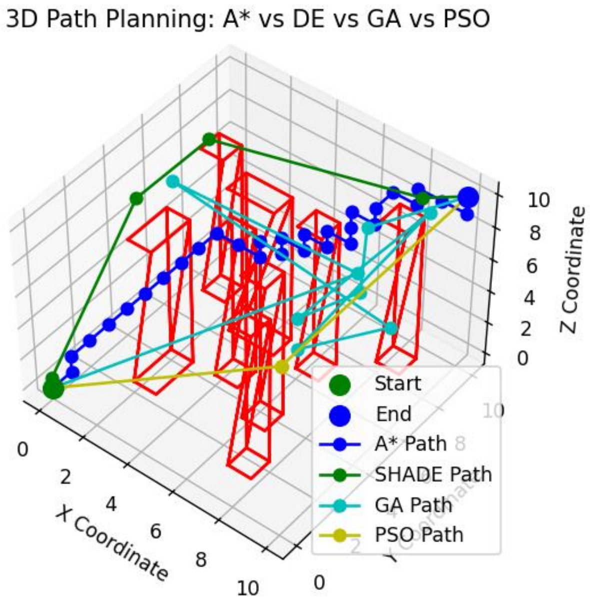

Figure 3: Route comparison of different path planning algorithms (Scenario 1)

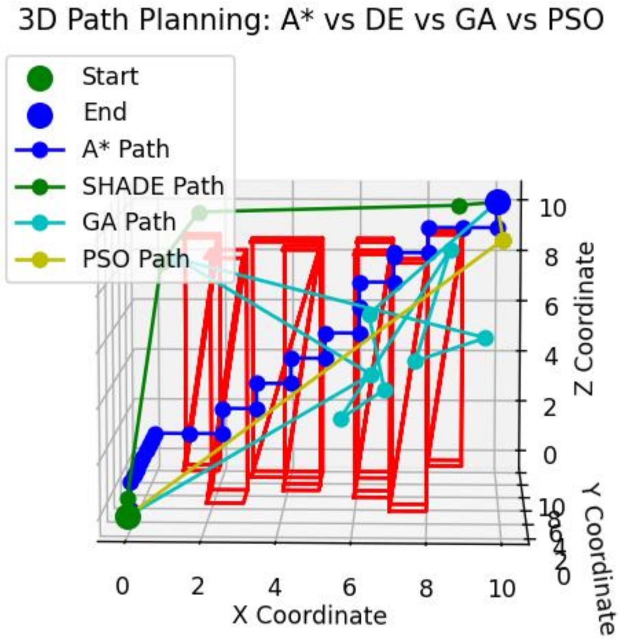

Figure 4: Comparison of routes of different path planning algorithms (front view of scene 1)

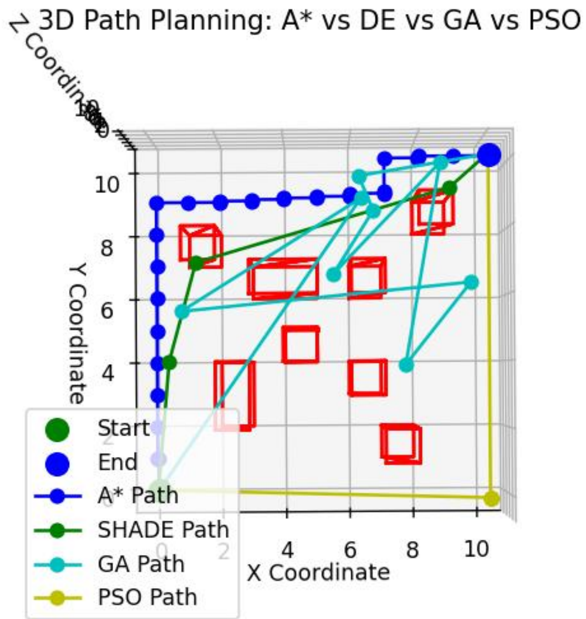

Figure 5: Comparison of routes of different path planning algorithms (top view of scenario 1)

