

Article

A Cooperative Game Hybrid Optimization Algorithm Applied to UAV Inspection Path Planning in Urban Pipe Corridors

Chuanyue Wang, Lei Zhang *, Yifan Gao, Xiaoyuan Zheng and Qianling Wang

School of Artificial Intelligence and Data Science, Hebei University of Technology, Tianjin 300132, China

* Correspondence: 2007094@hebut.edu.cn

Abstract: This paper proposes an improved algorithm applied to path planning for the inspection of unmanned aerial vehicles (UAVs) in urban pipe corridors, which introduces a collaborative game between spherical vector particle swarm optimization (SPSO) and differential evolution (DE) algorithms. Firstly, a high-precision 3D grid map model of urban pipe corridors is constructed based on the actual urban situation. Secondly, the cost function is formulated, and the constraints for ensuring the safe and smooth inspection of UAVs are proposed to transform path planning into an optimization problem. Finally, a hybrid algorithm of SPSO and DE algorithms based on the Nash bargaining theory is proposed by introducing a cooperative game model for optimizing the cost function to plan the optimal path of UAV inspection in complex urban pipe corridors. To evaluate the performance of the proposed algorithm (GSPSODE), the SPSO, DE, genetic algorithm (GA), and ant colony optimization (ACO) are compared with GSPSODE, and the results show that GSPSODE is superior to other methods in UAV inspection path planning. However, the selection of algorithm parameters, the difference in the experimental environment, and the randomness of experimental results may affect the accuracy of experimental results. In addition, a high-precision urban pipe corridors scenario is constructed based on the RflySim platform to dynamically simulate the optimal path planning of UAV inspection in real urban pipe corridors.

Keywords: UAV inspection; path planning; spherical vector particle swarm optimization; differential evolution; Nash bargaining theory; urban pipe corridors

MSC: 93C85



Citation: Wang, C.; Zhang, L.; Gao, Y.; Zheng, X.; Wang, Q. A Cooperative Game Hybrid Optimization Algorithm Applied to UAV Inspection Path Planning in Urban Pipe Corridors. *Mathematics* **2023**, *11*, 3620. <https://doi.org/10.3390/math11163620>

Academic Editor: Xiaosong Du

Received: 22 July 2023

Revised: 15 August 2023

Accepted: 17 August 2023

Published: 21 August 2023



Copyright: © 2023 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

Urban comprehensive pipe corridors are the core infrastructure supporting the normal operation of the city, which effectively improves the real problems of “road zipper” and “aerial spider web” in the process of urban construction and brings a lot of convenience to the daily life of urban residents while improving the aesthetics of the modernized city. It can be said that urban comprehensive pipe corridors are equivalent to the blood vessels for the city to transport vitality to ensure that the energy flow and material flow of the modern city can run more smoothly. However, since the corridors are easily affected by many factors in the process of operation, it is necessary to actively implement the operation and maintenance work based on realistic requirements. However, in addition to municipal pipes in comprehensive pipe corridors, there are other additional auxiliary facilities, such as fire alarms; fire, monitoring, and the sewage environment are very complex. The traditional mode of operation and maintenance management work mostly rely on the manual inspection form, but because of the large scale of the corridors, which contain a few “inverted rainbow” nodes, this greatly increases the difficulty of manual inspection, not only threatening the personal safety of the inspectors but also being unable to guarantee the accuracy of the inspection results [1].

As we all know, UAVs have the advantages of small size, simple mechanical structure, high safety, low cost, and more, which can be used as an unmanned inspection force to

inspect narrow indoor spaces and pipe corridors to effectively improve the inspection efficiency and solve the problems of low efficiency, high labor cost, high labor intensity, and low safety [2–4]. Facing the intricate urban pipe corridors, the optimal path planning of UAVs is a necessary condition to ensure the safe and successful implementation of the inspection process; therefore, this paper proposes to research the path planning of UAV inspection, aiming to obtain the optimal flight path of UAV inspection in pipe corridors to provide the theoretical support for safety and accurate inspection.

UAV inspection path planning means that UAV inspection plans a flight path from the starting point to the target point, as well as satisfying the safety, path distance, fuel consumption, and other constraints, which is crucial for the smooth execution of inspection [5,6]. Generally, the optimal path refers to minimizing the distance between UAV visit locations to reduce the time and fuel. Furthermore, the planned path is required to guide UAVs to avoid obstacles safely and effectively in the corridors and to meet feasibility constraints, such as flight time, flight altitude, fuel consumption, turn rate, and climb angle [7–9].

In recent years, researchers have proposed various methods for addressing the issue of path planning, including graph search methods [10–12], linear programming methods [13], and heuristic methods [14–17]. The PSO algorithm is a kind of heuristic bionic optimization algorithm, which does not require the optimized function to have the properties of differentiable, derivable, continuous, etc., and it can converge to the global optimal solution with a large probability, which has a faster computational speed and a better global search ability compared with traditional optimization algorithms. However, for functions with multiple local extreme points, it is easy to fall into the local extreme points, resulting in premature convergence. The PSO algorithm is generally applicable to a class of high-dimensional optimization problems with multiple local extreme points, which do not need to be solved with very high accuracy. Therefore, researchers proposed many improved algorithms, such as adaptive optimization algorithms, hybrid optimization algorithms, multi-objective optimization algorithms, and so on. Liu, H. et al. proposed a modified PSO called MPSO. A chaos-based non-linear inertia weight is used to balance capacities better [18]. Shin, J.-J. et al. [19] proposed an improved particle swarm optimization (PSO) algorithm used for finding an optimal path, which is composed of pre-processing steps, multi-swarm PSO algorithm, and post-processing steps [19]. Phung, M. et al. proposed a new spherical-vector-based particle swarm optimization algorithm (SPSO) by introducing the correspondence between particle position and speed, turn angle, and pitch angle, and they evaluated it by comparing it with other optimization algorithms in eight benchmark test scenarios for validation. The results showed that SPSO can efficiently search for the configuration space of UAV, which outperforms the other particle swarm optimization (PSO) variables and other most advanced meta-heuristic optimization algorithms [20]. Xu proposed a tri-objective optimization problem for planning UAV paths, which considers the path's length, height, and turning angle and used a multi-objective PSO algorithm (MO-PSO) to solve it [21]. Among the many versions of improved algorithms, the hybrid particle swarm algorithm (HPSO), which is formed by combining the advantages of other algorithms, is the most favored form of improved algorithm by researchers at present. Huo et al. proposed a hybrid differential symbiotic organism search (HDSOS) algorithm, which was proposed by combining the mutation strategy of differential evolution (DE) with the modified techniques of symbiotic organism search (SOS) to solve the path planning problem [22]. Abhishek et al. proposed a hybrid path planning algorithm, which combines PSO with the harmony search algorithm (HSA), which performs well regarding path selection, obstacle avoidance, and path length minimization [23]. Chen et al. proposed an enhanced version of the chimp optimization algorithm (TRS-ChOA) to solve the UAV path planning problem in a 3D environment [24]. Hao et al. proposed an improved artificial potential field method and introduced a collision risk assessment mechanism to solve the problems of local minimums, unreachable targets, and unreasonable obstacle avoidance

techniques in traditional artificial potential field method path planning [25]. A comparative analysis of related work is shown in Table 1.

Table 1. Comparative analysis of related work.

Reference	Contributions	Limitation
Hao, L. et al. (2020) [18]	This paper proposes a modified PSO called MPSO. A chaos-based non-linear inertia weight is used to balance capacities better.	Its convergence and stability lack theoretical support. There are some parameters in MPSO, which increase the difficulty of problem solving.
Shin, J.-J. et al. (2020) [19]	An improved particle swarm optimization (PSO) algorithm is proposed for finding an optimal path, which is composed of pre-processing steps, multi-swarm PSO algorithm, and post-processing steps.	It is not suitable to include constraints related to UAV maneuvering; therefore, this may lead to large errors between the planned path and the flight path.
Phung, M. et al. (2021) [20]	This paper presents a new algorithm named spherical-vector-based particle swarm optimization (SPSO) algorithm, which efficiently searches the configuration space of the UAV via the correspondence between the particle position and the speed, turn angle, and climb/dive angle of the UAV.	The optimization algorithm converges quickly and easily falls into local optimum.
Dadvar, M. et al. (2021) [26]	This paper proposes a new algorithm, which uses a coalition or cooperation model in the game theory to combine the DE and PSO algorithms to maintain a balance between the exploration and exploitation capabilities by preventing population stagnation and avoiding the local optimum.	Failure to incorporate the maneuvering characteristics of the UAV into the algorithm does not further improve its navigation capabilities.
This work	<ol style="list-style-type: none"> 1. A path planning scheme for UAV inspection in urban pipe corridors is proposed. 2. Considering all grids in the corridors as obstacles, the safety cost function is formulated for the corridor grid map model. 3. A hybrid algorithm of SPSO algorithms, which incorporate the maneuvering characteristics of the UAV and DE algorithms based on the Nash bargaining theory, is proposed by introducing a Nash bargaining cooperation game model, which dramatically increases the searching ability of the algorithms. 4. A high-precision urban pipe corridor 3D grid map model and scenario based on the RflySim platform are constructed as an experimental verification environment. 	

