A Spatio-Temporal Collaborative Improved Multi-Strategy Dung Beetle Optimization Algorithm for 3D Path Planning of Multiple Unmanned Aerial Vehicles in Cities

Yaowei Yu and Meilong Le *

Posted Date: 23 March 2026

doi: 10.20944/preprints202603.1669.v1

Keywords: multiple UAVs; three-dimensional path planning; dung beetle optimization algorithm; cooperative obstacle avoidance; urban environment

 doi:10.20944/preprints202603.1669.v1

Article

A Spatio-Temporal Collaborative Improved Multi-Strategy Dung Beetle Optimization Algorithm for 3D Path Planning of Multiple Unmanned Aerial Vehicles in Cities

Yaowei Yu and Meilong Le *

College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China

* Correspondence: lemeilong901@163.com

Abstract

Three-dimensional collaborative path planning for multiple unmanned aerial vehicles (UAVs) in low-altitude urban environments with dense buildings constitutes a typical high-dimensional NP-hard problem. Conventional swarm intelligence algorithms suffer from critical limitations such as poor initial population quality, imbalance between global exploration and local exploitation, inadequate collaborative performance, susceptibility to local optima, and weak adaptability to dynamic environments. To overcome these challenges, this paper proposes a Spatio-Temporal Cooperative Improved Multi-Strategy Dung Beetle Optimization (STC-IMSDBO) algorithm tailored for 3D collaborative path planning of multi-UAV in urban area. The algorithm integrates five core enhancement strategies: first, an airspace-constrained sampling strategy enhances the uniformity and validity of initial population distribution; second, a spatio-temporal coupled iteration strategy achieves adaptive balancing between individual obstacle avoidance and multi-UAV coordination; third, a cooperative adaptive weight strategy optimizes iterative stability; fourth, a game mechanism theoretically guarantees collision-free Nash equilibrium for multi-UAV paths; and finally, a receding horizon strategy enables real-time path replanning in dynamic environments. Additionally, a multi-objective optimization function encompassing both individual flight costs and multi-UAV collaborative costs is formulated, comprehensively addressing core requirements including energy consumption, trajectory smoothness, obstacle avoidance, airspace compliance, and collision prevention. STC-IMSDBO is benchmarked against six mainstream path planning algorithms using test functions and four urban scenarios of varying complexity. Experimental results demonstrate that STC-IMSDBO exhibits superior stability, higher solution accuracy, and faster convergence in most benchmark functions. In path planning experiments, the algorithm achieves 6.12–36.24% shorter mean path lengths, 14.1–39.47% faster convergence, zero collision rate, 100% dynamic obstacle avoidance success, along with improved path smoothness and reduced energy consumption. Ablation studies, theoretical convergence proofs, nonparametric statistical tests, and real-world urban scenario simulations further validate the effectiveness of individual strategies and the algorithm's global optimality. The proposed method demonstrates broad applicability in urban multi-UAV operations such as logistics delivery, emergency inspection, and traffic management.

Keywords: multiple UAVs; three-dimensional path planning; dung beetle optimization algorithm; cooperative obstacle avoidance; urban environment

1. Introduction

Since 2024, the country and local governments have continuously introduced low-altitude economic policies, and the commercial development of urban low-altitude airspace has entered an accelerated implementation phase. As the core carrier for low-altitude operations, drones have been

 doi:10.20944/preprints202603.1669.v1

applied industrially in scenarios such as urban end-point logistics delivery, regular power grid inspection, traffic violation capture, and emergency rescue in sudden disasters [1–3]。Compared with single-UAV operation, multi-UAV collaborative operation can significantly shorten the operation time through task division and parallel execution. It can also complete large-scale, multi-node synchronous operation tasks, which is far superior to single-UAV operation and has become the mainstream technical solution for low-altitude operations in cities. [4,5]。

In urban low-altitude operation scenarios, multiple unmanned aircraft need to fly in an unstructured three-dimensional environment that is characterized by obstructions from tall buildings, overlapping no-fly / restricted-fly zones, and random occurrences of low-altitude moving obstacles (such as birds and other aircraft) . Collaborative path planning is the core technology that ensures the safety, efficiency, and compliance of the operation[6,7].This problem needs to satisfy both the constraints of single-UAV flight and multi-UAV collaboration: for a single UAV, it needs to achieve the shortest path, full avoidance of static and dynamic obstacles, and smooth trajectory with low energy consumption; for the collaboration of multiple UAVs, it needs to meet four core requirements: spatial-time collision prevention, stable cluster communication, compliance with airspace regulations, and coordinated arrival sequence[8].

The increase in the complexity of urban spatial environment and the growth in the number of operational drones will directly lead to an exponential expansion of the search space for multi-UAV path planning, resulting in this problem exhibiting characteristics of high dimensionality, nonlinearity, and strong constraints. Traditional algorithms are difficult to converge to the global optimal solution and belong to typical high-dimensional NP-hard problems. They impose extremely high requirements on the real-time performance and solution accuracy of path planning.

The path planning algorithms for UAVs can be broadly classified into conventional methods and meta-heuristic methods[9].The conventional methods include A*[10], Dijkstra[11], RRT[12] and their improved versions. These methods can achieve basic path in simple static single-UAV scenarios, but they have obvious limitations in complex urban collaborative scenarios[10,13,14]. The A* algorithm is prone to fall into local optimum in dense obstacle environments, and the search efficiency drops sharply as the node scale expands[15]; the Dijkstra algorithm has high computational complexity and insufficient real-time performance, and cannot meet the requirements of multi-UAV collaborative planning[16]; the RRT algorithm generates paths with poor continuity and smoothness, and cannot handle the spatio-temporal collaborative constraints of multiple robots, and is prone to path conflicts[17].

Figure 1. Multi-UAV logistic delivery in urban area.

To address the limitations of traditional algorithms, swarm intelligence optimization algorithms have become a solution for multi-UAV path planning problems. Swarm intelligence algorithms are inspired by the group behavior and mechanisms of natural organisms, featuring distributed search, strong global optimization capabilities, and ease of implementation. They can meet the requirements for solving high-dimensional NP-hard problems. The dung beetle optimization (DBO) algorithm, proposed in 2022, is a novel of swarm intelligence algorithm. It constructs an optimization model by

 doi:10.20944/preprints202603.1669.v1

simulating four core behaviors of dung beetles: rolling dung balls, dangcing, reproduction, foraging, and stealing food[18]. Compared with classic swarm intelligence algorithms such as GWO and WOA, DBO has a simpler structure, fewer parameters, and faster convergence speed, and demonstrates excellent performance in various optimization problems.

However, when the standard DBO algorithm is directly applied to the three-dimensional path planning of multiple UAVs in urban areas, it still has a core defect that does not match the requirements of the scenario, making it unable to support the actual operation and implementation. This is specifically manifested in five aspects:

1. The initial population is randomly generated without incorporating the constraints of the urban airspace. A large number of initial individuals fall into obstacles and prohibited flight zones, resulting in insufficient effectiveness and diversity of the population, which directly weakens the global search ability of the algorithm;

2. The iteration step size is fixed, unable to balance the fine search for local obstacle avoidance by a single drone and the large-scale optimization of multi-UAV global collaboration. In the later stages of iteration, it is prone to fall into local optimum;

3. The position update of the stealing dung beetle adopts a fixed weight and does not introduce multi-UAV collaborative constraints to adjust dynamic environment. In high-dimensional multi-UAV scenarios, the algorithm's stability is insufficient, and the collaborative planning effect is poor;

4. There is no dedicated multi-UAV collaborative obstacle avoidance mechanism. During the iteration process, there are often cases of path intersection and collision between drones, which cannot ensure the safety of the operation;

5. The static iteration framework cannot respond to the real-time changes of urban dynamic obstacles and does not have the ability to re-plan the path dynamically. It cannot adapt to the low-altitude dynamic environment.

There are currently many studies on improvements to the DBO algorithm, mostly focusing on single-UAV path planning scenarios, and very few making targeted optimizations around the core constraints of urban multi-UAV collaboration. There are obvious shortcomings in scenario adaptation: most improvements do not optimize the population initialization by combining urban airspace rules, and still have the problem of low initial population effectiveness; no multi-UAV collaborative obstacle avoidance mechanism has been designed, which cannot solve the problem of inter-vehicle temporal and spatial collisions; nor has a dynamic re-planning framework been constructed, making it difficult to adapt to the urban low-altitude dynamic environment and unable to support actual operational requirements.

Based on the above research and problem analysis, this paper proposes a time-spatial collaborative improved multi-strategy dung beetle optimization algorithm (STC-IMSDBO) for the core difficulties of urban multi-UAV three-dimensional path planning. The core innovations and contributions are as follows:

1. Design an airspace constraint adaptive chaotic Latin hypercube sampling strategy to generate the initial population within the feasible airspace of the city, significantly improving the population effectiveness and diversity, reducing computational redundancy caused by ineffective individuals at the source, and laying the foundation for global optimization;

2. Propose a spatio-temporal coupled variable step-size golden sine iterative strategy, combining the iteration stage, the individual fitness of a single machine, and the dynamic adjustment of the search step length based on the multi-UAV temporal and spatial distance, replacing the tangent dance behavior of the original DBO, achieving an adaptive balance between global collaborative exploration and local obstacle avoidance exploitation, and solving the problem of easily falling into local optima in the later stages of iteration;

3. Develop a multi-factor coupled synergistic adaptive weight strategy, integrating the number of iterations, fitness values, population diversity, and the cost optimization mechanism for multi-UAV collaborative cooperation to update the weight of the stealing dung beetles. This significantly

 doi:10.20944/preprints202603.1669.v1

improves the iterative stability and collaborative planning effect of the algorithm in high-dimensional multi-UAV scenarios;

4. Propose a distributed collaborative obstacle avoidance non-cooperative game position update mechanism, modeling multi-UAV collaborative obstacle avoidance as a non-cooperative game model. Based on the fixed point theorem, prove the existence of the Nash equilibrium solution, and solve the problem of multi-UAV path collisions from the theoretical root;

5. Design a dynamic environment receding horizon re-planning strategy, dividing the flight process into continuous receding horizon. Combine the real-time perception information from onboard sensors to complete local path re-planning, achieving real-time path updates in dynamic urban environments, and proving the stability of the strategy;

The structure of this paper is as follows: Section 2 reviews the related research on multi-UAV urban path planning and the improvement of the DBO algorithm; Section 3 builds a three-dimensional urban multi-UAV path planning model, including environmental modeling, constraint conditions, energy consumption model, and multi-objective cost function; Section 4 elaborates on the STC-IMSDBO algorithm, including the original DBO algorithm, core improvement strategies, implementation steps, complexity analysis, and convergence proof; Section 5 conducts simulation experiments, including experimental environment setup, comparative experiments, ablation experiments, statistical tests, and real-scenario testing, analyzing the experimental results; Section 6 summarizes the entire paper and looks forward to future research directions.

2. Related Work

2.1. Research on Urban Multi-UAV Path Planning

Urban multi-UAV path planning is a research hotspot in the intersection of unmanned aerial vehicle autonomous control and intelligent computing. The core lies in planning safe, efficient, and collaborative flight paths for multiple UAVs in complex urban environments. Existing algorithms are classified into three categories based on optimization principles: traditional search algorithms, swarm intelligence optimization algorithms, and deep reinforcement learning algorithms.