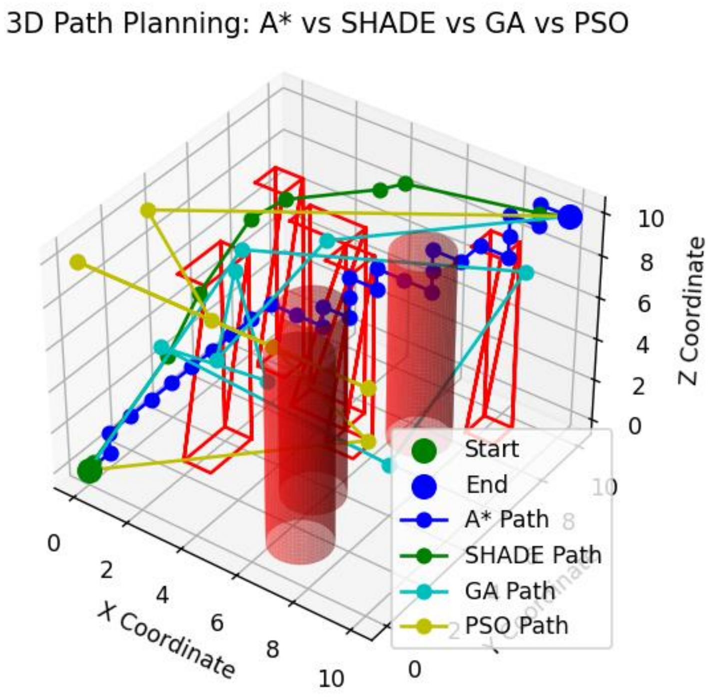

Figure 6: Route comparison of different path planning algorithms (Scenario 2)

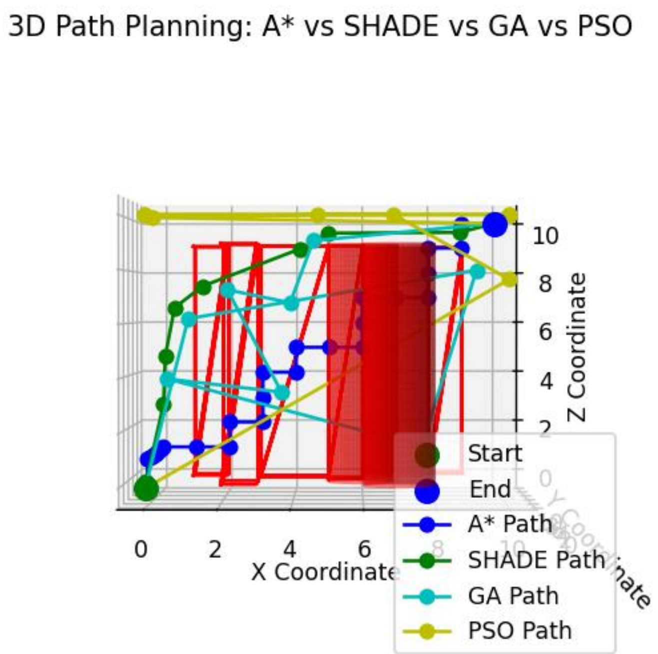

Figure 7: Comparison of routes of different path planning algorithms (front view of scene 2)

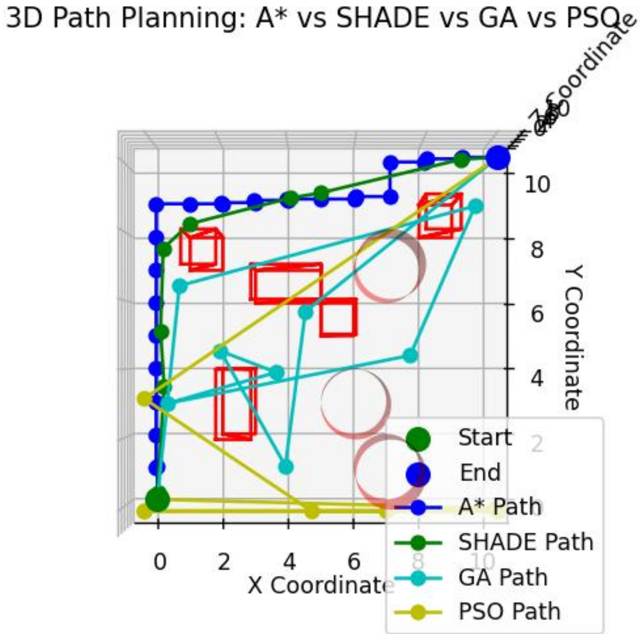

Figure 8: Comparison of routes using different path planning algorithms (top view of scenario 2)