Based on the above comparison table, both SPSO and DE algorithms have their own drawbacks and thus cannot show better performance. The SPSO algorithm introduced for the UAV path planning problem is developed based on the correspondence between the intrinsic motion components of the UAV and the search space, but it still suffers from a tendency to converge prematurely and fall into a local optimum when solving complex optimization problems, whereas the DE algorithm relies heavily on the test vector generation strategy and the parameter values used. Above all, this paper proposes a path planning scheme for UAV inspection in urban pipe corridors, which proposes a hybrid optimization algorithm GSPSODE combining the SPSO and DE algorithms by applying a cooperative game model, which can improve the searchability and diversity of the algorithm, as well as finding the best solution, iteratively optimizing the cost function and planning the optimal inspection path of the UAV [26]. Therefore, the contributions of this research are four-fold:

- (i) A cost function is formulated to transform path planning into an optimization problem. Considering all grids in the corridors as obstacles, the safety cost function is formulated for the corridor grid map model.
- (ii) A hybrid algorithm of SPSO and DE algorithms based on the Nash bargaining theory is proposed by introducing a Nash bargaining cooperation game model, which dramatically increases the searching ability of the algorithms.

- (iii) SPSO, PSO, and DE are introduced in comparative analysis experiments to evaluate the effectiveness of the GSPSODE algorithm.
- (iv) A high-precision urban pipe corridor 3D grid map model and scenario based on the RflySim platform are constructed as an experimental verification environment for UAV inspection path planning schemes to evaluate the effectiveness and feasibility of the path planning scheme generated by the GSPSODE algorithm.

2. Problem Formulation and Model Construction

This paper proposes an urban UAV inspection path planning scheme toward multiple inspection points, which replaces the manual inspection method in the traditional mode with UAV inspection. The first step is to construct a high-precision 3D grid map model of urban pipe corridors using octree and to plan the location information of the inspection points according to the actual environment of urban pipe corridors. Afterward, the cost function is formulated, which contains the optimal criteria and constraints associated with UAV. The optimal solution of the cost function is received through optimization, meaning the optimal inspection path planned for the UAV, as described below.

2.1. Map of Urban Pipe Corridors

This research is based on urban pipe corridors as the target inspection environment. The 3D point cloud urban pipe corridor map is obtained with a 3D laser scanner, and then, the 3D grid map model is constructed accordingly. The map information can generally be obtained in 3D point cloud format by sensors such as LiDAR and depth cameras, and the point cloud map can be transformed into an occupied grid map stored in an octree data structure when a certain spatial resolution is set. Each tree node represents a spatial cube, which can realize the compression of a point cloud map, and this research applies this type of map to UAV path planning [27]. The 3D point cloud urban pipe corridor map model is obtained, as shown in Figure 1.

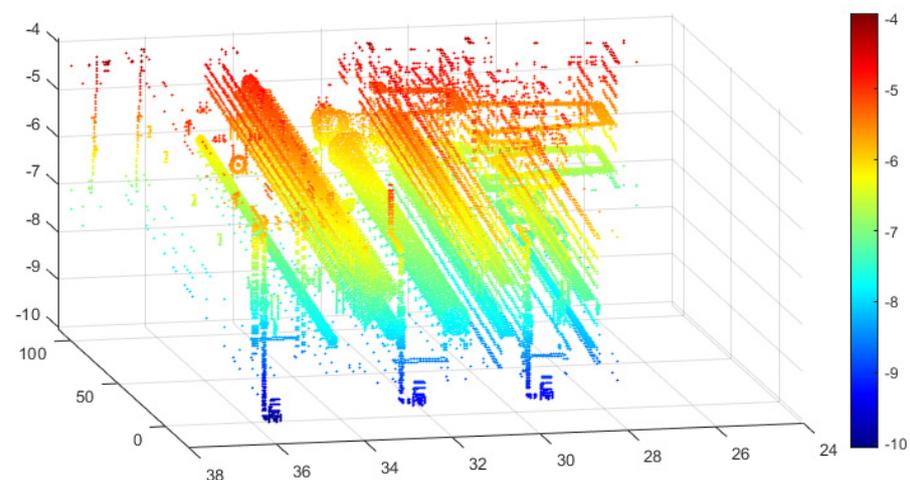


Figure 1. Three-dimensional point cloud urban pipe corridor map model.

An octree is a data structure based on the tree hierarchy, which is defined as, except for the empty tree, any node in the tree having exactly 8 or 0 children. The octree makes full use of the spatial correlation of the forms, and it takes up far less storage space than direct storage through an array of three-dimensional voxels. Its main advantage is convenient management, and it can be convenient to search for a certain small square using the dichotomous method. Once the depth reaches a certain level, it could almost fit all the 3D models [28]. An example diagram representing the space occupied by an octree is shown in Figure 2. The fully occupied squares in Figure 2a are defined in black, the partially occupied ones in gray, and the unoccupied ones in white, respectively. The structure of an octree is shown in Figure 2b.

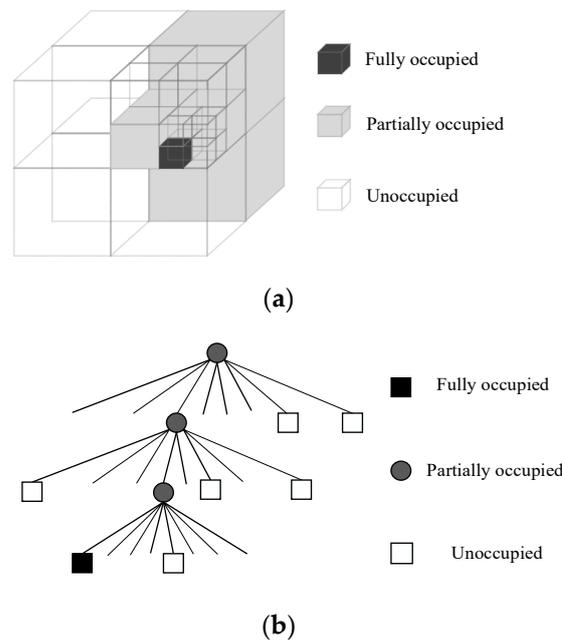


Figure 2. Octree storage example diagram. (a) Three-dimensional space occupation diagram; (b) Octree storage structure.

The transformation of point cloud data into a 3D grid map with an octree structure not only saves storage space significantly without losing the spatial structure but also improves computational speed, which contributes to the environment perception of UAV inspection. The high-precision urban pipe corridor 3D grid map model is shown in Figure 3.

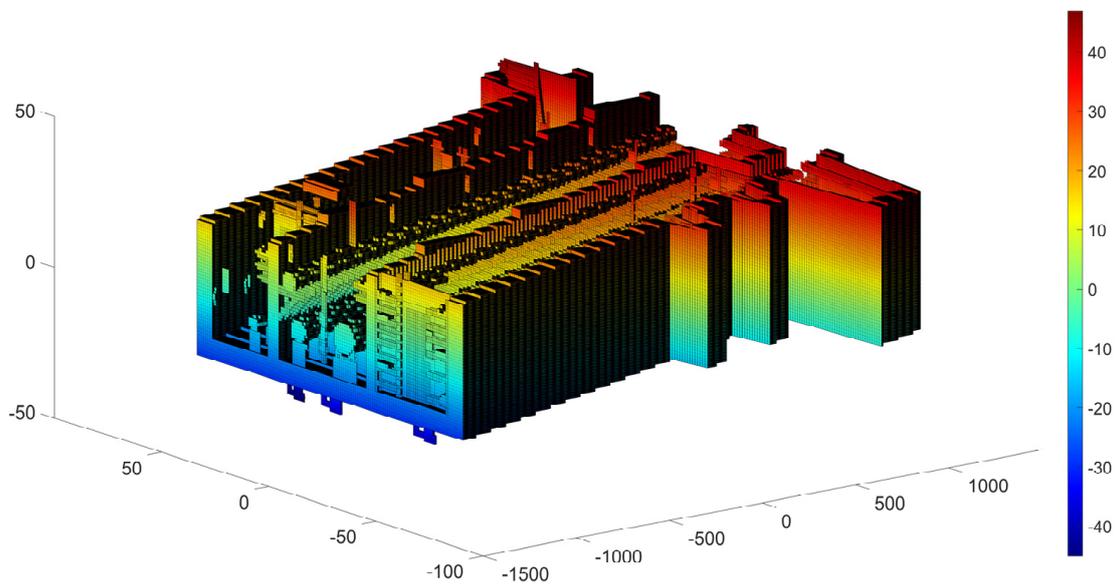


Figure 3. High-precision urban pipe corridor 3D grid map model.