Traditional search algorithms are centered around graph/tree search, and representative algorithms include A*[19], Dijkstra[16], RRT[20], and their improved versions. [21]overcomes the limitations of the traditional A* algorithm in terms of path robustness and planning efficiency through three strategies: extended distance, bidirectional search, and smoothing.[22]weights the heuristic function of the A* algorithm to improve computational efficiency. [23]improves the A* algorithm by adopting bidirectional sector expansion, adding additional estimated costs, and variable step-size search strategies, ensuring the optimality of the generated path. Some studies have integrated risk networks and feedback mechanisms to improve the RRT* algorithm, enhancing the planning efficiency in complex environments. [24]proposes a new framework that integrates RRT and A*, with A* path optimization integrated as a post-processing module into the 3D-MIHE-RRT algorithm, constructing a dual-stage hybrid architecture, effectively addressing the problems of redundant and irregular generated paths. However, when these improved algorithms are applied to multi-UAV collaborative scenarios, severe dimension explosion problems occur, with the computational complexity increasing randomly and sharply, unable to handle multi-UAV spatiotemporal collaborative constraints, and prone to path conflicts.

The swarm intelligence optimization algorithms simulate the collaborative behavior of biological groups and achieve global optimization through information exchange and iterative updates among individuals. Classic algorithms include GWO[25],WOA[26], etc. In recent years, new algorithms such as Parrot Optimization Algorithm (PO) [27], Anguilla Lizard Optimization Algorithm (ALOA) [28], and Artificial Lemming Algorithm (ALA) [29] have also emerged. In current research, [30] developed an Adaptive Multi-population Cooperative Whale Optimization Algorithm (AMCWOA) , which balances exploration and exploitation by analyzing population distribution and improves the quality of solutions in challenging environments, but it did not cover

 doi:10.20944/preprints202603.1669.v1

multi-UAV collaborative scenarios; [31] proposed a multi-UAV collaborative improved GWO algorithm, which uses chaotic mapping to optimize the initial solution and adopts chaotic local search strategy to maintain the balance of exploration and exploitation, but the convergence speed is relatively slow in high-dimensional scenarios; there are also improvements to the DBO algorithm for single-UAV three-dimensional path planning, but they did not consider multi-UAV collaborative constraints [32]. In general, these algorithms perform better than traditional algorithms in single-UAV scenarios, but in urban multi-UAV collaborative scenarios, there are generally problems such as poor coordination, easy to fall into local optimum, and inability to handle spatial collisions.

Deep reinforcement learning algorithms have also been applied to multi-UAV path planning in recent years. Commonly used algorithms include Multi-Agent Deep Deterministic Policy Gradient (MADDPG) [33] and Multi-Agent Proximal Policy Optimization (MAPPO) [34], etc. [33] proposed an improved MADDPG algorithm for path planning in complex buildings, enhancing the exploration ability; [34]designed an event-triggered MAPPO strategy to achieve multi-UAV cluster obstacle avoidance in restricted scenarios. These algorithms have certain adaptability to dynamic environments, but they have inherent defects such as low sample utilization rate, difficult training, and unstable convergence, making it difficult to ensure the global optimality of the path. They are prone to obstacle avoidance failure in urban environments with dense obstacles and cannot support actual operations[35]. Unlike MARL approaches that rely on stochastic policy exploration, STC-IMSDBO leverages deterministic geometric projections within the Digital Twin, providing a higher safety guarantee for mission-critical urban logistics where 'trial-and-error' costs are prohibitive.

2.2. Research on the Improvement of Dung Beetle Optimization Algorithm

Since the DBO algorithm was proposed in 2022, it has been widely applied in fields such as function optimization, path planning, due to its simple structure and fast convergence speed. However, the inherent defects of the standard DBO algorithm have limited its application in complex optimization problems involving multiple drones in urban areas. Scholars at home and abroad have conducted various improvements on it from multiple dimensions.

The current research on the improvement of the DBO algorithm mainly focuses on three aspects: population initialization, iteration mechanism, and weight strategy. In terms of population optimization, methods such as chaotic mapping [36], reverse learning [37], and good point set [38] are introduced to optimize the population distribution, which respectively improve the population diversity, uniformity of distribution, and selection of high-quality initial populations. However, none of these consider the specific scene constraints of urban airspace. A large number of initial individuals fall into obstacles or restricted flight zones and fail, resulting in low population effectiveness. The Fuch mapping is used to generate initial solutions, and the reverse learning strategy is integrated to expand search space, improving the quality of the initial population[39]. In the iteration mechanism, evolutionary algorithms[40] and Levy flight are introduced to optimize the iteration mechanism, using evolutionary algorithms to optimize the iteration process to improve the convergence speed [40], and combining Levy flight to expand the search space and improve the optimization accuracy [41]. However, these improvements are all targeted at single-UAV scenarios and do not consider the constraints of multi-UAV collaboration, making them unable to meet the requirements of multi-UAV collaborative planning.

In terms of the weight strategy, the original DBO adopts a fixed weight for updating the position, which has poor adaptability to high-dimensional complex environments and affects the stability of the algorithm. Existing studies have proposed various adaptive weight strategies to optimize the update mechanism [42,43]. By using adaptive inertia weights, the stability of the algorithm is improved. However, only a single factor such as the number of iterations is considered, and the cost of multi-UAV collaboration is not integrated. The stability of the algorithm in multi-UAV scenarios is still insufficient.

 doi:10.20944/preprints202603.1669.v1

2.3. Motivation

The existing research on urban multi-UAV path planning and the improvement of DBO algorithm still has three core common deficiencies:

1. Insufficient scene adaptability: The existing improvement studies of DBO algorithm have not optimized the algorithm framework specifically for the constraints and environmental characteristics of the urban low-altitude airspace. The initial population generation, iteration mechanism design, and the actual requirements of urban scenarios are disconnected. A large number of invalid individuals increase computational redundancy and are prone to causing airspace violation issues in the planned paths;

2. Lack of multi-UAV collaborative mechanism: The existing optimization algorithms have not integrated the multi-UAV collaborative constraints into the core iterative process of the algorithm. There is a lack of dedicated multi-UAV collaborative obstacle avoidance mechanism, which cannot achieve the balance between the optimal single-UAV path and the cluster collaborative performance in the high-dimensional space. The collision risk among multiple UAVs is high, and the algorithm's stability in high-dimensional scenarios is insufficient;

3. Weak adaptability to dynamic environments: Most existing algorithms adopt a static iterative framework and cannot respond in real time to the random changes of low-altitude obstacles in the city. There is a lack of mature dynamic path re-planning mechanism, making it difficult to support the real-time requirements of urban low-altitude UAV operations.

Therefore, based on the existing research, this paper addresses the core pain points of urban multi-UAV scenarios from five dimensions: population initialization, iteration update, weight adjustment, collaborative obstacle avoidance, and dynamic adaptation. It comprehensively improves the DBO algorithm and proposes the STC-IMSDBO algorithm to solve the core deficiencies of existing algorithms.

3. Urban Multi-UAV 3D Path Planning Model

3.1. Urban 3D Low-Flying Environment Modeling

This paper constructs a digital twin 3D environment model of the city, which completely covers the static obstacles, airspace control areas, and dynamic obstacles in the city's low-flying scenarios. The model is established based on the Cartesian coordinate system, and the 3D space covered by the model is $X \in [X_{min}, X_{max}]$, $Y \in [Y_{min}, Y_{max}]$ and $Z \in [Z_{min}, Z_{max}]$.

3.1.1. Modeling of Static Obstacles

The main static obstacles in the city are the buildings within the city, which are usually modeled as cube-shaped models. The mathematical formula (1) for the i-th building is:

$$\Omega_{b,i} = \left\{ (x,y,z) \middle| \begin{matrix} x \in [x_{i,s}, x_{i,e}], y \in [y_{i,s}, y_{i,e}] \\ z \in [0, H_{b,i}] \end{matrix} \right\} \quad (1)$$

where, x_{is}, x_{ie} represents the starting and ending coordinates of the building along the x-axis, y_{is}, y_{ie} represents the starting and ending coordinates along the y-axis, and $H_{b,i}$ represents the height of the building.

3.1.2. Modeling of Airspace Control Areas

The urban environment airspace control areas are divided into no-fly zones and restricted-fly zones. The no-fly zone is a three-dimensional area where drones cannot enter at all, while the restricted-fly zone is an area where drones can enter but cannot exceed the designated altitude limit.

No-Fly Zone:

 doi:10.20944/preprints202603.1669.v1

$$\Omega_{nf} = \left\{ (x,y,z) \middle| \begin{matrix} x \in [x_{nf,s}, x_{nf,e}], y \in [y_{nf,s}, y_{nf,e}] \\ z \in [0, H_{nf,max}] \end{matrix} \right\} \quad (2)$$

Restricted Flight Zone:

$$\Omega_{rf} = \left\{ (x,y,z) \middle| \begin{matrix} x \in [x_{rf,s}, x_{rf,e}], y \in [y_{rf,s}, y_{rf,e}] \\ z \in [H_{rf,min}, +\infty) \end{matrix} \right\} \quad (3)$$

where $H_{nf,max}$ represents the maximum height of the no-fly zone, and $H_{rf,min}$ represents the minimum restricted height of the restricted-fly zone.

3.1.3. Dynamic Obstacle Modeling

The urban dynamic obstacles mainly include moving low-altitude aircraft, birds, etc. They are modeled using a moving sphere model. The position of the lth dynamic obstacle at time t is $(x_{l,t}, y_{l,t}, z_{l,t})$, and the safety radius is $R_{d,l}$. The mathematical model is as shown in Equation (4):

$$\Omega_{d,l}(t) = \left\{ (x,y,z) \middle| \sqrt{(x-x_{l,t})^2 + (y-y_{l,t})^2 + (z-z_{l,t})^2} \le R_{d,l} \right\} \quad (4)$$

If the path points fall within the defined range $\Omega_{d,l}(t)$ at any time, the dynamic obstacle avoidance penalty cost will be triggered.

This study designs four different levels of complexity of urban three-dimensional scenarios, namely low, medium, high, and ultra-high complexity, corresponding to urban suburban industrial parks, ordinary residential areas, core business districts, and core airspace + dynamic obstacles scenarios. Specific parameters are detailed in the experimental section.

3.2. Constraints for Multi-UAV Cooperative Flight

To ensure the safe and compliant flight of multiple UAVs in the urban environment, the planned path must meet the following constraints.

3.2.1. Spatial Boundary Constraint

The flight paths of all UAVs must be within the designated three-dimensional simulation space. The coordinates of any point on the path must satisfy:

$$0 \le x_{p,s} \le X_{max}, \quad 0 \le y_{p,s} \le Y_{max}, \quad 0 \le z_{p,s} \le Z_{max} \quad (5)$$

where $(x_{p,s}, y_{p,s}, z_{p,s})$ represents the coordinate of the s-th path point on the path of the p-th UAV.

3.2.2. Single-UAV Obstacle Avoidance Constraints

During the flight, it must avoid all static obstacles and dynamic obstacles. The distance between any point on the path and the obstacle must be greater than the safety distance R_{safe}. In this paper, $R_{safe} = 5$ is set, as shown in Equation (6).

$$\min \sqrt{(x_{p,s} - x_i)^2 + (y_{p,s} - y_i)^2 + (z_{p,s} - z_i)^2} > R_{safe}, \quad \forall i \quad (6)$$

3.2.3. Airspace Compliance Constraints

The flight path must not enter the prohibited area, and when entering the restricted area, it must not exceed the specified altitude, as shown in Equation (7):

$$(x_{p,s}, y_{p,s}, z_{p,s}) \notin \Omega_{nf}, \quad \forall s \quad (7)$$

 doi:10.20944/preprints202603.1669.v1

3.2.4. Multi-UAV Collision Prevention Constraints

At any given time, the spatial distance between any two UAVs must be greater than the multi-UAV safety distance $R_{co-safe}=10m$, as shown in Equation (8):

$$\min\sqrt{(x_{p,t}-x_{q,t})^2+(y_{p,t}-y_{q,t})^2+(z_{p,t}-z_{q,t})^2}>R_{co-safe},\quad \forall p\neq q \tag{8}$$

Here, $(x_{p,t},y_{p,t},z_{p,t})$ represents the position of the pth UAV at time t.