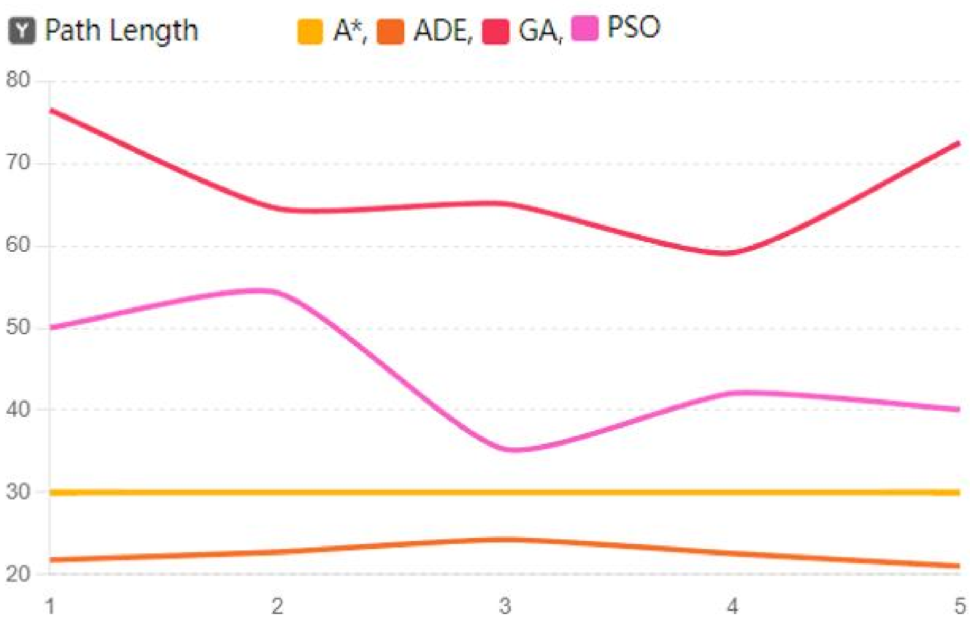

Figure 9: Comparison of experimental data of different path planning algorithms

According to the experimental results, the SHADE algorithm shows obvious advantages in path length and calculation time. The analysis is as follows:

The SHADE algorithm has the shortest path length among the four algorithms, which shows that it is very efficient in path optimization. Compared with the SHADE algorithm, the path length of the A* method is shorter, but not optimal. The path lengths of the PSO method and the GA algorithm are significantly longer, indicating that path and algorithm optimization are still needed.

When the search space is small, the A* method shows excellent efficiency. The calculation time of the SHADE method is relatively the longest, but it can produce the highest quality optimized route. The calculation time of the particle swarm optimization (PSO) and genetic algorithm (GA) techniques is moderate.

The SHADE algorithm and the A* algorithm show excellent obstacle avoidance capabilities because they have no collisions with obstacles. However, the PSO algorithm and the GA algorithm have collisions, so the ability of these two algorithms to cross obstacles in complex situations is limited.

Analysis of images

A* path (blue): The path has a somewhat linear characteristic, circumventing all barriers, albeit it has a greater length.

SHADE path (cyan): The path is relatively tortuous, but it avoids all obstacles, and the path length is the shortest, showing good optimization ability.

GA path (green): The path is obviously longer, and part of the path collides with obstacles, showing poor obstacle avoidance and path optimization capabilities.

PSO path (yellow): The path length and calculation time are reasonable, nevertheless, a portion of the path intersects with obstacles, indicating a need for enhancement in obstacle avoidance capability.

When comparing the length of the path and the time it takes to calculate, the SHADE algorithm demonstrates greater performance in path planning. Despite its relatively long computation time, the SHADE method is capable of identifying the shortest path, surpassing other algorithms by a wide margin. In practical scenarios, the degree of path optimization usually takes precedence over the time required to calculate the path. This is because a shorter path may reduce the energy consumption of the drone, thereby reducing the flight cost of the drone. In addition, the SHADE algorithm effectively eliminates collision events, thereby greatly improving the safety of the drone. Therefore, the SHADE algorithm has obvious advantages in navigating complex environments, and this article recommends it as a path planning method.