2.2. Constraints

To ensure smooth and efficient inspection of UAVs in urban corridors, the planned optimal path must satisfy multiple constraints, such as path length, safety, feasibility, smoothing cost, etc. The cost function to be optimized is constructed based on multiple constraints, as follows:

(I) Path length constraints

The flight path of the UAV is represented by a list of n waypoints, each of which corresponds to a path node $P_{ij} = (x_{ij}, y_{ij}, z_{ij})$ in the urban pipe corridor grid map. The Euclidean distance between two path nodes is computed as $\|\overrightarrow{P_{ij}P_{i,j+1}}\|$, which results in the cost function $F_1(X_i)$ associated with the path length:

$$F_1(X_i) = \sum_{j=1}^{n-1} \|\overrightarrow{P_{ij}P_{i,j+1}}\|. \tag{1}$$

(II) Safety and feasibility constraints

In addition to path length, the planned paths must avoid obstacles in urban pipe corridors to ensure the safety of UAV inspection. Figure 4 shows the Euclidean distance from each path point to its $l^3 - 1$ neighboring grids, which is used to judge whether the neighboring obstacles are “too close”, where l represents the safety distance between the path node and the obstacle. Each grid in the corridor map model is considered an obstacle, and the entire 3D corridor grid map model will be traversed to judge whether it is located within the $l^3 - 1$ neighboring grids of any path point or whether the path is “too close” to the grid map model, which will produce certain costs $F_2(X_i)$, as follows:

$$F_2(X_i) = k \sum_{i=1}^l d(X_i). \tag{2}$$

where k is the penalty coefficient used to control the extent of overclose behavior.

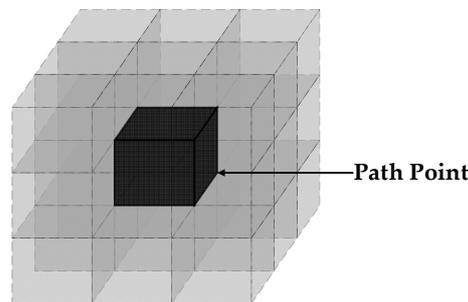


Figure 4. The $l^3 - 1$ neighboring grids of the path point.

(III) Smooth constraints

The smoothing cost evaluates the turning rate and climbing rate necessary to generate feasible paths to reduce the number of turning points in UAV paths. As shown in Figure 5, the turning angle, ϕ_{ij} , is the angle between two consecutive path segments, $\overrightarrow{P'_{ij}P'_{i,j+1}}$ and $\overrightarrow{P'_{i,j+1}P'_{i,j+2}}$, projected on the horizontal plane O_{xy} . Let \vec{k} be the unit vector in the direction of the z-axis, and let the climbing angle ψ_{ij} be the angle between the path segment $\overrightarrow{P_{ij}P_{i,j+1}}$ and its projection $\overrightarrow{P'_{ij}P'_{i,j+1}}$ onto the horizontal plane. The projected vector $\overrightarrow{p'_{ij}p'_{i,j+1}}$, the turning angle ϕ_{ij} , and the climbing angle ψ_{ij} are computed as shown in Equations (3)–(5).

$$\overrightarrow{p'_{ij}p'_{i,j+1}} = \vec{k} \times \left(\overrightarrow{p'_{ij}p'_{i,j+1}} \times \vec{k} \right), \tag{3}$$

$$\phi_{ij} = \arctan \left(\frac{\|\overrightarrow{P'_{ij}P'_{i,j+1}} \times \overrightarrow{P'_{i,j+1}P'_{i,j+2}}\|}{\overrightarrow{P'_{ij}P'_{i,j+1}} \cdot \overrightarrow{P'_{i,j+1}P'_{i,j+2}}} \right), \tag{4}$$

$$\psi_{ij} = \arctan \left(\frac{z_{i,j+1} - z_{ij}}{\| \vec{P}'_{ij}; P'_{i,j+1} \|} \right). \tag{5}$$

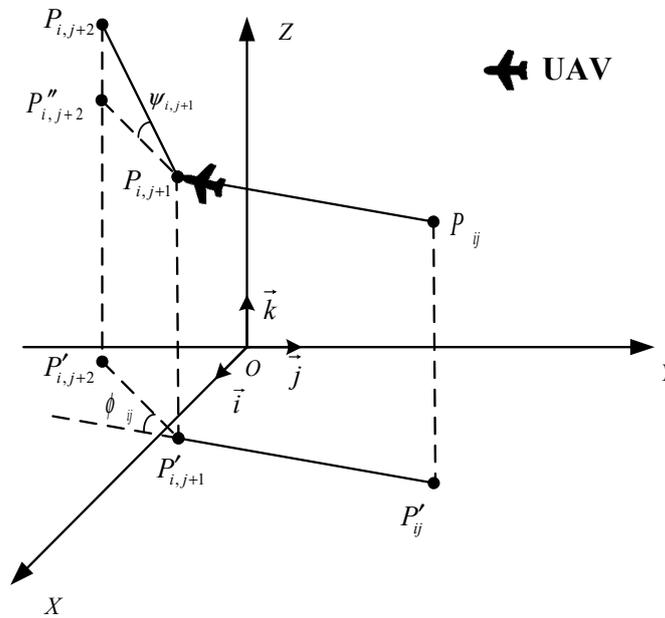


Figure 5. Turning and climbing angle calculation.

As shown above, the smooth cost $F_3(X_i)$ is computed as

$$F_3(X_i) = a_1 \sum_{j=1}^{n-2} \phi_{ij} + a_2 \sum_{j=1}^{n-1} |\psi_{ij} - \psi_{i,j-1}|. \tag{6}$$

where a_1 and a_2 are, respectively, the penalty coefficients of the turning and climbing angles.

2.3. Overall Cost Function

By considering the optimality, safety, and feasibility constraints associated with path X_i , the cost function $F(X_i)$ can be defined as

$$F(X_i) = \sum_{k=1}^3 b_k F_k(X_i). \tag{7}$$

where b_k is the weight coefficient, and $F_1(X_i)$ to $F_3(X_i)$ are, respectively, the costs associated with path length, safety and feasibility restrictions, and smoothness. X_i is the decision variable, including the list of n waypoints.

3. GSPSODE for UAV Path Planning

With the cost function $F(X_i)$ defined above, path planning becomes an optimization problem, where the aim is to find the path X^* , which minimizes $F(X_i)$. This section proposes a new algorithm for solving optimization problems—a hybrid algorithm of SPSO and DE algorithms using the cooperation model from the game theory to maintain a balance between the exploration and development capacity by preventing population stagnation and avoiding the local optimum.

3.1. Nash Bargaining Cooperation Game Model

The Nash bargaining cooperation game aims to bargain for the income generated by the cooperation to determine the income distribution scheme. When bargaining on an

issue, two or more parties negotiate an agreement and solution and act on it. How the cooperation profits are distributed between the parties depends on the balance of power and the use of skills of the parties. Therefore, it is essential to go through the game of all parties bargaining and reaching a consensus and cooperation.

SPSO and DE are two players in the state space, which use the Nash bargaining theory to play a cooperative game to find the best solution. The bargaining problem is formed by the pair (F, v) , where F is a feasible set of allocations, and a closed and convex subset of R^2 , $v = (v_1, v_2)$ is the disagreement point determining the outcome of the two players if the negotiations have failed. It should be noted that the feasible set F always contains the disagreement point. Nash presents a unique solution to the bargaining theory based on five principles, including Pareto optimality, individual rationality, linear invariance, independence of irrelevant alternatives, and symmetry.

$$\begin{aligned}
 f(F, v) \in \operatorname{argmax}((x_1 - v_1)(x_2 - v_2)), \\
 (x_1, x_2) \in F, \\
 x_1 \geq v_1, x_2 \geq v_2.
 \end{aligned}
 \tag{8}$$