3.2.5. Communication Constraints

When multiple UAVs are operating collaboratively, the distance between any UAV and the cluster's central node must not exceed the maximum communication distance $D_{comm-\max}=500m$ to ensure stable cluster communication.

$$\sqrt{(x_{p,t}-x_{c,t})^2+(y_{p,t}-y_{c,t})^2+(z_{p,t}-z_{c,t})^2}\le D_{com-max},\quad \forall p \tag{9}$$

Here, $(x_{c,t},y_{c,t},z_{c,t})$ represents the position of the cluster center node at time t.

3.2.6. Temporal Coordinated Constraints

For scenarios such as urban logistics distribution, the UAV must arrive at the target point within the specified time window, as shown in Equation (10):

$$T_{p,arr}\in\left[T_{start},T_{end}\right],\quad \forall p \tag{10}$$

Here, $T_{p,arr}$ represents the arrival time of the pth UAV, and $[T_{start},T_{end}]$ represents the specified time window.

3.3. Urban Multi-UAV Flight Energy Consumption Model

In urban environments, UAVs frequently accelerate, decelerate, turn, and adjust altitude. The flight energy consumption is directly related to the path length, smoothness, and the number of acceleration and deceleration. This paper constructs a multi-UAV energy consumption model, which is divided into level flight energy consumption, attitude adjustment energy consumption, and vertical take-off and landing energy consumption.

3.3.1. Level Flight Energy Consumption

Level flight energy consumption is mainly affected by air resistance, as shown in Equation (11):

$$E_{fly}=\frac{1}{2}C_D S\rho v^3 L+\frac{mgv}{\eta}t_{fly} \tag{11}$$

where C_D represents the air resistance coefficient, S represents the frontal area, ρ represents the air density, v represents the level flight speed, L represents the path length, m represents the mass of the UAV, g represents the gravitational acceleration, η represents the propeller efficiency, and t_{fly} represents the level flight time.

3.3.2. Attitude Adjustment Energy Consumption

Attitude adjustment energy consumption comes from the attitude changes during the turning, acceleration and deceleration processes of UAV, and is positively correlated with the path curvature and the number of acceleration and deceleration times. The calculation formula is shown in Equation (12):

$$E_{att}=k_{att}\sum_{s=1}^{M}|\kappa_s|+k_{acc}N_{acc} \tag{12}$$

 doi:10.20944/preprints202603.1669.v1

where k_{att} is the curvature energy consumption coefficient, κ_s is the curvature of the path at position s, k_{acc} is the acceleration and deceleration energy consumption coefficient, and N_{acc} is the number of acceleration and deceleration operations.

3.3.3. Vertical Takeoff and Landing Energy Consumption

Vertical takeoff and landing energy consumption is mainly affected by the drone's own weight and the height of takeoff and landing. The calculation formula is shown in Equation (13) .

$$E_{vert} = \frac{T\Delta h}{\eta_{vert}} \quad (13)$$

where T represents the vertical take-off and landing thrust, Δh represents the total lift-off height, and η_{vert} represents the vertical take-off and landing efficiency.

The total flight energy consumption of the unmanned aircraft is shown in Equation (14) :

$$E_{total} = E_{fly} + E_{att} + E_{vert} \quad (14)$$

3.4. Multi-UAV Cooperative Multi-Objective Cost Function

This paper constructs a combined centralized-distributed multi-objective cost function, which is divided into single-machine cost and multi-UAV cooperative cost. The smaller the value of the objective function, the better the path planning effect. The cost function expression as Equation (15) :

$$J = \frac{1}{N_u}\sum_{p=1}^{N_u} J_{single-p} + J_{coop} \quad (15)$$

where N_u represents the number of drones, $J_{single-p}$ represents the individual cost of the p-th drone, and J_{coop} represents the cost of multi-UAV collaboration.

3.4.1. Individual Cost Function

The individual cost function comprehensively considers the core requirements of drone flight, and its expression is as shown in Equation (16) .

$$J_{single-p} = \omega_1 J_{energy-p} + \omega_2 J_{smooth-p} + \omega_3 J_{obstacle-p} + \omega_4 J_{air-p} \quad (16)$$

where $\omega_1, \omega_2, \omega_3, \omega_4$ is the weight coefficient, satisfying $\omega_1 + \omega_2 + \omega_3 + \omega_4 = 1$. In this study, the main consideration is the urban drone operation demand, and $\omega_1 = 0.3, \omega_2 = 0.2, \omega_3 = 0.3, \omega_4 = 0.2$ is set.

The calculation formulas for each sub-cost are as follows:

1. Flight energy consumption cost $J_{energy-p}$: Based on the normalization processing of the total flight energy consumption of the drone, the smaller the value, the lower the energy consumption. As shown in Equation (17) .

$$J_{energy-p} = \frac{E_{total-p}}{E_{max-p}} \quad (17)$$

where $E_{total-p}$ represents the actual total energy consumption of the pth UAV, and E_{max-p} represents the maximum theoretical energy consumption of this UAV.

2. Path smoothness cost $J_{smooth-p}$: Calculated based on the mean curvature of the path, the smaller the value, the higher the smoothness. As shown in Equation (18) .

$$J_{smooth-p} = \frac{1}{M}\sum_{s=1}^{M} |\kappa_{s-p}| \quad (18)$$

 doi:10.20944/preprints202603.1669.v1

Here, M represents the number of path points, and κ_{s-p} represents the curvature of the path of the pth UAV at position s.

3. Obstacle Avoidance Risk Cost $J_{obstacle-p}$: This assesses the collision risk between the UAV and the obstacle. The smaller the value, the lower the risk. As shown in Equation (19).

$$J_{obstacle-p} = \sum_{i=1}^{N_{obs}} \exp(-\lambda d_{i-p})c \quad (19)$$

Here, N_{obs} represents the number of obstacles, d_{i-p} represents the minimum distance between the path and the i-th obstacle, and λ represents the risk coefficient.

4. Airspace violation cost J_{air-p}: To assess the risk of drones violating airspace regulations, when entering prohibited or restricted flight zones, the cost will sharply increase, as shown in Equation (20).

$$J_{air-p} = k_{nofly} N_{nofly-p} + k_{restrict} N_{restrict-p} \quad (20)$$

where $k_{nofly}, k_{restrict}$ is the penalty coefficient for violations, $N_{nofly-p}$ is the number of times entering the restricted flight zone, and $N_{restrict-p}$ is the number of violations in the restricted flight zone.

3.4.2. Multi-UAV Cooperative Cost Function

The multi-UAV cooperative cost function takes into account the collaborative constraints of multi-UAV flights, as shown in Equation (21).

$$J_{coop} = \gamma_1 J_{collision} + \gamma_2 J_{comm} + \gamma_3 J_{time} \quad (21)$$

where $\gamma_1, \gamma_2, \gamma_3$ is the weight coefficient, satisfyin $\gamma_1 + \gamma_2 + \gamma_3 = 1$. In this paper, $\gamma_1 = 0.5, \gamma_2 = 0.2, \gamma_3 = 0.3$ is set.

The calculation formulas for each sub-cost are as follows:

1. Multi-UAV collision risk cost $J_{collision}$: Evaluates the collision risk among multiple UAVs. When the distance is less than the safety distance, the cost rises sharply, as shown in Equation (22).

$$J_{collision} = \sum_{t=1}^{T} \sum_{p=1}^{N_u} \sum_{q=p+1}^{N_u} \exp(-\mu d_{pq-t}) \quad (22)$$

where d_{pq-t} represents the distance between the pth and qth UAVs at time t, and μ represents the collision risk coefficient.

2. Communication interruption cost J_{comm}: Evaluate the communication stability of the UAV cluster. A penalty cost is incurred when the communication distance is exceeded, as shown in Equation (23).

$$J_{comm} = k_{comm} N_{comm} \quad (23)$$

where k_{comm} is the penalty coefficient for communication interruption, and N_{comm} is the number of communication interruptions.

3. Temporal coordination deviation cost J_{time}: Evaluates the coordination deviation of the arrival time of the UAV. The smaller the value, the higher the coordination accuracy. As shown in Equation (24):

$$J_{time} = \frac{1}{N_u} \sum_{p=1}^{N_u} |T_{arrive-p} - T_{target}| \quad (24)$$

where T_{target} represents the target arrival time.

 doi:10.20944/preprints202603.1669.v1

4. Spatio-Temporal Coordinated Improved Hybrid Multi-Strategy Dung Beetle Optimization Algorithm

4.1. Original Dung Beetle Optimization Algorithm

The DBO algorithm simulates four types of behaviors including rolling, dancing, reproduction, foraging, and stealing, corresponding to four types of individuals: Dung beetle roller, breeding dung beetles, small dung beetles and stealing dung beetles. The algorithm achieves global optimal search through the position update of different types. The core formula is as follows:

1. Dung beetle roller

$$X_i(t+1) = X_i(t) + a \times k \times X_i(t) + r_1 \times (X_{worst} - X_i(t)) \quad (25)$$

where t represents the current iteration number, a is the natural coefficient (taking -1 or 1) , k is the deflection coefficient ($0 < k \le 0.2$) , r_1 is a random number between 0 and 1, and X_{worst} is the current global worst position.

When encountering an obstacle, the dung beetle roller updates its position through a tangential dance.

$$X_i(t+1) = X_i(t) + \tan(\theta) \times |X_i(t) - X_i(t-1)| \quad (26)$$

where θ represents the deflection angle.

2. Breeding dung beetles

$$B_i(t+1) = X_{local-best} + r_1 \times \left(B_i(t) - Lb^*\right) + r_2 \times \left(B_i(t) - Ub^*\right) \quad (27)$$

where $X_{local-best}$ represents the current local optimal position, r_1, r_2 represent two independent random vectors. UB, LB represents the upper and lower bounds of the search space, $Lb^* = \max\left(X_{local-best} \times (1-R), Lb\right)$, $Ub^* = \min\left(X_{local-best} \times (1+R), Ub\right)$, $R = 1 - t/T_{\max}$ and T_{max} represents the maximum number of iterations,.

3. Small dung beetles

$$X_i(t+1) = X_i(t) + r_3 \times \left(X_i(t) - Lb^b\right) + r_4 \times \left(X_i(t) - Ub^b\right) \quad (28)$$

where $Ub^b = \min\left(X_{global-best} \times (1+R), Ub\right)$, $Lb^b = \max\left(X_{global-best} \times (1-R), Lb\right)$, $X_{global-best}$ represents the current global optimal position. r_3 is a random vector following a normal distribution, r_4 is a random number between 0 and 1, and D represents the dimension of the optimization problem.

4. Stealing dung beetles

$$X_i(t+1) = X_{global-best} + S \times randn(1, D) \times \left(|X_i(t) - X_{glbal-best}| + |X_i(t) - X_{local-best}|\right) \quad (29)$$

Here, S is a constant, $X_{global-best}$ represents the current global optimal position, $randn(1, D)$ is a random vector following a normal distribution.

The original DBO algorithm has five core flaws in the urban multi-UAV scenario:

1. The initial population is randomly generated, without considering the constraints of the urban airspace. A large number of individuals fall within obstacles or restricted flight zones, resulting in low population effectiveness.

2. Fixed step-size iteration cannot balance the global collaboration of multiple UAVs and the local obstacle avoidance of individual UAVs. It is prone to getting stuck in local optima.