3.2.3 DE algorithm for multi-path problem

Experimental design

The objective of this experiment is to assess the effectiveness of the adaptive differential evolution (SHADE) algorithm in solving complex issues with several paths in a three-dimensional setting. During the experiment, several initial and final positions were established, along with varied barriers such as cuboids and cylinders. We assessed the efficacy and excellence of the SHADE algorithm by analyzing its performance on various pathways and measuring parameters such as path length and computation time.

The experimental scene is set as follows:

Starting point and end point:

Path 1: starting point [0,0,0], end point [10,10,10]

Path 2: starting point [0,10,0], end point [10,5,10]

Path 3: starting point [10,0,0], end point [0,10,10]

Set multiple cuboid or cylindrical obstacles to increase the complexity of path planning

Experimental results

The following Tables 5, 6 and Figures 9 and 10 record the average distance and running time of the SHADE algorithm on different paths, as well as the detailed paths.

Table 5: Performance description of the SHADE algorithm for multipath problems (Scenario 1)

Path	Average distance (meters)	Run time (seconds)
1	19.575	67.398
2	17.443	
3	20.179	

Table 6: Performance description of the SHADE algorithm for multipath problems (Scenario 2)

Path	Average distance (meters	Run time (seconds)
1	19.401	60.528
2	17.704	
3	21.864	

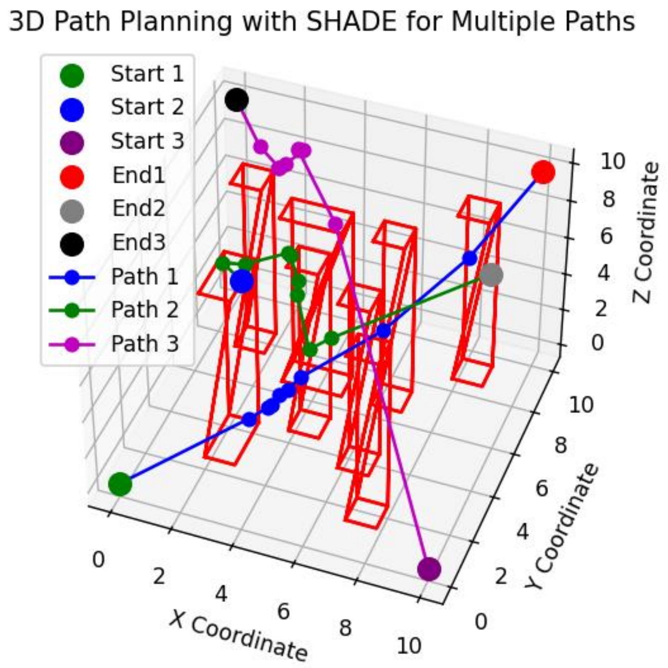

Figure 10: SHADE algorithm for multi-path route design (scenario 1)

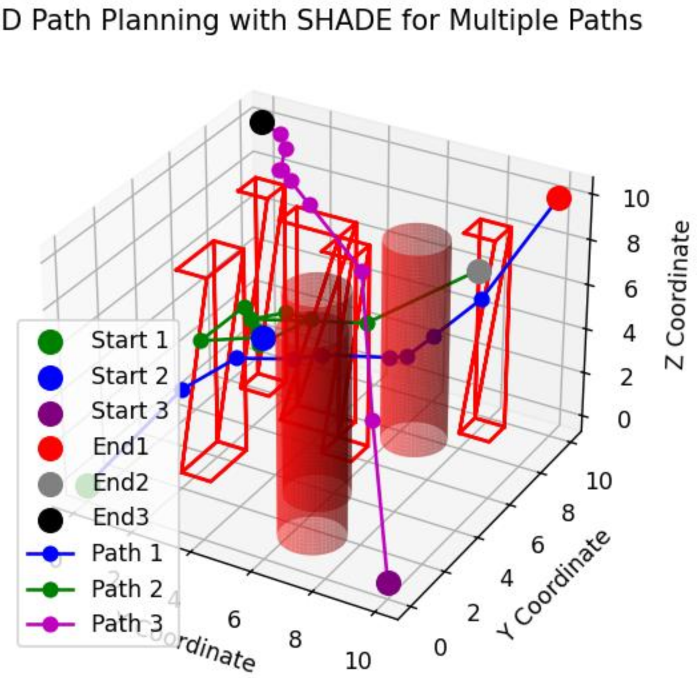

Figure 11: SHADE algorithm for multi-path route design (scenario 2)

Image analysis

Path 1 (blue): The path is brief and circumvents all barriers, showcasing the efficiency of the SHADE algorithm in optimizing a single path.
Path 2 (green): This way likewise circumvents any barriers and prior paths, and it is quite straightforward, providing more evidence of the effectiveness of the SHADE algorithm in intricate surroundings.
Path 3 (purple): Despite its highly convoluted nature, this road effectively circumvents all barriers and prior paths, showcasing the SHADE algorithm's global search and obstacle avoidance capabilities.