3.2. Spherical Vector Particle Swarm Optimization (SPSO)

(I) SPSO is one of the PSO variants, which is created by introducing a spherical vector coordinate system. Each path is encoded as a set of vectors used to describe the movement of the UAV between neighboring waypoints, as represented by the three components of magnitude ρ , elevation ψ , and azimuth ϕ , corresponding to the UAV's speed, turn angle, and climb angle. The flight path Ω_i is represented by a $N \times 3D$ hyper spherical vector consisting of N nodes. The SPSO algorithm where Ω_i is the position of the particle, and then, $\Delta\Omega_i$ is the velocity increase in that particle, is shown in Equation (9):

$$\begin{aligned}
 \Omega_i &= (\rho_{i1}, \psi_{i1}, \phi_{i1}; \rho_{i2}, \psi_{i2}, \phi_{i2}; \dots; \rho_{iN}, \psi_{iN}, \phi_{iN}), N = n - 2, \\
 \Delta\Omega_i &= (\Delta\rho_{i1}, \Delta\psi_{i1}, \Delta\phi_{i1}; \Delta\rho_{i2}, \Delta\psi_{i2}, \Delta\phi_{i2}; \dots; \Delta\rho_{iN}, \Delta\psi_{iN}, \Delta\phi_{iN}), \\
 \rho &\in (0, \text{path_length}), \\
 \psi &\in (-\pi/2, \pi/2), \\
 \phi &\in (-\pi, \pi).
 \end{aligned}
 \tag{9}$$

(II) Consider each path as a volume-less particle in a $N \times 3D$ search space, flying in the search space at a certain speed, which is dynamically adapted according to its own flight experience and that of its companions. The i th particle is denoted as $\Omega_i = (u_{i1}^k, u_{i2}^k, \dots, u_{ij}^k)$, and the best position it experiences is denoted as a $1 \times N$ -dimensional vector $Q_{i,best}$. The best position experienced by all particles in the population is denoted as a $1 \times N$ -dimensional vector $Q_{g,best}$. u_{ij}^k and Δu_{ij}^k are, respectively, the $1 \times 3D$ spherical vector $(\rho_{ij}^k, \psi_{ij}^k, \phi_{ij}^k)$ and velocity $(\Delta\rho_{ij}^k, \Delta\psi_{ij}^k, \Delta\phi_{ij}^k)$. For each generation, the update is performed according to the following Equation (10):

$$\begin{aligned}
 \Delta u_{ij}^{k+1} &\leftarrow w^k \Delta u_{ij}^k + \eta_1 r_{1i} (q_{ij}^k - u_{ij}^k) + \eta_2 r_{2j} (q_{gj}^k - u_{ij}^k), \\
 u_{ij}^{k+1} &\leftarrow u_{ij}^k + \Delta u_{ij}^{k+1}, \\
 Q_{i,best} &= (q_{i1}, q_{i2}, \dots, q_{iN}), \\
 Q_{g,best} &= (q_{g1}, q_{g2}, \dots, q_{gN}).
 \end{aligned}
 \tag{10}$$

- (III) To determine $Q_{i,best}$ and $Q_{g,best}$, it is necessary to map the flight path Ω_i in the spherical vector coordinate system to path X_i used for evaluating the costs. The mapping $\xi: \Omega \rightarrow X$ of vector $u_{ij} = (\rho_{ij}, \psi_{ij}, \phi_{ij}) \in \Omega_i$ to waypoint $P_{ij} = (x_{ij}, y_{ij}, z_{ij}) \in X_i$ and the local and global best positions can be computed as Equations (11) and (12):

$$\begin{aligned} x_{ij} &= x_{i,j-1} + \rho_{ij} \sin \psi_{ij} \cos \phi_{ij}, \\ y_{ij} &= y_{i,j-1} + \rho_{ij} \sin \psi_{ij} \sin \phi_{ij}, \\ z_{ij} &= z_{i,j-1} + \rho_{ij} \cos \psi_{ij}. \end{aligned} \tag{11}$$

$$Q_{i,best} = \begin{cases} \Omega_i & \text{if } F(\xi(\Omega_i)) < F(\xi(Q_{i,best-1})) \\ Q_{i,best-1} & \text{otherwise} \end{cases}, \tag{12}$$

$$Q_{g,best} = \underset{Q_{i,best}}{\operatorname{argmin}} F(\xi(Q_i)).$$

3.3. Differential Evolution (DE)

Like the genetic algorithm, the differential evolution (DE) [29,30] algorithm is an optimization algorithm based on the modern intelligence theory, which guides the direction of the optimization search through the group intelligence generated by cooperation and competition among individuals within a population. It continuously evolves to retain suitable individuals and eliminate inferior ones, guiding the search for the optimal solution.

- (I) First, a mutation operation is performed to select two individuals from among the parent individuals to generate a difference vector, which is scaled by a scale factor and then summed with the base vector X_{i1} to generate the experimental individual.

$$U_i = X_{i1} + F(X_{i2} - X_{i3}) \tag{13}$$

where X_{i1} is the base vector, and X_{i2} and X_{i3} are the two other parents, which are selected randomly from the population, such that $i \neq i_1 \neq i_2 \neq i_3$. F is the scaling factor—a positive constant amplifying the difference vector.

- (II) Then, crossover operations are performed on the parent individuals and the corresponding experimental individuals to generate new offspring individuals. Common types of crossovers are binomial and exponential, etc.

$$X'_i = \begin{cases} U_i & \text{if } r_j \leq CR \text{ or } j = j_{rand} \\ X_i & \text{otherwise} \end{cases} \tag{14}$$

where $CR \in [0, 1]$ is the crossover rate with a prescribed constant; r_j is a uniform random number within the range $[0, 1]$; and j_{rand} is a randomly selected integer within the range $[1, D]$.

- (III) Finally, a selection operation is performed between the parent and offspring individuals, and individuals meeting the requirements are saved to the next generation population.

3.4. The Proposed Algorithm

The proposed algorithm improves the search by collaborating through a coalition or cooperation model in the game theory. After a given iteration, the player DE and SPSO algorithms enter the game environment and play cooperative games based on the Nash bargaining theory to achieve the best balance or solution in the state space. The flow chart of the proposed method is given in Figure 6.

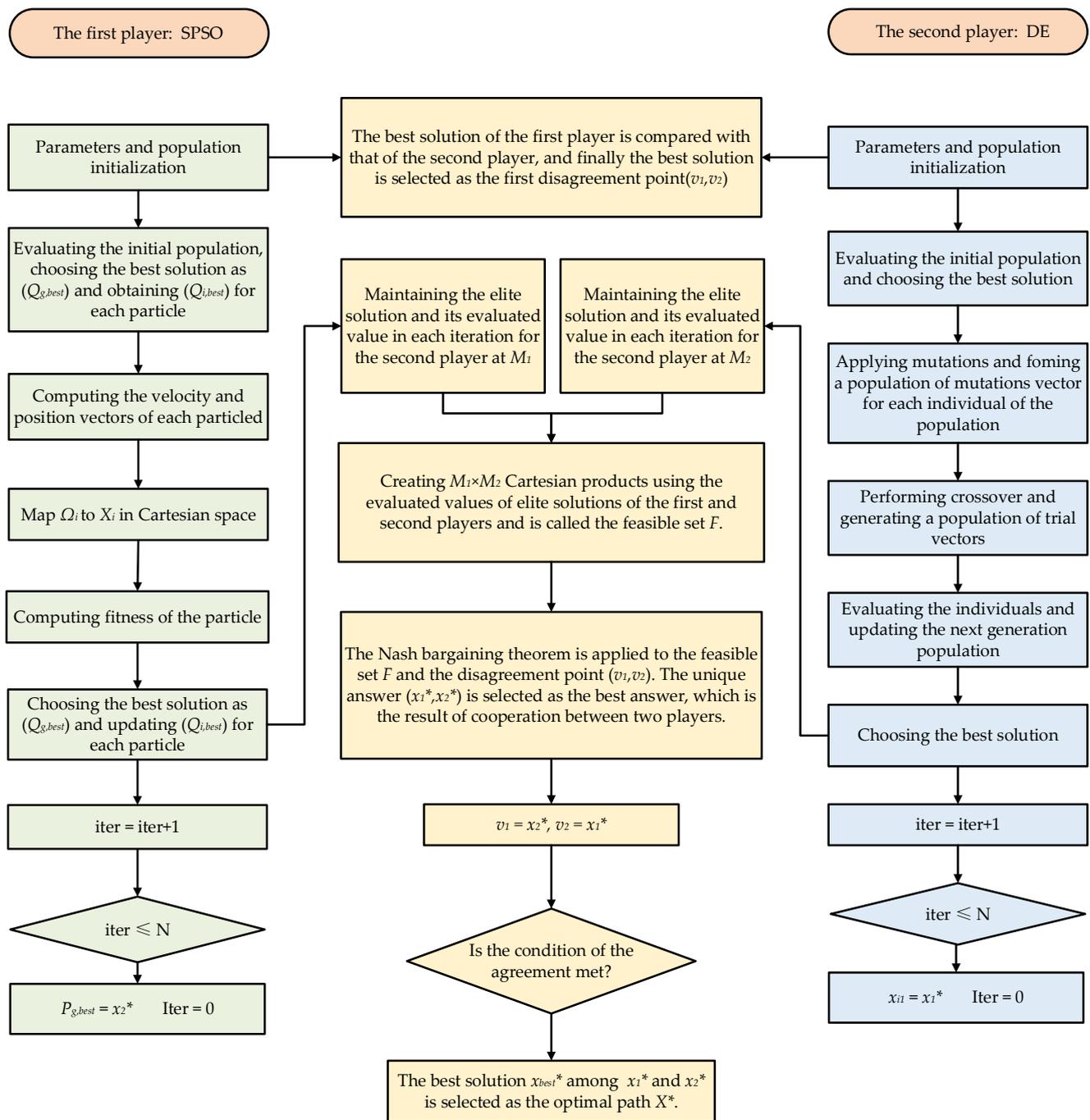


Figure 6. The flow chart of the proposed method.