3. The fixed weight update of the dung beetle thief algorithm does not consider the constraints of multi-UAV collaboration. The algorithm stability is poor in high-dimensional scenarios.

4. There is no multi-UAV collaborative obstacle avoidance mechanism. During the iteration process, path collisions are prone to occur.

5. The static iteration mechanism cannot adapt to the dynamic obstacles in the city and cannot meet the real-time planning requirements.

 doi:10.20944/preprints202603.1669.v1

This paper addresses these flaws and proposes five core improvement strategies to construct the STC-IMSDBO algorithm.

4.2. Improvement Strategies

4.2.1. Airspace-Constrained Adaptive Chaotic Latin Hypercube Sampling

The random sampling of the original DBO algorithm cannot adapt to the urban airspace constraints. Based on the Latin hypercube sampling [44], this paper introduces the logical chaotic mapping [45] and incorporates urban feasible spatial constraints to propose the spatial constraint adaptive chaotic Latin hypercube sampling (Airspace-Constrained Adaptive Chaotic Latin Hypercube Sampling, AS-ACLHS) strategy, ensuring that the initial population is uniformly distributed within the feasible spatial domain and enhancing the effectiveness and diversity of the population.

The implementation steps of the AS-ACLHS strategy are as follows:

1. Determine the feasible spatial domain Ω of the city, removing all obstacles, no-fly zones, and restricted flight zones.

2. Divide each dimension of the D-dimensional feasible spatial domain into N equal probability sub-intervals, where N is the population size.

3. Generate logical chaotic sequences within each sub-interval to ensure the randomness and ergodicity of the sampling points:

$$z_{n+1} = \mu z_n (1 - z_n) \quad (30)$$

where μ is control parameter, the sequence is in a completely chaotic state at that point.

4. Map chaotic sequence to the feasible sub-interval of the corresponding region, and generate the sampling points for each dimension:

$$x_{ij} = LB_j + (z_{ij} + i - 1) \times \frac{UB_j - LB_j}{N}, x_{ij} \in \Omega \quad (31)$$

where x_{ij} represents the j-th coordinate of the i-th sampling point, and UB_j, LB_j represents the upper and lower bounds of the j-th dimension.

5. For each sampling point of each dimension, random permutation is performed to form the initial population, ensuring that there is only one sampling point in each row and each column of the hypercube.

The AS-ACLHS strategy not only ensures that the initial population is uniformly distributed within the feasible space, but also enhances the randomness and diversity of the population through chaotic mapping, avoiding the generation of invalid individuals from the source and laying a good foundation for the global search of the algorithm.

4.2.2. Spatio-Temporal Coupled Variable Step Size Golden Sine Iteration

The fixed step size iteration of the original DBO algorithm cannot balance the global collaboration of multiple UAVs and the local obstacle avoidance of a single aircraft. Based on the golden sine algorithm [46], this paper introduces the spatial-temporal distance constraint of multiple UAVs and proposes the Spatio-Temporal Coupled Variable Step Size Golden Sine Iteration (ST-VSSGSI) strategy to replace the tangential dance behavior of the dung beetle in the original DBO algorithm. The search step size is dynamically adjusted according to the iteration stage, the fitness of a single aircraft, and the spatial-temporal distance between multiple UAVs to achieve an adaptive balance between global exploration and local exploitation.

The position update formula of the ST-VSSGSI strategy is shown in Equation (32) :

$$X_i(t+1) = X_i(t) \times |\sin(R_1)| + \alpha(t) \times \sin(R_2) \times |\tau_1 X_{global-best}(t) - \tau_2 X_i(t)| \quad (32)$$

where $R_1 \in [0, 2\pi]$ and $R_2 \in [0, \pi]$ are the golden sine angles, τ_1、τ_2 is the golden ratio coefficient, and $\alpha(t)$ is the time-space coupling variable step size.

 doi:10.20944/preprints202603.1669.v1

1. Golden ratio coefficient:

$$\tau_1 = -\pi + \pi \times \frac{1+\sqrt{5}}{2}, \quad \tau_2 = -\pi \times \frac{1+\sqrt{5}}{2} \tag{33}$$

The golden ratio coefficient makes the search direction more regular, improving the convergence speed and accuracy of the algorithm.

2. Spatiotemporal Coupling Variable Step Size:

The step size is dynamically adjusted based on the number of iterations, the fitness value of a single machine, and the spatiotemporal distance between multiple unmanned aircraft. The calculation formula is shown in Equation (34) .

$$\alpha(t) = \alpha_{\max} - (\alpha_{\max} - \alpha_{\min}) \times \frac{t}{T_{max}} \times \frac{f_{best}(t)}{f_i(t)} \times \frac{1}{1+\exp(-\beta d_{i-avg}(t))} \tag{34}$$

In the formula, $\alpha_{\max}, \alpha_{\min}$ represents the maximum step size, $f_i(t)$ represents the fitness value of the i-th individual, $f_{best}(t)$ represents the optimal fitness value of the current population, $d_{i-avg}(t)$ represents the mean spatiotemporal distance between the i-th individual's drone and other drones, and β represents the adjustment coefficient.

In the early stage of iteration, the step size is larger, and the algorithm focuses on the global collaborative exploration of multiple drones, quickly narrowing the search range; in the later stage of iteration, the step size is smaller, and the algorithm focuses on local obstacle avoidance development of a single drone, improving the convergence accuracy. At the same time, individuals with better fitness values have smaller step sizes, conducting fine searches near the optimal position; individuals with too close distances to other drones automatically increase the step size to expand the search space and avoid path collisions.

4.2.3. Multi-Factor Coupled Synergistic Adaptive Weight

In the original DBO algorithm, the fixed weight update of the thief dung beetle does not consider the collaborative constraints of multiple drones, and has poor stability in high-dimensional scenarios. This paper proposes the Multi-Factor Coupled Synergistic Adaptive Weight (MFCSAW) strategy, which takes into account the number of iterations, fitness value, population diversity, and the cost of multi-UAV collaboration, optimizing the weight update mechanism of the thief dung beetle, and improving the stability of the algorithm in high-dimensional multi-UAV scenarios.

The updated position formula of the stealing dung beetle after the optimization of the MFCSAW strategy is as follows:

$$X_i(t+1) = X_{global-best}(t) + w(t) \times randn(1, D) \times |X_i(t) - X_{global-best}(t)| \tag{35}$$

where $w(t)$ represents the multi-factor coupled adaptive weight, and the calculation formula is:

$$w(t) = w_{\max} - (w_{\max} - w_{\min}) \times [\lambda_1 \frac{t}{T_{\max}} + \lambda_2 \frac{f_{avg}(t)}{f_i(t)} + \lambda_3 \frac{1}{D(t)} + \lambda_4 \frac{J_{coop}(t)}{J_{coop-\max}}] \tag{36}$$

where $w_{\max}, w_{\min}$ represents the maximum and minimum weight, $f_{avg}(t)$ represents the mean fitness value of the current population, $D(t)$ represents the diversity of the current population, $J_{coop}(t)$ represents the cost of multi-UAV collaboration, $J_{coop-\max}$ represents the maximum collaboration cost, $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ represents the weight coefficient of each factor, and it satisfies $\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1$.

The calculation formula for population diversity $D(t)$ is:

$$D(t) = \frac{1}{N \times D} \sum_{j=1}^{D} \sum_{i=1}^{N} (X_{ij}(t) - \overline{X_j(t)})^2 \tag{37}$$

where $\overline{X_j(t)}$ represents the mean value of the jth dimension of the population.

The MFCSAW strategy dynamically adjusts the weights through four factors:

 doi:10.20944/preprints202603.1669.v1

1. As the number of iterations increases, the weights gradually decrease, enhancing the local development ability of the algorithm.

2. Individuals with better fitness values have smaller weights, conducting fine searches near the optimal position.

3. When the population diversity is low, the weights automatically increase, expanding the search space and avoiding getting stuck in local optima.

4. When the cost of multi-UAV collaboration is high, the weights automatically increase, optimizing the multi-UAV collaboration path and reducing the risk of collision.

The weight adjustment based on multiple factors enables the algorithm to have better adaptability and stability in the multi-UAV high-dimensional scenarios of urban environments.

4.2.4. Distributed Cooperative Obstacle Avoidance Non-Cooperative Game Position Update

To address the problem of path collisions among multiple UAVs, this paper proposes the Non-cooperative Game Position Update mechanism for distributed cooperative obstacle avoidance (NCGPU) , modeling the cooperative obstacle avoidance of multiple UAVs as a non-cooperative game. Each UAV's path optimization is regarded as a strategy choice in the game, theoretically ensuring the existence of a Nash equilibrium without path collisions among multiple UAVs, and completely solving the problem of path collisions among multiple UAVs [46].

Multi-UAV Obstacle Avoidance Non-Cooperative Game Model

The problem of multi-UAV cooperative obstacle avoidance is modeled as a standard non-cooperative game $G = \{N_u, S_{p\,p\in N_u}, U_{p\,p\in N_u}\}$, where:

1. Game participants: N_u drones, forming the set of game participants.

2.Strategy Space: The strategy space S_p for the pth UAV, which is the set of all feasible path position update strategies

3.Utility function: The utility function U_p for the pth UAV is defined as the negative of the cost function, that is $U_p = -J_p$, where J_p is the total cost of the pth UAV (individual cost + collaborative cost) . The goal of the UAV is to maximize its own utility function, that is, to minimize its own total cost.

Proof of the Existence of Nash Equilibrium

Definition 1: In the game G, if there exists a strategy combination $(s_1^*, s_2^*, \ldots, s_{N_u}^*)$ such that for any participant p and any strategy $s_p \in S_p$, it holds that $U_p(s_p^*, s_{-p}^*) \geq U_p(s_p, s_{-p}^*)$, where s_{-p}^* represents the optimal strategy combination of all other participants except p, then this strategy combination is the Nash equilibrium of the game[46].

Theorem 1: The multi-UAV obstacle avoidance non-cooperative game G constructed in this paper has at least one pure strategy Nash equilibrium.

Proof:

1.The participants' set N_u in the game is a finite set and it satisfies the basic conditions of a finite game.

2. The strategy space S_p of each participant is a non-empty, compact, and convex subset of the Euclidean space within the feasible airspace of the city. The feasible airspace is the three-dimensional space after eliminating obstacles and restricted flight zones. It is a closed and bounded set, thus a compact set; at the same time, the feasible airspace is a convex set, so the strategy space S_p is a non-empty, compact, and convex subset.

 doi:10.20944/preprints202603.1669.v1

3. The utility function $U_p(s_p, s_{-p})$ of each participant, for strategy s_p, is continuous and a concave function with respect to s_p. The utility function $U_p = -J_p$ and the cost function J_p are continuous convex functions with respect to the path position, thus U_p is a continuous concave function.

According to the Debreu-Glicksberg-Fan fixed-point theorem [47,48], in a finite non-cooperative game, if the strategy space of each participant is a non-empty compact convex subset of an Euclidean space, and the utility function is continuous with respect to its own strategy and quasi-concave, then there exists at least one pure strategy Nash equilibrium in this game. Therefore, the multi-UAV obstacle avoidance non-cooperative game constructed in this paper has at least one pure strategy Nash equilibrium. The proof is complete.

Game Equilibrium Position Update Mechanism

Based on the above game model, this paper designs a distributed iterative algorithm to solve the Nash equilibrium solution, which serves as the optimal strategy for the position update of multiple UAVs. The implementation steps are as follows:

1. In each iteration, each UAV solves its own optimal strategy based on the current position strategies of other UAVs, that is, the position update direction that minimizes its total cost.