The SHADE method is very effective in identifying shorter paths. A total of 40 experiments were conducted in this study, with 20 experiments in each environment. The experiments demonstrated the excellent path optimization skills of the SHADE algorithm. It can effectively avoid all obstacles in path planning as well as existing paths, perfectly ensuring collision-free navigation. This shows that the SHADE algorithm has a highly robust obstacle avoidance capability in complex situations. The SHADE method can solve both single-path problems and multi-path problems. Empirical results show that the SHADE algorithm can generate valid paths from each initial position to the end point with high path quality. Although the SHADE method performs well in route optimization, its computational time in multi-path problems is still short, showing excellent computational efficiency.

The experiments show that the SHADE algorithm has powerful path optimization and obstacle avoidance skills when facing multi-path challenges in three-dimensional environments. It performs well in terms of path length, obstacle avoidance performance, and computational time, and is particularly suitable for path planning in complex situations. Subsequent research can improve the computational efficiency of the SHADE algorithm and integrate the advantages of other algorithms to achieve high efficiency and flexibility in path planning.

4. Conclusions

This paper conducts a comprehensive analysis of different path planning algorithms in a three-dimensional environment, and deeply studies the route analysis of different DE algorithms. It also analyzes the advantages and disadvantages of adaptive differential evolution algorithm (SHADE), genetic algorithm (GA), particle swarm optimization algorithm (PSO) and classic A* algorithm in solving path planning problems. Finally, it solves the route conflict problem of DE algorithm in multi-path route planning. The research results show that the SHADE algorithm performs very well in optimizing paths and avoiding obstacles, especially in complex three-dimensional environments.

4.1 Main conclusions

The SHADE algorithm has a strong optimization ability in calculating path length. Through multiple studies, it was found that the SHADE method consistently produces shorter path lengths compared to other DE algorithms and commonly used path-finding algorithms. This demonstrates its advantage in effectively identifying the shortest path. This advantage is also confirmed in multi-path planning problems. The SHADE method is able to find a shorter path, whether it involves single or multi-path planning. The SHADE algorithm has excellent global search capabilities and successfully alleviates the problem of local optimal solutions (Ghambari et al., 2020, p. 56).

And obstacle avoidance is an important evaluation criterion for path planning algorithms. The SHADE algorithm effectively avoided obstacles without any collisions in all the trials conducted, thus successfully demonstrating its strong ability to navigate in complex environments (Storn & Price, 1997, p. 85). In contrast, genetic algorithms (GA) and particle swarm optimization algorithms (PSO) do not perform well in avoiding obstacles

(Goldberg, 1989, p. 123; Kennedy & Eberhart, 1995, p. 195).

The practical implementation of path planning algorithms must take computational efficiency into account as a key component. Empirical results show that although the SHADE method takes slightly longer to compute than the A* algorithm, it can produce better path quality and has stronger obstacle avoidance capabilities. When comparing the SHADE algorithm with the genetic algorithm (GA) and the particle swarm optimization algorithm (PSO), the SHADE algorithm does not have a significant difference in computational time. However, it outperforms the GA and PSO algorithms in path optimization and obstacle avoidance (Eberhart & Shi, 2001, p. 114), so the SHADE method successfully strikes a good balance between computational efficiency and path quality, making it very suitable for applications that require accurate path planning.

The multi-path planning problem poses a significant obstacle in route planning, necessitating the algorithm's capability to simultaneously plan numerous pathways and prevent conflicts between them. The SHADE algorithm excels at multi-path planning. The SHADE method is capable of efficiently generating several collision-free pathways under various beginning and ending point configurations. Furthermore, each path is comparatively optimized (Nikolos et al., 2003, p. 899). The SHADE algorithm demonstrates its capacity to handle intricate path planning issues, showcasing its versatility and resilience.