- (I) The initial populations of the two players, SPSO and DE, are evaluated independently by the cost function, and all solutions of the two players are evaluated to select the best solution for the first and second players.
- (II) The two better solutions are compared, so that the best solution is chosen as the disagreement point (v_1, v_2) in the first round of the game. Since the game has the same conditions for both players and is symmetric, the values are equal in the first round $v_1 = v_2$.
- (III) In each round of the game, the two players each run N times, and in each iteration, the best solutions and their evaluation values are stored for the first player in M_1 and

the second player in M_2 , respectively, which means that N elite solutions are stored in the sets M_1 and M_2 for each iteration:

$$M_1 = \min(pop_1) \quad M_2 = \min(pop_2) \tag{15}$$

- (IV) The feasible set F is the Cartesian product of the best solutions of the two players in N iterations:

$$F = M_1 * M_2 = \{(x_1, x_2) \mid x_1 \in M_1 \text{ and } x_2 \in M_2\} \tag{16}$$

- (V) After N iterations, SPSO and DE enter the game environment and play the cooperative game based on the Nash bargaining theory, and the unique solution $f(F, v) = (x_1^*, x_2^*)$ is selected as the best solution due to the Pareto optimality property.

$$f(F, v) = \operatorname{argmax}((x_1 - v_1)(x_2 - v_2)) \tag{17}$$

$$(x_1^*, x_2^*) = \begin{cases} f(F, v) & \text{otherwise} \\ (v_1, v_2) & v_1 < x_1 \text{ and } v_2 < x_2 \end{cases} \tag{18}$$

- (V) x_1^* is the game profit of the first player, which is used as the base vector in the DE mutation operator in the next iteration, resulting in a change in the DE mutation operator. x_2^* is the second player's game profit, which acts as the lead particle in SPSO to update the particle velocity vector in the next iteration, changing the direction of the particle's motion. In addition, the only solution in this round of Nash bargaining (x_1^*, x_2^*) is seen as a point of disagreement (v_1, v_2) in the next round of Nash bargaining.

$$U_i = x_1^* + F(X_{i_2} - X_{i_3}) \tag{19}$$

$$\Delta u_{ij}^{k+1} \leftarrow w^k \Delta u_{ij}^k + \eta_1 r_{1i} (q_{ij}^k - u_{ij}^k) + \eta_2 r_{2j} (x_2^* - u_{ij}^k) \tag{20}$$

$$v_1 = x_2^*, \quad v_2 = x_1^* \tag{21}$$

- (VII) If the conditions of the protocol are satisfied, the best value from the two values (x_1^*, x_2^*) is chosen as the final solution; otherwise, the game is repeated for the next round.

Overall, the Nash bargaining theory helps balance this algorithm's exploration and exploitation capabilities. Choosing (x_1, x_2) from the feasible set F allows a thorough exploration and search of the problem space to find the optimal solution. Moreover, selecting the disagreement points (v_1, v_2) also allows for extracting a better solution.

4. Results and Analysis

To evaluate the performance and application feasibility of GSPSODE, set three pipe corridors with different levels of complexity as the standard scenarios. Digital comparison experiments are conducted to analyze the results. The inspection paths of UAV are dynamically simulated in the high-precision urban pipe corridor scenario constructed based on the RflySim platform, as follows.

4.1. Scenario Setup

The scenarios used for the evaluation are based on the urban pipe corridor 3D grid map model constructed above, and three benchmarking scenarios with different levels of complexity are generated. In these scenarios, red circles represent the start point of the UAV path planning; yellow diamonds represent the location of the inspection points, which need to remain; and green circles represent the end point of the UAV path planning.

The estimated circumstances of the problem are stationary. For comparison, five well-established optimization techniques (the SPSO, PSO [19], DE, GA [31], and ACO [32]) were compared with our proposed GSPSODE algorithm. Table 2 shows the parameter settings of all algorithms. For fairness, the population size $P = 500$ for all algorithms, the number of waypoints $N = 25$, and the maximum iterations $I = 500$. All experiments were conducted on MatlabR2019a version. P_c and P_m are crossover operators and mutation operators in the genetic algorithm, and α , β , γ and τ_0 are the pheromone factors, heuristic function factors, guiding function factors, and initial pheromone concentrations in the ant colony algorithm.

Table 2. Parameters used in the experiments.

Algorithm	Parameters
GSPSODE	$MaxIter = 1000, MaxFES = 2000$
SPSO	$\omega = 1, \eta_1 = 1.5, \eta_2 = 1.5$
PSO	$\omega = 1, \eta_1 = 1.5, \eta_2 = 1.5$
DE	$CR = 0.9, F = 0.5, r = 0.98$
GA	$P_c = 0.2, P_m = 0.2$
ACO	$\alpha = 10, \beta = 1, \gamma = 1, \tau_0 = 1.2$

4.2. Comparative Analysis

The top view of the inspection paths generated is shown in Figure 7. All algorithms can generate feasible paths, which satisfy the constraints, but the optimality varies with different scenarios. The GSPSODE algorithm plans the smoothest and shortest paths in most cases, while paths generated by the DE algorithm consist of only a few line segments and sharp turns. In the path planning of three different scenarios, the GSPSODE algorithm has the best performance. Compared with the other algorithms, the path planned by GSPSO is smoother, and the planned path length is the shortest. The GA algorithm has the worst performance, and the planned path is very uneven. Compared with the GA algorithm, the PSO and DE algorithms have a better overall effect, but the planned path is slightly worse when avoiding obstacles. The smoothness of the path planned by the ACO and SPSO algorithms is improved compared with other comparison algorithms, but the turning angle is still not smooth enough when facing the inspection point, and the turning angle is relatively large.

In each comparison, all algorithms are run 10 times to find the mean and standard deviation values. The optimum, mean, and standard deviation values of the fitness obtained by the six algorithms for path planning in three scenarios with different levels of complexity are shown in Table 3. The GSPSODE algorithm shows the best performance in every scenario; optimal fitness values are obtained, and the convergence stability standard deviation value is the smallest, which is because GSPSODE effectively improves the effects of some particles being far from the local optimum and the bad overall convergence of the algorithm. The optimal fitness values of the six algorithms are also affected by the pipe corridors' complexity. For example, Scene 2 has few obstacles, and the regular pipe corridor grid model makes the inspection points easy to recognize, so the smallest fitness values are obtained through iteration. However, many obstacles are set in Scene 3, and the inspection points are all located in narrow places, making path planning more complex. Hence, the fitness values of the planned inspection path are larger.

Figure 8 shows the fitness value iterations of the six algorithms in three scenarios, and the fitness values of each algorithm during the iterations are helpful for further comparative analysis. As can be seen, all algorithms converge well, with slight differences in their minimum fitness values, and GSPSODE can obtain the optimal solution first, and the fitness values of GSPSODE in the three scenarios are lower than those of the other five algorithms, followed by SPSO. In Figure 8a, GSPSODE and ACO converge rapidly early on and converge to an optimal value of about 50 iterations. The comparison with the final results shows that GSPSODE has the highest accuracy, followed by ACO. It is also found that GSPSODE converges to the optimal value very quickly at the beginning of the iteration, which proves that the proposed search strategy can accelerate the convergence speed and

improve the convergence accuracy. In Figure 8b, GSPSODE performs better than other algorithms and rapidly converges to the optimal value. As can be seen in Figure 8c, DE outperforms both GSPSODE and SPSO at the beginning of the iteration. However, as the iteration begins, GSPSODE and SPSO rapidly converge to a minimum value.