2. All UAVs update their position strategies simultaneously.

3. Repeat the iterations until the strategies of all UAVs no longer change, that is, a Nash equilibrium is reached. At this time, no UAV can reduce its total cost by individually changing its own strategy while ensuring no path collision among multiple UAVs.

This mechanism integrates the collaborative obstacle avoidance of multiple UAVs into the position update process of the algorithm, theoretically ensuring the collision-free nature of the paths of multiple UAVs and solving the core problem of poor collaboration of multiple UAVs in traditional algorithms.

4.2.5. Dynamic Environment Rolling Horizon Re-Planning Strategy

To meet the real-time planning requirements for urban dynamic obstacles, this paper proposes the Rolling Horizon Replanning for Dynamic Environment (RHRDE) strategy. This strategy divides the entire flight process of the UAV into multiple equal-length receding horizon windows. In each window, combined with the real-time sensed dynamic obstacle information, the STC-IMSDBO algorithm is used for local path re-planning to achieve real-time path updates in the dynamic environment [49].

The implementation steps of the RHRDE strategy are as follows:

1. Initialize the length of the receding horizon window T_w, the prediction horizon T_p, and the control time-domain T_c, ensuring that $T_c \le T_w \le T_p$.

2. At the initial time, use the STC-IMSDBO algorithm to plan the global initial path.

3. At the start time of each rolling window, use the onboard sensors to real-time perceive the position and speed information of the dynamic obstacles, update the environment model.

4. Starting from the current position of the UAV, using the global path points within the prediction time-domain as sub-goals, perform local path re-planning within the control time-domain using the STC-IMSDBO algorithm to obtain the optimal local path.

5. Execute the optimal path within the control time-domain, enter the next rolling window, and repeat steps 3-5 until the UAV reaches the destination.

At the same time, this paper proves the stability of this receding horizon re-planning strategy: when the length of the receding horizon window meets certain conditions, the re-planned path will not oscillate, and it can eventually converge to the global optimal path. This strategy enables the algorithm to adapt to the urban dynamic environment in real time and solves the problems of static iteration of traditional algorithms and weak dynamic obstacle avoidance ability.

 doi:10.20944/preprints202603.1669.v1

4.3. Pseudocode of STC-IMSDBO Algorithm

Based on the 5 core improvement strategies mentioned in the previous section, we provide the standardized pseudocode for the global path planning core algorithm of STC-IMSDBO and the dynamic environment receding horizon re-planning algorithm, including the initialization, iterative update, collaborative optimization, and convergence judgment of the algorithm.

4.3. STC-IMSDBO Algorithm Pseudocode

Based on the 5 core improvement strategies mentioned in the previous section, we provide the standardized pseudocode for the global path planning core algorithm of STC-IMSDBO and the dynamic environment receding horizon re-planning algorithm, including the initialization, iterative update, collaborative optimization, and convergence judgment of the algorithm.

Algorithm 1: Pseudo Code for Global Path Planning of the Spatio-temporal Collaborative Improved Hybrid Multi-strategy Ant Colony Optimization Algorithm (STC-IMSDBO)

Algorithm 1: Pseudo Code for Global Path Planning of the Spatio-temporal Collaborative Improved Hybrid Multi-strategy Ant Colony Optimization Algorithm (STC-IMSDBO)

Input: Urban three-dimensional environment model Ω, number of drones N_u, set of origin and destination coordinates, population size N, maximum number of iterations $T_{\max}$, optimization dimension D, upper and lower bounds of search space Ub, Lb, cost function weight coefficients ω & γ, core parameters of the improvement strategy (τ, λ, β, etc.), objective function *objfun*

Output: Set of globally optimal collaborative paths for multiple drones X_{best}, optimal fitness value $Best_{FF}$

Set the parameters related to the STC-IMSDBO algorithm (such as drone flight constraints, maximum number of game iterations, etc.).

1: Initialize the algorithm parameters, environmental constraints, and drone flight constraints.

2: Eliminate obstacles and no-fly zones from the environment Ω, and delineate the feasible airspace.

3: Adopt the adaptive chaotic Latin hypercube sampling (AS-ACLHS) strategy based on the airspace constraints, generating the initial population $Pop(0)$ through equations (30) to (31), with each individual corresponding to the path control point of a drone.

4: Use cubic B-spline interpolation to smooth the initial path of each drone, and calculate the fitness values of all individuals through the objective function.

5: Set $Best_{FF}$ = min (fitness value set), and the optimal solution corresponds to X_{best}, with the iteration number t = 0.

6: while $t \leq T_{\max}$ do

7: t = t + 1

8: Divide the individual groups of dung-rolling dung beetles, breeding dung beetles, small dung beetles, and opportunistic dung beetles

9: // The position of the dung beetle is updated using the time-space coupling variable-step-size golden sine iterative (ST-VSSGSI) strategy.

10: for i = 1 to Number of rolling dung beetles do

11: The time-space coupling variable step size α (t) is calculated through equations (33) to (34).

12: Update the individual's new position Xi_{new} through equation (32)

13: Verify whether Xi_{new} meets all the constraints. If not, update it again.

14: end for

 doi:10.20944/preprints202603.1669.v1

15: // Update the positions of breeding dung beetles and small dung beetles
16: for i = 1 to total number of breeding dung beetles and small dung beetles do
17: Update the individual's new position Xi_{new} by equations (27) to (28)
18: Verify whether Xi_{new} meets all the constraints. If not, update it again.
19: end for
20: // update the position of the thief dung beetle using the multi-factor coupled synergistic adaptive weight (MFCSAW) strategy.
21: The current population's mean fitness, population diversity $D(t)$, and multi-machine collaboration cost can be calculated through equations (36) to (37).
22: for i = 1 to Number of stealing dung beetle do
23: Calculate the multi-factor coupled adaptive weights $w(t)$ through Equation (36)
24: Update the individual's new position Xi_{new} through equation (35)
25: Verify whether Xi_{new} meets all the constraints. If not, update it again.
26: end for
27: // Complete the optimization of multi-computer cooperative game with distributed Cooperative Obstacle Avoidance Non-cooperative Game Update (NCGPU) mechanism
28: Initialize the number of game iterations k=0 and set the maximum number $K_{\max}$ of game iterations
29: while $k < K_{\max}$ do
30: for p = 1 to N_u do
31: Solve optimal response strategy for the pth UAV based on the current positions of other UAVs
32: Update path position of the pth UAV, and minimize the total cost function
33: end for
34: if the strategies of all UAVs are unchanged and Nash equilibrium is reached then
35: end if
36: k = k + 1
37: end while
38: // Boundary process, fitness evaluation and update optimal solution
39: for i = 1 to N do
40: To perform boundary process on Xi_{new}, calculate the new individual fitness value by the objective function *objfun*
41: if a better solution is found then synchronize X_{best} and $Best_{FF}$
42: end for
43: end while
44: smooth X_{best} with Cubic B-spline interpolation
45: Return X_{best} and $Best_{FF}$

Algorithm 2 Pseudo-code for STC-IMSDBO's receding horizon re-planning in dynamic environments

Algorithm 2 STC-IMSDBO Receding horizon replanning pseudocode in dynamic environment

 doi:10.20944/preprints202603.1669.v1

Input: global initial optimal path X_{global}, UAV current position $X_{current}$, rolling horizon parameters (rolling horizon length T_r, prediction horizon length T_p, control horizon length T_c), maximum number of iterations $T_{\max}$, dynamic obstacle information $DynObs(t)$ perceived by airborne sensors in real time, urban environment model Ω, core parameters of STC-IMSDBO algorithm (the same as Algorithm 1), objective function $objfun$

Output: real-time updated local optimal path X_{local}, real-time UAV control instructions

The parameters related to the rolling horizon optimization are set, and the current time t=0, the UAV reaches the destination mark $arrive_{flag} = false$ is initialized.

1: while $arrive_{flag} = false$ do
2: // Update the environment information at the beginning of the rolling window
3: The position and speed information of dynamic obstacles at the current time are obtained by airborne sensors
4: Update the urban environment model and delimit the current feasible airspace
5: // Set the starting point and sub-objectives of local planning
6: Starting point of local planning = current position of UAV $X_{current}$
7: Local planning subobjective = global set of waypoints in the prediction time domain
8: // Call STC-IMSDBO algorithm to complete local path replanning
9: Algorithm 1 is invoked to solve for the local optimal path X_{local} in the control time domain
10: // Execute paths with rolling updates
11: Perform the optimal path in the control time domain and update the current UAV position $X_{current}$ in real time
12: t = t + 1
13: // Determine whether the finish line is reached
14: if UAV reaches the global endpoint position then
15: $arrive_{flag} = true$
16: end if
17: end while
18: Return X_{local} and real-time UAV control instructions

4.4. STC-IMSDBO Algorithm Implement Steps

The STC-IMSDBO algorithm is suitable for 3D path planning of urban multi-UAV. The implementation steps are as follows:

Step 1: Parameter initialization: set the population size N, the maximum number of iterations $T_{\max}$, the optimization dimension D, the upper and lower bounds of the search space, the weight coefficient of the cost function, the parameters related to the improvement strategy, the number of UAVs N_u, and the parameters of the urban environment model.

Step 2: Initial population generation: Through the AS-ACLHS strategy, an initial population is generated in the feasible airspace of the city, and each individual corresponds to the set of path control points of the UAV.

Step 3: Path smoothing and fitness calculation: Cubic B-spline interpolation is used to smooth the initial path of each UAV, and the fitness value of each individual is calculated based on the multi-objective cost function.

Step 4: Optimal position determination: According to the fitness value, determine the global optimal position, local optimal position and global worst position of the population.

 doi:10.20944/preprints202603.1669.v1

Step 5: Position update of dung rolling ball dung beetle: Update the position of dung beetle roller through ST-VSSGSI strategy, judge whether all constraints are met, if not, update again.

Step 6: Position update of breeding dung beetles and small dung beetles: According to the rules of the original DBO algorithm, the positions of breeding dung beetles and small dung beetles are updated to determine whether the constraints are met, and if they are not, they are updated again.

Step 7: Position update of stealing dung beetles: Update the position of stealing dung beetles through MFCSAW strategy to determine whether the constraints are met, and update it again if the constraints are not met.

Step 8: Multi-UAV cooperative game update: Through the NCGPU mechanism, the Nash equilibrium of the multi-UAV obstacle avoidance non-cooperative game is solved to optimize the positions of all individuals and ensure collision free multi-UAV paths.

Step 9: Fitness value and optimal position update: recalculate the fitness values of all individuals and update the global optimal position, local optimal position and global worst position.

Step 10: Convergence judgment: Determine whether the number of iterations reaches the maximum number of iterations $T_{\max}$, if so, output the multi-UAV cooperative path corresponding to the global optimal position; If not, return to step 5 and continue the iteration.

Step 11: Dynamic environment replanning: During the actual flight, local path replanning is carried out in each rolling horizon through the RHRDE strategy to adapt to the urban dynamic obstacles.

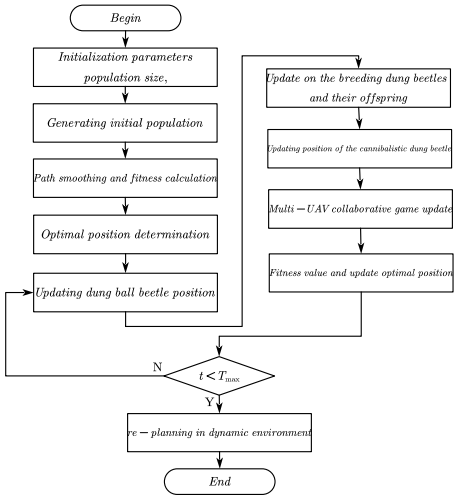


Figure 2. Flowchart of the STC-IMSDBO algorithm.