Through a comparative analysis of algorithm performance in many experimental circumstances, it is evident that the SHADE algorithm excels in path optimization, obstacle avoidance, and computing economy. The remarkable flexibility and resilience of this system in complicated three-dimensional situations make it an excellent option for solving path planning challenges. Despite the computational efficiency benefits of the A* method, it tends to produce somewhat lengthy paths and experiences a large increase in computation in high-dimensional situations. While the genetic algorithm (GA) and the particle swarm optimization algorithm (PSO) possess some ability to search globally, they are inadequate when it comes to avoiding obstacles and optimizing paths (Goldberg, 1989, p. 123; Kennedy & Eberhart, 1995, p. 195).

4.2 Future Developments

Although the SHADE algorithm shows good performance, the choice of its parameter values significantly affects the overall effect of the method. Subsequent research can enhance the parameter configuration of the SHADE method by using an automatic parameter modification system to improve its computational efficiency and route quality. Combining the benefits of other optimization algorithms, like the heuristic search strategy of the A* algorithm or the mutation and crossover mechanism of genetic algorithms, can help the SHADE algorithm become more robust and efficient in path planning (Beard et al., 2002, p. 762; Mettler et al., 2003, p. 104).

This study was mainly tested in a static environment. Subsequent investigations can use the SHADE algorithm in a dynamic environment to evaluate its effectiveness in dealing with dynamic obstacles and performing real-time path planning. Combined with feedback from practical applications, the method can be iteratively enhanced and adjusted to suit a variety of application scenarios.

Hence, our work has validated the efficacy of the SHADE algorithm in three-dimensional route planning via rigorous experimentation and diligent data analysis. The algorithm demonstrates exceptional performance in optimizing routes, circumventing obstructions, and conducting efficient calculations, making it an outstanding option for resolving intricate path planning issues. Further study will enhance the effectiveness of the SHADE algorithm and expand its range of applications to address challenges in a wider range of real-world situations. The successful implementation of the SHADE algorithm not only offers innovative solutions in the realm of path planning, but also presents a fresh avenue for the progress of technologies like intelligent navigation and autonomous driving (Storn & Price, 1997, p. 85; Tanabe & Fukunaga, 2013, p. 76).

References

Beard, R.W., McLain, T.W., Goodrich, M.A. and Anderson, E.P. (2002). Coordinated Target Assignment and Intercept for Unmanned Air Vehicles. IEEE Transactions on Robotics and Automation, 18(6), pp. 911-922.

Brintaki, A.N. and Nikolos, I.K. (2005). Coordinated UAV Path Planning Using Differential Evolution. Operational Research: An International Journal, 5(3), pp. 487-502.

Ghambari, S., Lepagnot, J., Jourdan, L., and Idoumghar, L. (2020). UAV Path Planning in the Presence of Static and Dynamic Obstacles.

Mettler, B., Tischler, M.B., and Kanade, T. (2003). System identification of a small-scale unmanned rotorcraft for flight control design. Journal of the American Helicopter Society, 47(1), pp. 50-63.

Nikolos, I.K., Valavanis, K.P., Tsourveloudis, N.C., and Kostaras, A. (2003). Evolutionary Algorithm based offline/online path planner for UAV navigation. IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics, 33(6), pp. 898-912.

Storn, R. and Price, K. (1997). Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. Journal of Global Optimization, 11, pp. 341-359.

Eberhart, R.C., & Shi, Y. (2001). Particle swarm optimization: developments, applications and resources. Proceedings of the 2001 Congress on Evolutionary Computation, 1, pp. 81-86.

Goldberg, D.E. (1989). Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley.

Hart, P.E., Nilsson, N.J., & Raphael, B. (1968). A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE Transactions on Systems Science and Cybernetics, 4(2), pp. 100-107.

Kennedy, J., & Eberhart, R.C. (1995). Particle swarm optimization. Proceedings of ICNN'95 - International Conference on Neural Networks, 4, pp. 1942-1948.

R. Tanabe and A. Fukunaga, "Success-history based parameter adaptation for Differential Evolution," 2013 IEEE Congress on Evolutionary Computation, Cancun, Mexico, 2013, pp. 71-78, doi: 10.1109/CEC.2013.6557555.