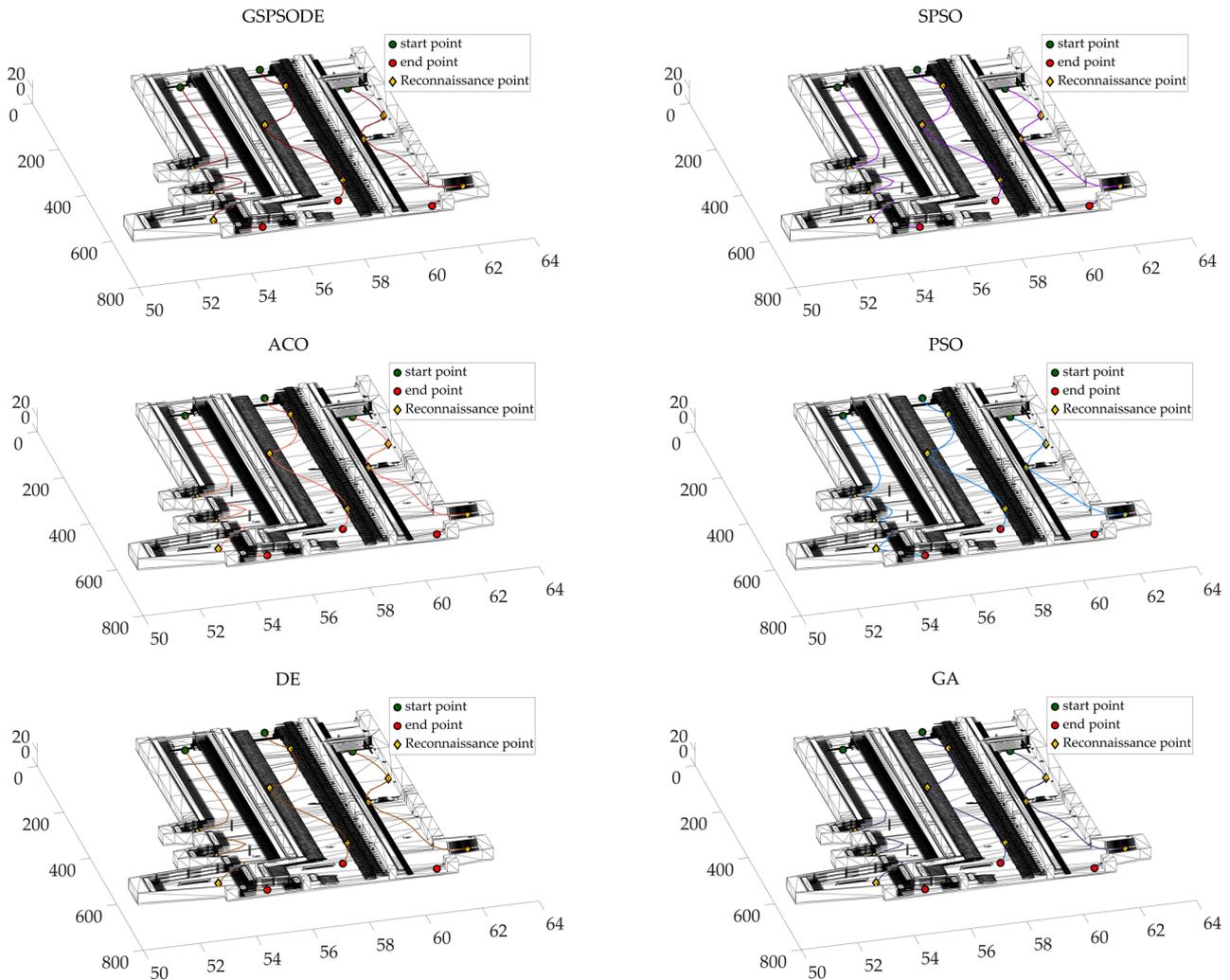


Figure 7. The top view of the inspection paths generated.

4.3. RflySim Simulation

To evaluate the effectiveness of the generated paths for actual UAV operations, a high-precision 3D urban pipe corridor simulation scenario is conducted based on the RflySim platform to dynamically demonstrate the path planning effect of UAV inspection in urban corridors, showing that the GSPSODE algorithm can plan feasible, safe, and optimal paths for UAV inspection.

The RflySim platform is a UAV flight control ecosystem released by the Reliable Flight Control Group at Beihang University. In Ref [33], a comparison between the multi-wing flight entity experiment and the simulation experiment based on the RflySim platform validates the high accuracy level of the platform’s simulation. The platform underwent quantitative analysis tests and comparison experiments with fault injection, and its credibility was above 90%, which fully confirms the high fidelity and practicality of the platform (with 60% representing the lowest value in the accuracy confidence interval).

Table 3. Fitness values of the six algorithms in the three scenarios.

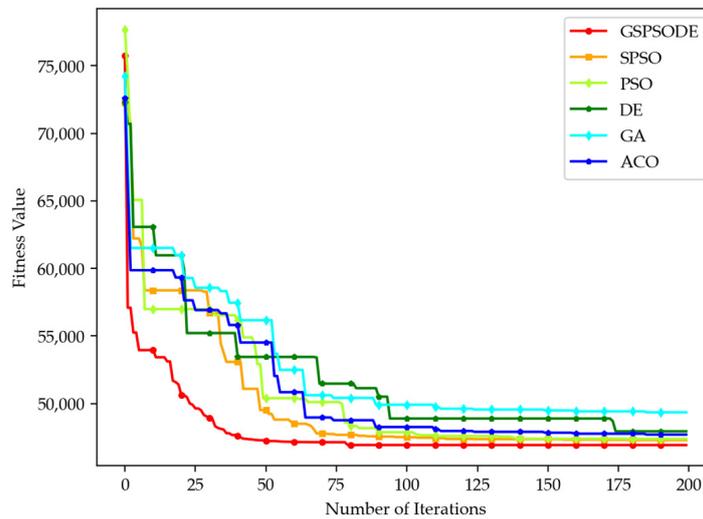
Algorithm	Evaluation Scores	Scene 1	Scene 2	Scene 3
GSPSODE	Best	5.68×10^4	4.69×10^4	6.70×10^4
	Mean	5.70×10^4	4.70×10^4	6.70×10^4
	Std	24.58	35.46	30.77
	Runtimes(s)	151.25	122.16	166.58
SPSO	Best	5.72×10^4	4.72×10^4	6.95×10^4
	Mean	5.73×10^4	4.75×10^4	6.97×10^4
	Std	63.38	88.25	240.13
	Runtimes(s)	102.12	98.31	117.40
PSO	Best	5.95×10^4	4.74×10^4	7.14×10^4
	Mean	5.96×10^4	4.76×10^4	7.16×10^4
	Std	94.23	207.91	274.21
	Runtimes(s)	96.12	81.20	112.64
DE	Best	5.79×10^4	4.79×10^4	7.17×10^4
	Mean	5.80×10^4	4.83×10^4	7.20×10^4
	Std	102.73	132.53	310.13
	Runtimes(s)	80.28	72.61	105.83
GA	Best	6.02×10^4	4.94×10^4	7.20×10^4
	Mean	6.05×10^4	4.97×10^4	7.24×10^4
	Std	120.63	191.38	361.75
	Runtimes(s)	161.19	149.83	180.66
ACO	Best	5.72×10^4	4.77×10^4	7.09×10^4
	Mean	5.75×10^4	4.80×10^4	7.13×10^4
	Std	89.17	62.14	100.62
	Runtimes(s)	79.12	63.13	103.52

The core value of the platform is reflected in the ring simulation of the software and hardware, including the distinctive CopterSim, the visual system plug-in, the developed model, etc. The composition of the software platform is shown in Figure 9.

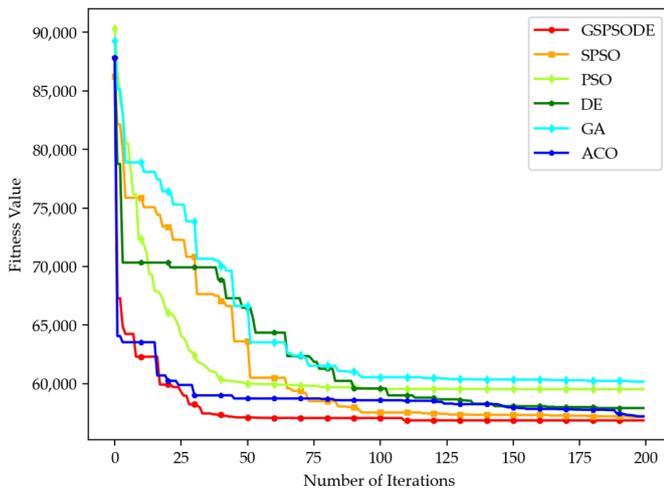
This research constructs a high-precision 3D urban pipe corridor simulation scenario based on the RflySim platform, which contains three urban pipe corridors with different complexities, including stairs, power cables, pipes, communication fiber optic cables, monitoring equipment, firefighting, ventilation equipment, and other obstacles. Compared to the path planning global overview map, the dynamic simulation of UAV inspection path planning in the urban pipe corridor environment can give a strong visual impact. The path planning problem in this case study is static; the initial conditions, such as initial position, inspection stopping point, and termination position, are given; and the grids in the map model of the city pipeline corridor are used as the spatial constraint in the boundary conditions. The optimal inspection path point data information of the UAV is obtained through the optimization algorithm and transmitted to the RflySim platform for dynamic simulation.