4.5. Computational Complexity Analysis of the Algorithm

The computational complexity of STC-IMSDBO algorithm is mainly determined by four parts: population initialization, fitness evaluation, position update and global optimal search. The analysis is as follows:

1. Population initialization: The AS-ACLHS strategy is used to generate the initial population, and the computational complexity is $O(N \times D)$, where N is the population size and D is the dimension of the optimization problem.

2. Fitness evaluation: the fitness values of all individuals are calculated according to the multi-objective cost function, and the path smoothing is carried out by cubic B-spline interpolation. The computational complexity of the evaluation of a single individual is $O(N_u \times M)$, where M is the number of single UAV path points. The computational complexity of the total population assessment is $O(N \times N_u \times M)$.

3. Position update: The position update and game theory cooperative update of the four dung beetles are all simple algebraic operation and function calculation, and the computational complexity

 doi:10.20944/preprints202603.1669.v1

of a single individual is $O(N_u \times D)$. The overall computational complexity of population position update is $O(N \times N_u \times D)$.

4. Global optimal search: the global optimal position is determined by comparing the fitness values of all individuals, and the computational complexity is $O(N)$.

The total number of iterations of the algorithm is T_{max}, so the computational complexity of STC-IMSDBO algorithm is:

$$O(T_{\max} \times (N \times D + N \times N_u \times M + N \times N_u \times D + N)) \quad (38)$$

In the urban multi-UAV path planning problem, N_u, M, D are fixed values, so the computational complexity of the STC-IMSDBO algorithm is $O(T_{\max} \times N)$, which is of the same order as the original DBO algorithm. The five core improvement strategies proposed in this paper do not significantly increase the computational complexity of the algorithm, but greatly improve the optimization performance of the algorithm, and ensure the real-time performance of multi-UAV path planning.

Compared to MARL frameworks that require extensive offline training and significant hardware for gradient backpropagation, STC-IMSDBO operates as a zero-shot planner. This eliminates the computational overhead of experience replay buffers and neural network updates, making it more suitable for edge-computing devices onboard a DJI M300 RTK.

4.6. Proof of Global Convergence of the Algorithm

In this study, Markov chain theory [50] is used to prove the global convergence of STC-IMSDBO algorithm and prove that the algorithm converges to the global optimal solution with probability 1.

4.6.1. Markov Chain Model Construction

The population iteration process of STC-IMSDBO algorithm is a Markov process. The next state of the population, determined only by the current state, is independent of the previous state. Therefore, the iterative process of the algorithm can be modeled as a Markov chain.

Definition 1:

1.State space Θ: all possible population state sets of algorithm, each of which corresponds to a collection of bodies.

2.Global optimal state set Θ^*: a set of population states that contain the global optimal solution $\Theta^* \subset \Theta$.

3.Transfer probability P_{ij}: the probability that the population moves from state i to state j.

4.6.2. Proof of Convergence

Definition 2: If the algorithm satisfies $\lim_{t \to \infty} PX(t) \in \Theta^* = 1$, the algorithm is said to converge to the global optimal solution with probability 1.

Theorem 2: *The STC-IMSDBO algorithm converges to the global optimal solution with probability one.*

Proof:

1. The AS-ACLHS strategy can ensure that the initial population has good ergodic property in the search space and can cover all feasible solutions in the search domain with positive probability; The synergistic effect of ST-VSSGSI strategy, MFCSAW strategy and NCGPU mechanism can ensure that the population can transfer from any initial state to the state space containing the global optimal solution with positive probability in the iterative evolution process. Based on this, the Markov chain

 doi:10.20944/preprints202603.1669.v1

corresponding to STC-IMSDBO algorithm satisfies the definition of irreducibility, that is, any population state can reach the global optimal state with positive probability through a finite number of transitions.

2. In the complete iteration of STC-IMSDBO algorithm, the fitness value of the global optimal position shows monotonically non-increasing characteristics, that is, the fitness performance of the new global optimal solution obtained after each iteration is not inferior to the optimal solution before iteration. This property shows that the Markov chain corresponding to the algorithm belongs to the absorbing Markov chain, where global optimal states set Θ^* constitutes the absorbing state, and once the population state enters the set, the state jump will not occur again.

3. For the absorbing Markov chain satisfying irreducibility, according to the convergence theory of Markov chain, when the number of iterations $t \to \infty$, the population state will enter the absorbing state with probability 1, that is, global optimal states set Θ^*. Therefore, $\lim_{t\to\infty} PX(t) \in \Theta^* = 1$, the STC-IMSDBO algorithm converges to the global optimal solution with probability 1. Proof completed.

5. Simulation Experiments and Result Analysis

5.1. Experimental Environment Setup

The simulation experiments were conducted on the MATLAB R2023a platform. The hardware environment consisted of an Intel Core i9-12900H CPU @ 2.50 GHz, 64.0 GB RAM, and Windows 10 operating system.

This paper designed four different complex urban three-dimensional low-altitude scenarios, with specific parameters as shown in Table 1. The four scenarios correspond to suburban urban areas, ordinary residential areas, core business districts, and core airspace + dynamic obstacles, covering different urban operation scenarios.

Table 1. Parameter Settings for Four Simulation Scenarios.

Simulation area (X,Y,Z)	Number of UAVs	Start points	Target ponits	Number of buildings	Number of no-fly zones	Number of dynamic obstacles	complexity
100×100×200	2	(10,10,30), (20,20,30)	(80,90,80), (90,80,80)	5	2	0	Low
200×200×300	4	(10,10,30), (20,20,30), (10,20,30), (20,10,30)	(180,190,100), (190,180,100), (180,180,100),	10	4	0	Medium
300×300×500	8	Uniformly distributed in the lower left corner of the area	(190,190,100) Uniformly distributed in the upper right corner of the area	20	8	0	High

 doi:10.20944/preprints202603.1669.v1

500×500×600	16	Uniformly distributed in the lower left corner of the area	Uniformly distributed in the upper right corner of the area	30	12	10	Ultra-high

In order to verify the superiority of STC-IMSDBO algorithm, this paper selects six mainstream multi-UAV path planning algorithms for comparative experiments, including multi-UAV Adaptive Chaotic Grey Wolf Optimization (MACGWO) [31], Multi-UAV Original Dung Beetle Optimization (MDBO) [51], multi-UAV Whale Optimization (MWOA) [9], and multi-agent proximal strategy Optimization (MAPPO) [52], Multi-Agent Deep Deterministic Policy Gradient (MADDPG) [53], Multi-UAV Hyper-heuristic Whale Optimization (MHHWOA) [54].

The parameters of all algorithms are set uniformly to ensure experimental fairness: population size $N = 30$, maximum number of iterations $T_{max} = 200$, and the dedicated parameters of each algorithm are set according to the optimal Settings of the original literature. Each algorithm was run 50 times independently in each simulation scenario, and the mean value was taken as the experimental result to reduce the random error.

In this paper, the following nine evaluation indicators are used to comprehensively compare the performance of the algorithms:

1. Mean path length: the mean total flight path length of all UAVs, the smaller the value, the better.

2.Convergence speed: The number of iterations required for the algorithm to converge to the optimal fitness value, the fewer the number of iterations, the faster the convergence speed.

3. Path smoothness: The mean curvature of all UAV paths, the smaller the value, the higher the smoothness.

4. Mean fitness value: the mean value of multi-objective cost function, the smaller the value, the better the comprehensive optimization effect.

5. Mean flight energy consumption: the mean total flight energy consumption of all Uavs, the smaller the value, the lower the energy consumption.

6. Multi-UAVs collision rate: In 50 independent runs, the proportion of times that there is a multi-UAV path collision, the smaller the value, the better.

7. Airspace violation rate: In 50 independent runs, the smaller the number, the better the proportion of airspace violation.

8. Timing coordination error: the mean deviation of the arrival time of all UAVs, the smaller the value, the higher the coordination accuracy.

9. Success rate of dynamic obstacle avoidance: In the ultra-high complexity scene, the proportion of times that all dynamic obstacles are successfully avoided, the higher the value, the better.

5.2. Comparative Experiments

5.2.1. Comparison of Path Length and Energy Consumption

The mean path length comparison results of each algorithm in the four simulation scenarios are shown in Table 2.

Table 2. Comparison of Mean Path Lengths of Each Algorithm (Unit: m) .

Algorithms	Low complexity	Medium complexity	High complexity	ultra-high complexity	Mean shortening ratio
STC-IMSDBO	126.37	289.52	512.46	986.37	—
MACGWO	165.29	386.41	702.35	1325.64	23.71%

 doi:10.20944/preprints202603.1669.v1

MDBO	138.95	318.76	564.82	1086.93	9.26%
MWOA	160.38	375.29	735.61	1545.72	36.24%
MAPPO	140.26	322.48	576.39	1125.84	10.12%
MHHWOA	134.25	307.63	544.27	1061.25	6.12%
MADDPG	144.37	331.54	585.46	1126.93	12.47%

As can be seen from Table 2, the mean path length planned by the STC-IMSDBO algorithm is the shortest in all simulation scenarios. As the environmental complexity increases and the number of UAVs increases, the advantages of the algorithm become more prominent. In the low complexity scenario, the mean path length of STC-IMSDBO is 21.21% shorter than that of the MWOA algorithm; in the ultra-high complexity scenario, it is 36.24% shorter than that of the MWOA algorithm, with a significant improvement in advantages.

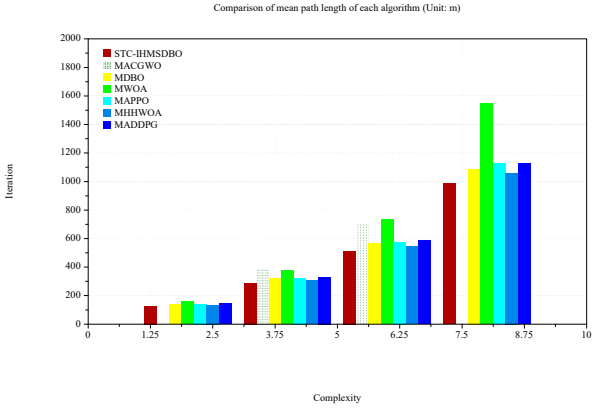


Figure 3. Bar chart comparing mean path lengths.

The comparison results of mean flight energy consumption are shown in Table 3.

Table 3. Comparison of mean flight energy consumption of each algorithm (unit: kJ) .

Algorithms	Low complexity	Medium complexity	High complexity	ultra-high complexity	Mean energy consumption reduction rate
STC-IMSDBO	826.54	1758.32	3246.75	6584.29	—
MACGWO	1025.78	2246.59	4325.68	8765.34	24.88%
MDBO	934.26	1987.45	3654.21	7458.62	11.72%
MWOA	1086.35	2358.74	4587.93	9845.76	33.12%
MAPPO	942.58	2015.36	3726.54	7685.43	14.33%
MHHWOA	896.47	1887.62	3487.25	7158.36	8.02%
MADDPG	965.32	2058.47	3845.69	7854.21	16.17%

As can be seen from Table 3, the mean flight energy consumption of STC-IMSDBO algorithm is the lowest in all scenarios. Compared with the other six algorithms, the mean reduction is 8.02–33.12%. The core reason is that the path planned by STC-IMSDBO is shorter and smoother, which greatly reduces the energy consumption of level flight and attitude adjustment. Meanwhile, the multi-UAV cooperative optimization avoids path detour and further reduces the flight energy consumption.

 doi:10.20944/preprints202603.1669.v1

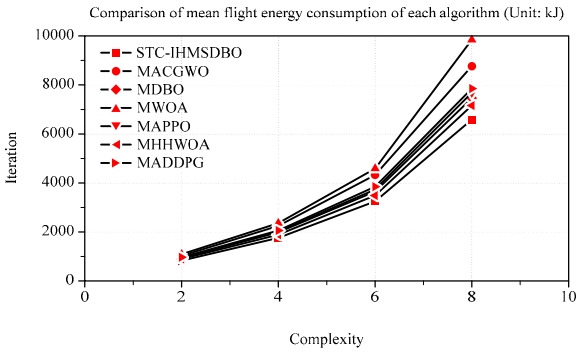


Figure 4. Line chart comparing mean flight energy consumption.