Figures 10 and 11 show the path planning effects of stairs and avoiding stairs obstacles in Scenes 1 and 3, respectively. The flight paths planned by the GSPSODE algorithm are collision-free and smooth, and UAV inspection is able to follow the planned paths successfully. The excellent match between the planned paths and the flight paths demonstrates the effectiveness of GSPSODE and the accuracy of the RflySim platform. In Scene 1, the UAV needs to inspect a staircase, and GSPSODE is able to generate a path, which smoothly navigates the stairs, avoiding any collisions with the walls or the stairs themselves. The UAV is able to follow the planned path accurately, ensuring a thorough inspection of the staircase. In Scene 3, the UAV needs to avoid the stairs obstacle, and GSPSODE is able to generate a path, which safely navigates around the obstacle while still ensuring a smooth and efficient flight path. The UAV is able to follow the planned path successfully, avoiding any collisions with the obstacle. The results demonstrate the effectiveness of GSPSODE in generating

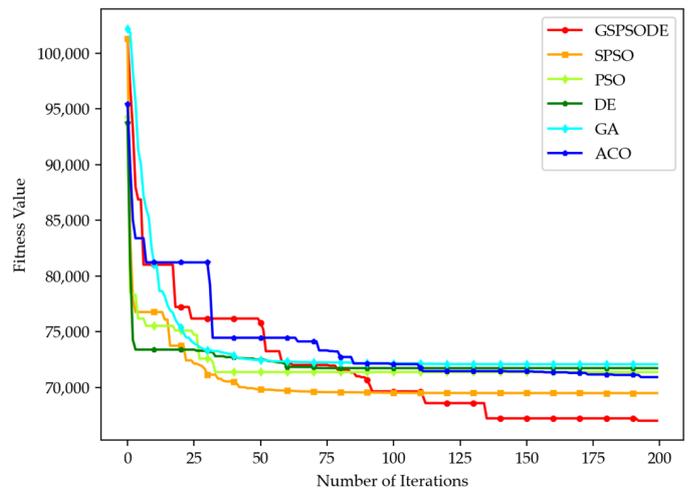
collision-free and smooth flight paths for inspection tasks in complex environments. The accuracy of the RflySim platform is also demonstrated by the excellent match between the planned paths and the flight paths. The combination of the GSPSODE algorithm and the RflySim platform provides a powerful tool for inspection tasks in various industries, such as construction, infrastructure, and search and rescue.



(a)



(b)



(c)

Figure 8. The fitness values of the six algorithms in the three scenarios. (a) The fitness values of the six algorithms in Scene 1. (b) The fitness values of the six algorithms in Scene 2. (c) The fitness values of the six algorithms in Scene 3.

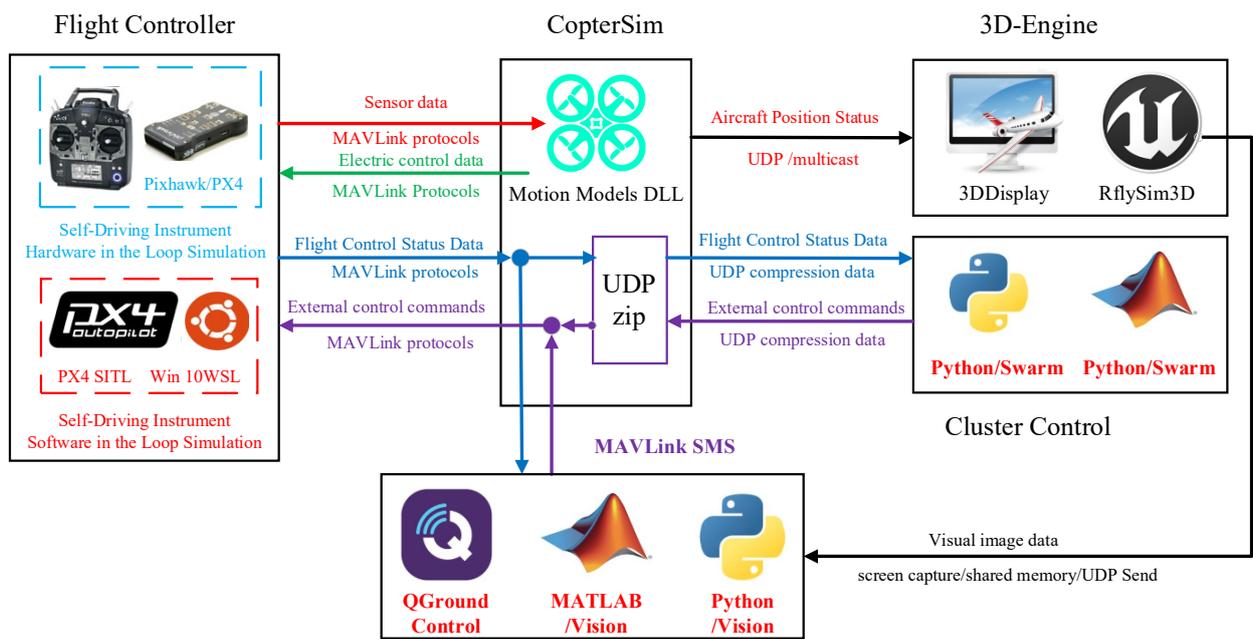


Figure 9. The composition of the software platform.

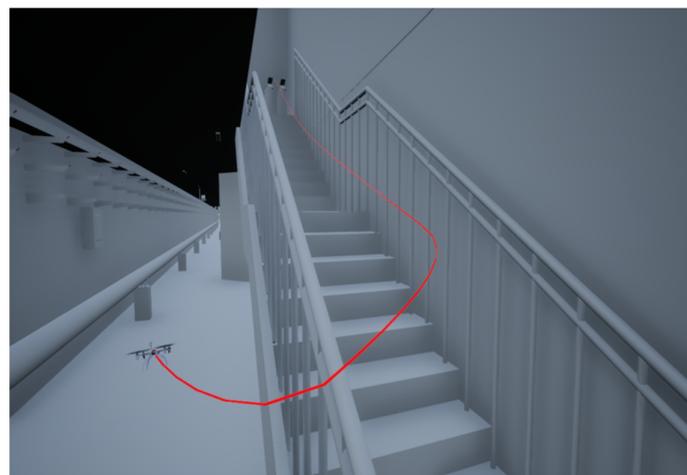


Figure 10. UAV inspection swoops downstairs.

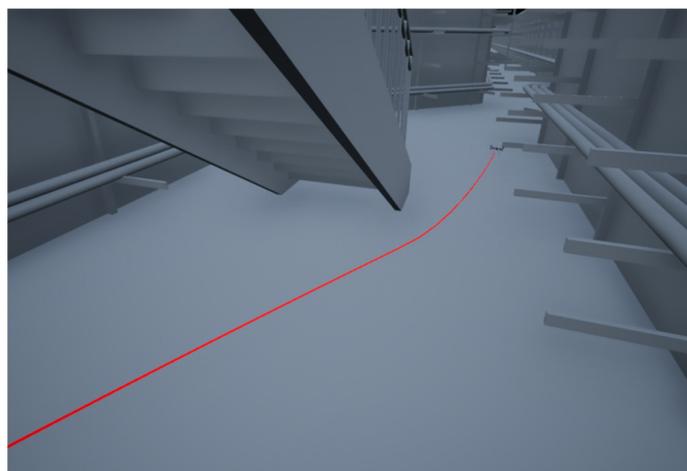


Figure 11. UAV inspection avoids stairs.

5. Conclusions

This paper proposes research on path planning for UAV inspection in urban pipeline corridors based on the GSPSODE algorithm, which mainly proposes constraints used to construct the objective function, improves the optimization algorithm used to optimize the objective function, and applies it to the path planning of UAV inspection in urban underground pipeline corridors. A hybrid optimization algorithm GSPSODE, combining the respective advantages of the SPSO algorithm and the DE algorithm, is proposed, where the two algorithms act as two players in a cooperative game based on the Nash negotiation theory in an attempt to maximize the profit gained by both parties, maintain the balance between exploration and exploitation capabilities, avoid local optimality, and increase the diversity of the proposed algorithms. In fact, the game environment creates a competitive environment between the algorithms, which greatly improves the searchability of the algorithms. The experimental results show that the proposed algorithm outperforms the classical DE algorithm and other algorithms, such as PSO, SPSO, GA, and ACO, and the creation of a game between two algorithms, DE and SPSO, in the search space can improve the efficiency of the GSPSODE algorithm. In our future work, an important issue should be addressed to improve the performance of GSPSODE. We will further focus on the search efficiency and accuracy by adjusting the parameters and further study dynamic path planning in unknown environments. Moreover, regarding large-scale urban underground pipe corridors, further adjustments can be considered to further improve the signal transmission, navigation and positioning, image recognition, infrastructure and infrared temperature measurement, and other intelligent functions to build a multi-UAVs pipe corridor inspection system, so as to build a modern smart city.

Author Contributions: Conceptualization, methodology, validation, writing, and software, C.W.; data curation, formal analysis, C.W., Y.G., L.Z. and Q.W.; supervision, funding acquisition, and review, Q.W., X.Z. and L.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the S&T Program of Hebei under grant number 21567698H.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Lu, D.; Zhang, Y.; Gong, Z.; Wu, T. A SLAM Method Based on Multi-Robot Cooperation for Pipeline Environments Underground. *Sustainability* **2022**, *14*, 12995. [[CrossRef](#)]
2. Li, R.; Ma, H. Research on UAV Swarm Cooperative Inspection and Combat Technology. In Proceedings of the 2020 3rd International Conference on Unmanned Systems (ICUS), Harbin, China, 27 November 2020; pp. 996–999.
3. Aljalud, F.; Kurdi, H.; Youcef-Toumi, K. Autonomous Multi-UAV Path Planning in Pipe Inspection Missions Based on Booby Behavior. *Mathematics* **2023**, *11*, 2092. [[CrossRef](#)]
4. da Silva, Y.M.R.; Andrade, F.A.A.; Sousa, L.; de Castro, G.G.R.; Dias, J.T.; Berger, G.; Lima, J.; Pinto, M.F. Computer Vision Based Path Following for Autonomous Unmanned Aerial Systems in Unburied Pipeline Onshore Inspection. *Drones* **2022**, *6*, 410. [[CrossRef](#)]
5. Cao, Y.; Wei, W.; Bai, Y.; Qiao, H. Multi-base multi-UAV cooperative inspection path planning with genetic algorithm. *Clust. Comput.* **2019**, *22*, 5175–5184. [[CrossRef](#)]
6. Zhen, Z.; Xing, D.; Gao, C. Cooperative search-attack mission planning for multi-UAV based on an intelligent self-organized algorithm. *Aerosp. Sci. Technol.* **2018**, *76*, 402–411. [[CrossRef](#)]
7. Wang, X.; Pan, J.-S.; Yang, Q.; Kong, L.; Snášel, V.; Chu, S.-C. Modified mayfly algorithm for UAV path planning. *Drones* **2022**, *6*, 134. [[CrossRef](#)]
8. Wang, H.; Wang, J.; Ding, G.; Chen, J.; Gao, F.; Han, Z. Completion Time Minimization with Path Planning for Fixed-Wing UAV Communications. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 3485–3499. [[CrossRef](#)]
9. Radácsi, L.; Gubán, M.; Szabó, L.; Udvaros, J. A Path Planning Model for Stock Inventory Using a Drone. *Mathematics* **2022**, *10*, 2899. [[CrossRef](#)]
10. Niewola, A.; Podsedkowski, L. L* algorithm—A linear computational complexity graph searching algorithm for path planning. *J. Intell. Robot. Syst.* **2018**, *91*, 425–444. [[CrossRef](#)]

11. Julius Fusic, S.; Ramkumar, P.; Hariharan, K. Path Planning of Robot Using Modified Dijkstra Algorithm. In Proceedings of the 2018 National Power Engineering Conference (NPEC), Madurai, India, 9 March 2018.
12. Zammit, C.; van Kampen, E.-J. Comparison Between A* and RRT Algorithms for 3D UAV Path Planning. *Unmanned Syst.* **2022**, *10*, 129–146. [[CrossRef](#)]
13. Yang, J.; Xu, X.; Yin, D.; Ma, Z.; Shen, L. A Space Mapping Based 0–1 Linear Model for Onboard Conflict Resolution of Heterogeneous Unmanned Aerial Vehicles. *IEEE Trans. Veh. Technol.* **2019**, *68*, 7455–7465. [[CrossRef](#)]
14. Shivgan, R.; Dong, Z. Energy-Efficient Drone Coverage Path Planning Using Genetic Algorithm. In Proceedings of the 2020 IEEE 21st International Conference on High Performance Switching and Routing (HPSR), Newark, NJ, USA, 11 May 2020.
15. Sharma, A. *Swarm Intelligence: Foundation, Principles, and Engineering Applications*, 1st ed.; Mathematical Engineering, Manufacturing, and Management Sciences; CRC Press: Boca Raton, FL, USA, 2022; ISBN 978-0-367-54661-8.
16. Sharma, A.; Shoal, S.; Sharma, A.; Pandey, J.K. Path Planning for Multiple Targets Interception by the Swarm of UAVs based on Swarm Intelligence Algorithms: A Review. *IETE Tech. Rev.* **2022**, *39*, 675–697. [[CrossRef](#)]
17. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [[CrossRef](#)]
18. Liu, H.; Zhang, X.; Tu, L. A modified particle swarm optimization using adaptive strategy. *Expert Syst. Appl.* **2020**, *152*, 11335. [[CrossRef](#)]
19. Shin, J.-J.; Bang, H. UAV Path Planning under Dynamic Threats Using an Improved PSO Algorithm. *Int. J. Aerosp. Eng.* **2020**, *2020*, 8820284–8820301. [[CrossRef](#)]
20. Phung, M.; Ha, Q.P. Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization. *Appl. Soft Comput.* **2021**, *107*, 107376. [[CrossRef](#)]
21. Xu, Z. Rotary unmanned aerial vehicles path planning in rough terrain based on multi-objective particle swarm optimization. *J. Syst. Eng. Electron.* **2020**, *31*, 130–141. [[CrossRef](#)]
22. Huo, L.; Zhu, J.; Li, Z.; Ma, M. A hybrid differential symbiotic organisms search algorithm for UAV path planning. *Sensors* **2021**, *21*, 3037. [[CrossRef](#)]
23. Abhishek, B.; Ranjit, S.; Shankar, T.; Eappen, G.; Sivasankar, P.; Rajesh, A. Hybrid PSO-HSA and PSO-GA algorithm for 3D path planning in autonomous UAVs. *SN Appl. Sci.* **2020**, *2*, 1805–1821. [[CrossRef](#)]
24. Chen, Q.; He, Q.; Zhang, D. UAV Path Planning Based on an Improved Chimp Optimization Algorithm. *Axioms* **2023**, *12*, 702. [[CrossRef](#)]
25. Hao, G.; Lv, Q.; Huang, Z.; Zhao, H.; Chen, W. UAV Path Planning Based on Improved Artificial Potential Field Method. *Aerospace* **2023**, *10*, 562. [[CrossRef](#)]
26. Dadvar, M.; Navidi, H.; Javadi, H.H.S.; Mirzarezaee, M. A cooperative approach for combining particle swarm optimization and differential evolution algorithms to solve single-objective optimization problems. *Appl. Intell.* **2021**, *52*, 1–20. [[CrossRef](#)]
27. Maalek, R. Field Information Modeling (FIM)TM: Best Practices Using Point Clouds. *Remote Sens.* **2021**, *13*, 967. [[CrossRef](#)]
28. Yu, C.; Cherfaoui, V.; Bonnifait, P.; Yang, D.-G. Managing Localization Uncertainty to Handle Semantic Lane Information from Geo-Referenced Maps in Evidential Occupancy Grids. *Sensors* **2020**, *20*, 352. [[CrossRef](#)] [[PubMed](#)]
29. Rodríguez-Molina, A.; Solís-Romero, J.; Villarreal-Cervantes, M.G.; Serrano-Pérez, O.; Flores-Caballero, G. Path-Planning for Mobile Robots Using a Novel Variable-Length Differential Evolution Variant. *Mathematics* **2021**, *9*, 357. [[CrossRef](#)]
30. Huang, C.; Zhou, X.; Ran, X.; Wang, J.; Chen, H.; Deng, W. Adaptive cylinder vector particle swarm optimization with differential evolution for UAV path planning. *Eng. Appl. Artif. Intell.* **2023**, *121*, 105942. [[CrossRef](#)]
31. Gubán, M.; Udvaros, J. A Path Planning Model with a Genetic Algorithm for Stock Inventory Using a Swarm of Drones. *Drones* **2022**, *6*, 364. [[CrossRef](#)]
32. Ji, H.; Hu, H.; Peng, X. Multi-Underwater Gliders Coverage Path Planning Based on Ant Colony Optimization. *Electronics* **2022**, *11*, 3021. [[CrossRef](#)]
33. Dai, X.; Ke, C.; Quan, Q.; Cai, K.Y. RFLySim: Automatic test platform for UAV autopilot systems with FPGA-based hardware-in-the-loop simulations. *Aerosp. Sci. Technol.* **2021**, *114*, 106727–106742. [[CrossRef](#)]

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.