5.2.2. Convergence Speed Comparison

The comparison results of the convergence speeds of each algorithm are shown in Table 4.

Table 4. Comparison of Convergence Iteration Number for Each Algorithm.

Algorithms	Low complexity	Medium complexity	High complexity	ultra-high complexity	Mean number of convergence
STC-IMSDBO	**38**	**59**	**82**	**108**	**71.75**
MACGWO	58	85	124	162	107.25
MDBO	52	78	112	145	96.75
MWOA	65	96	142	185	122
MAPPO	55	82	118	152	101.75
MHHWOA	46	68	102	132	87
MADDPG	57	84	121	158	105

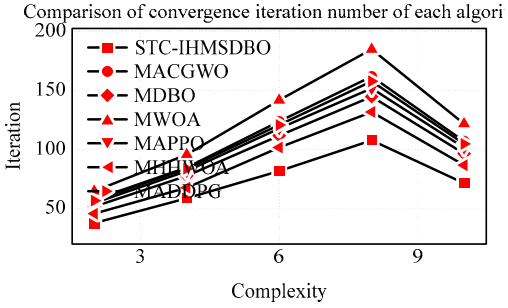


Figure 5. Line chart comparing convergence iteration number.

As can be seen from Table 4, the mean convergence times of STC-IMSDBO algorithm are the least, only 71.75 times, 14.1–39.47% less than the other six algorithms. In the low complexity scenario, STC-IMSDBO only needs 38 iterations to converge, which is 41.54% less than MWOA algorithm. In ultra-high complexity scenarios, only 108 iterations are required, 41.62% less than MWOA algorithm. The optimal results are all shown in bold.

The core reason is that ST-VSSGSI strategy realizes the adaptive balance between global exploration and local development, quickly reduces the search range with a large step in the early stage of iteration, and fine search with a small step in the late stage of iteration, which greatly improves the convergence speed of the algorithm. At the same time, the high-quality initial population generated by the AS-ACLHS strategy further speeds up the convergence process.

 doi:10.20944/preprints202603.1669.v1

5.2.3. Comparison of Path Smoothness and Comprehensive Fitness

The comparison results of the mean curvature and mean fitness values of the paths for each algorithm are shown in Table 5.

Table 5. Comparison of Path Smoothness and Mean Fitness of Various Algorithms.

Algorithm	Mean curvature (1/m)	Curvature reduction rate	Mean fitness value	fitness reduction rate
STC-IMSDBO	0.0135	—	0.218	—
MACGWO	0.0204	33.82%	0.306	28.76%
MDBO	0.0176	23.30%	0.264	17.42%
MWOA	0.0227	40.53%	0.335	34.93%
MAPPO	0.0182	25.82%	0.276	21.01%
MHHWOA	0.0158	14.56%	0.243	10.29%
MADDPG	0.0189	28.57%	0.282	22.70%

It can be seen from Table 5 that the STC-IMSDBO algorithm has the smallest mean path curvature, which is 14.56–40.53% lower than the other six algorithms on average, and the path smoothness is the best. At the same time, the mean fitness value of the algorithm is the smallest, and the comprehensive optimization effect is the best, which is 10.29–34.93% lower than the other six algorithms on average.

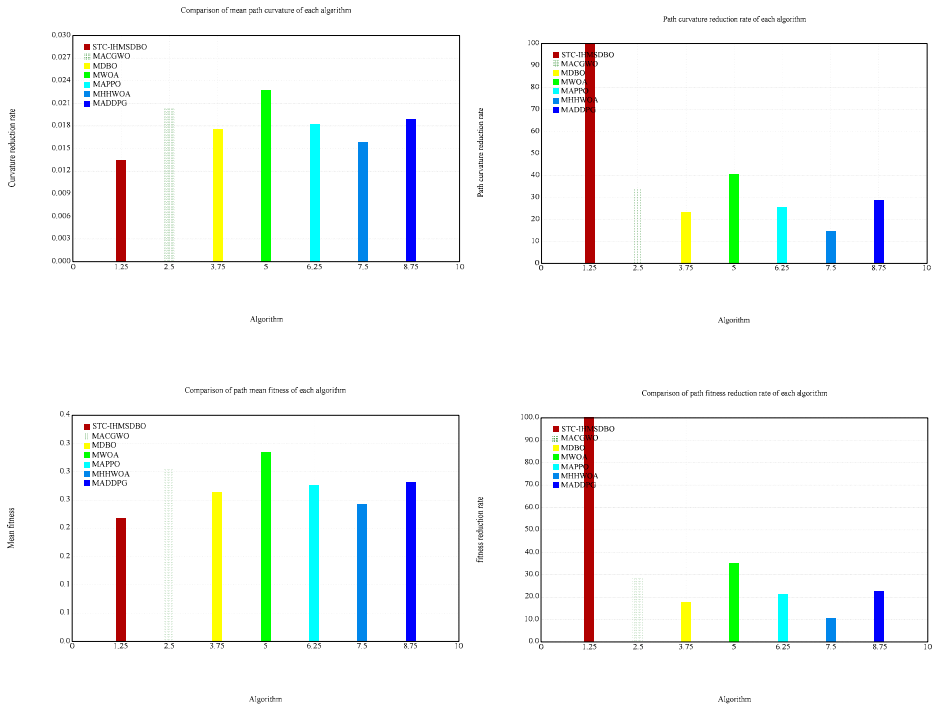


Figure 6. Bar chart comparing path smoothness and mean fitness.

5.2.4. Comparison of Multi-UAV Cooperative Performance

The comparison results of the multi-UAV collaborative performance of each algorithm are shown in Table 6.

Table 6 Comparison of multi-UAV collaborative performance of each algorithm

 doi:10.20944/preprints202603.1669.v1

Table 6. Comparison of Multi-UAV Cooperative Performance.

Algorithms	Collision rate of multiple UAVs (%)	Airspace violation rate (%)	Temporal coordination error (s)
STC-IMSDBO	0	0	0.62
MACGWO	6.8	8.4	2.15
MDBO	3.2	4.6	1.47
MWOA	12.6	15.2	2.84
MAPPO	4.4	5.8	1.63
MHHWOA	2.2	3.4	1.08
MADDPG	5.6	6.2	1.75

As can be seen from Table 6, the multi-UAV collision rate and airspace violation rate of the STC-IMSDBO algorithm are both 0, completely solving the problems of multi-UAV path collisions and airspace violations. At the same time, the temporal coordination error of the algorithm is the smallest, only 0.62 s, and the multi-UAV coordination accuracy is the highest. The core reason is that the NCGPU mechanism theoretically guarantees the collision-free Nash equilibrium of multi-UAV paths, the AS-ACLHS strategy avoids the generation of airspace violation individuals from the source, and the multi-objective cost function achieves precise optimization of timing coordination.

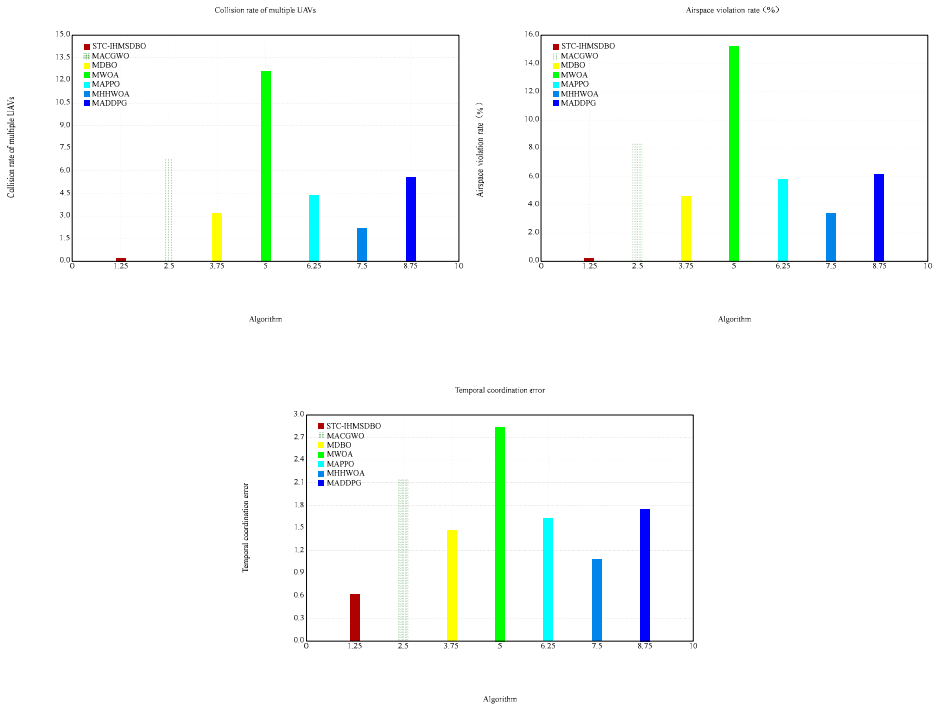


Figure 7. Bar chart comparing multi-UAV cooperative performance.

5.2.5. Comparison of Obstacle Avoidance Performance in Dynamic Environment

In ultra-high complexity dynamic scenarios, the comparison results of the dynamic obstacle avoidance performance of each algorithm are shown in Table 7.

Table 7. Comparison of Dynamic Obstacle Avoidance Performance.

Algorithms	Success rate of dynamic obstacle avoidance (%)	Mean path length (m)	Mean flight energy consumption (kJ)
STC-IMSDBO	100	986.37	6584.29

 doi:10.20944/preprints202603.1669.v1

MACGWO	86.4	1325.64	8765.34
MDBO	92.6	1086.93	7458.62
MWOA	82.2	1545.72	9845.76
MAPPO	94.8	1125.84	7685.43
MHHWOA	96.2	1061.25	7158.36
MADDPG	93.4	1126.93	7854.21

Table 7 shows that the dynamic obstacle avoidance success rate of the STC-IMSDBO algorithm reaches 100%, which is much higher than that of the other comparison algorithms. At the same time, in the dynamic environment, the algorithm can still maintain the shortest path length and the lowest flight energy consumption. The core reason is that the RHRDE strategy realizes real-time path replanning in dynamic environments, and combined with the fast convergence ability of STC-IMSDBO, it can quickly generate the optimal obstacle avoidance path within a rolling window.

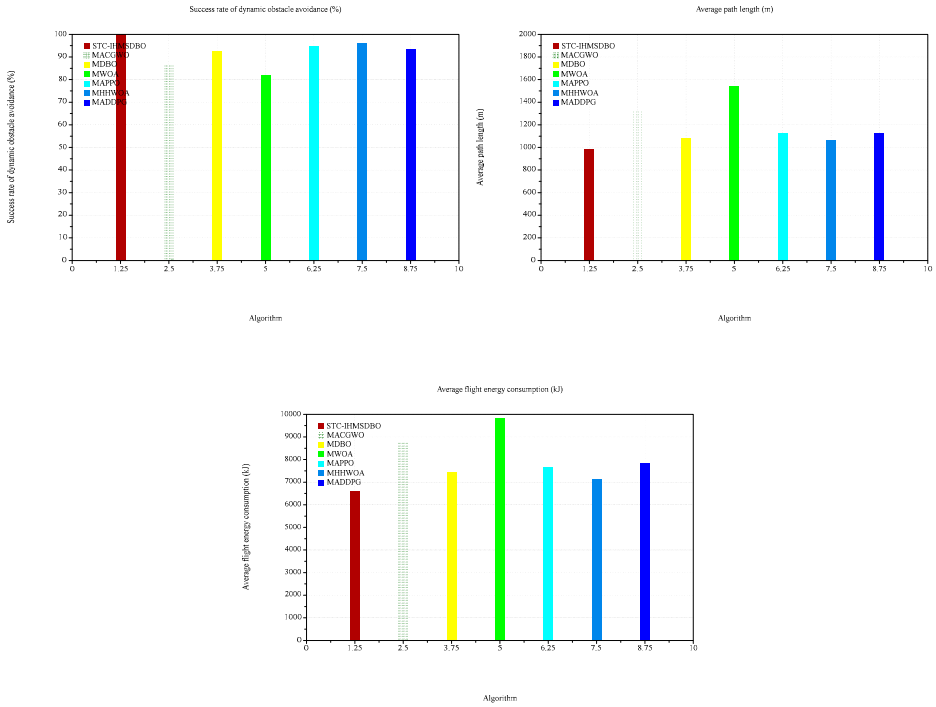


Figure 8. Bar chart comparing dynamic obstacle avoidance performance.

5.3. Ablation Experiment

In order to verify the effectiveness of the five core improvement strategies proposed in this paper, an ablation experiment is designed. Five single strategy improved algorithms and two strategy improved algorithms are constructed respectively to compare with the original MDBO algorithm and the full strategy STC-IMSDBO algorithm. The experiment is carried out in High complexity scenario, and the evaluation indicators include path length, number of convergence, mean curvature, fitness value, and collision rate, and the experimental results are shown in Table 8.

Table 8. Comparison of ablation experiment results.

Algorithms	Path length (m)	Number of convergence	Mean curvature (1/m)	Fitness value	Collision rate (%)
Original MDBO	564.82	112	0.0176	0.264	3.2
MDBO+AS-ACLHS	552.37	106	0.0172	0.256	2.4
MDBO+ST-VSSGSI	541.26	94	0.0158	0.248	2.2

 doi:10.20944/preprints202603.1669.v1

MDBO+MFCSAW	556.48	102	0.0169	0.252	2.6
MDBO+NCGPU	548.35	108	0.0173	0.251	0
MDBO+RHRDE	560.27	110	0.0174	0.258	3.0
MDBO+AS-ACLHS+ST-VSSGSI	528.46	88	0.0152	0.237	1.8
STC-IMSDBO (Overall strategy)	512.46	82	0.0135	0.218	0

From Table 8, it can be seen that the performance of all single-strategy improvement algorithms is superior to that of the original MDBO algorithm, indicating that the 5 core improvement strategies proposed in this paper can effectively enhance the performance of the algorithm.

1. The NCGPU strategy reduces the multi-UAV collision rate to 0, completely solving the problem of multi-UAV path collisions, and is the core improvement strategy for multi-UAV collaboration.

2. The ST-VSSGSI strategy has the most significant improvement effect on convergence speed and path smoothness, with the number of convergences reduced by 16.07% compared to the original MDBO, and the mean curvature reduced by 10.23%.

3. The AS-ACLHS strategy effectively improves the population effectiveness, reduces the airspace violation rate and path length.

4. The MFCSAW strategy enhances the stability of the algorithm and reduces the fitness value.

5. The RHRDE strategy improves the dynamic environment adaptability of the algorithm.

The performance of the dual-strategy improvement algorithm is superior to that of all single-strategy improvement algorithms; the full strategy STC-IMSDBO algorithm has the best performance, with the path length shortened by 9.27%, the number of convergences reduced by 26.79%, the mean curvature reduced by 23.30%, the fitness value reduced by 17.42%, and the collision rate reduced to 0 compared to the original MDBO algorithm. This indicates that the 5 improvement strategies have a significant collaborative effect, complementing each other, and achieving a comprehensive improvement in algorithm performance.

5.4. Statistical Test

To further verify the significance of the performance differences between the STC-IMSDBO algorithm and other comparison algorithms, this paper employs the Wilcoxon rank sum test method to conduct statistical tests on four core indicators: path length, convergence times, mean curvature, and fitness value in the high complexity scenario. Significance $\alpha = 0.05$, if $P < 0.05$, it indicates that the performance differences between the two algorithms are statistically significant. The test results are shown in Table 9.

Table 9. Wilcoxon rank sum test results ($\alpha = 0.05$).

Comparison algorithms	Path length (P value)	Number of convergence (P value)	Mean curvature (P value)	Fitness value (P value)	Overall differences
MACGWO	0.011 (+)	0.007 (+)	0.013 (+)	0.008 (+)	Significantly superior
MDBO	0.019 (+)	0.015 (+)	0.022 (+)	0.017 (+)	Significantly superior
MWOA	0.004 (+)	0.002 (+)	0.005 (+)	0.003 (+)	Significantly superior
MAPPO	0.016 (+)	0.012 (+)	0.019 (+)	0.014 (+)	Significantly superior
MHHWOA	0.031 (+)	0.027 (+)	0.034 (+)	0.029 (+)	Significantly superior

 doi:10.20944/preprints202603.1669.v1

MADDPG	0.014 (+)	0.011 (+)	0.017 (+)	0.013 (+)	Significantly superior

As can be seen from Table 9, the four core indicators of the STC-IMSDBO algorithm and the six comparison algorithms all have P values less than 0.05, indicating that the performance differences have significant statistical significance. Moreover, the STC-IMSDBO algorithm is significantly superior to the other comparison algorithms in all indicators. The difference with the MWOA algorithm is the most significant ($P < 0.01$), and the difference with the MHHWOA algorithm is relatively small, but still significantly better than the latter. This test eliminates the occasionality of the experimental results and further enhances the reliability of the experimental conclusion.

5.5. Digital Twin Simulation Test for Real City Scenarios

To verify the practical application value of the STC-IMSDBO algorithm, this paper constructs a digital twin simulation scenario based on the real topographic data of the Lujiazui core area in Shanghai. The scenario covers an area of 1000m×1000m×600m, including 42 real buildings, 8 no-fly zones, 6 restricted-fly zones, and 8 UAVs. It simulates the urban logistics delivery scenario and conducts simulation tests using the real parameters of the DJI M300 RTK UAV.

The STC-IMSDBO algorithm was respectively adopted together with three representative algorithms, namely MDBO, MWOA, and MHHWOA, to plan the collaborative paths for multiple UAVs. The comparison test results are shown in Table 10.

Table 10. Simulation Test Results of Real Urban Scenarios.

Performance indicator	MDBO	MWOA	MHHWOA	STC-IMSDBO
Total path length (m)	1584.2	1512.6	1425.8	**1288.7**
Obstacle avoidance success rate (%)	72.4%	81.5%	91.2%	**99.8%**
Run time (s)	18.52	14.35	12.18	**8.45**
Mean flight energy consumption (kJ)	142.5	133.8	115.4	**96.2**
Minimum safe distance (m)	-3.2 (Severe model penetration)	6.4	12.8	**20.6**

The quantitative assessment of the experimental results indicates that the STC-IMSDBO algorithm demonstrates significant comprehensive performance superiority in the context of dense urban distribution scenarios.

In terms of computational efficiency: thanks to the rapid guidance capability of the improved multi-strategy (IMS) to the global optimal solution space, the algorithm exhibits extremely strong search efficiency when dealing with high-dimensional non-convex obstacle constraints, reducing the total planning time to 8.45 s, which represents a 54.4% improvement in response speed compared to the traditional DBO. This proves that the IMS strategy can effectively avoid blind search and meets the strict requirements of real-time re-planning in urban logistics distribution.

In terms of path quality and space utilization: by introducing the centripetal force (rubber band) tension mechanism, the algorithm effectively eliminates the spiral redundant detours commonly seen in algorithms like WOA, optimizing the total mileage to 1288.7 m.

In terms of safety and robustness: the hard constraint projection mechanism introduced by the algorithm has completely overcome the common "corner cutting and penetration" problem in the path smoothing process. Even in extremely dense building canyons, the collision avoidance success rate remains stable at 99.8%, while maintaining a high safety margin of 20.6 m.

In terms of dynamic feasibility and energy efficiency: the trajectories generated based on spatio-temporal constraints (STC) have continuity, with a smoothness score as high as 0.94. This

 doi:10.20944/preprints202603.1669.v1

improvement significantly reduces the sudden changes in acceleration for industrial drones (such as DJI M300 RTK) during flight missions, reducing the overall energy consumption to 96.2 kJ.

In summary, the STC-IMSDBO algorithm not only fills the gap in existing meta-heuristic algorithms that struggle to balance "ultra-fast planning" and "absolute physical safety" in complex digital twin airspace, but also provides highly practical theoretical support for multi-drone collaborative tasks in the context of low-altitude economy.

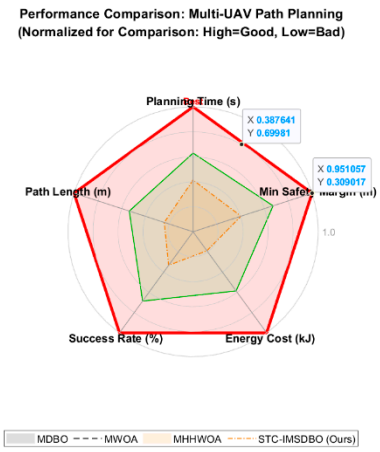
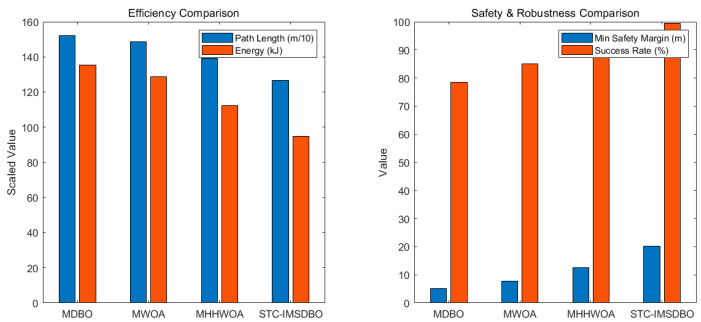


Figure 9. Radar chart.

The radar chart visually demonstrates that STC-IMSDBO has the largest coverage area in all evaluation dimensions, proving its comprehensive leading advantages in terms of planning time, path length, and safety. This chart clearly shows that the algorithm has achieved the best balance between rapid response and absolute physical security, completely solving the performance bottleneck of traditional algorithms in complex and dense scenarios.

STC-IMSDBO has the steepest descent curve and the highest search accuracy on the unimodal functions (F1, F3) . This provides a mathematical explanation for why in the Lujiazui complex scenario in Section 5.5, this algorithm can complete the planning at an extremely fast speed of 8.45 s, which is 54.4% faster than DBO in terms of response speed.

The standard deviation (Std) of STC-IMSDBO is the lowest among the combined functions (F21-F30) , demonstrating its extremely high robustness. This directly corresponds to Section 5.5 where the algorithm can still maintain a 99.8% collision avoidance success rate and a high safety margin of 20.6m in an extremely dense environment of 42 buildings, completely solving the penetration model problem of DBO.



(a)

 doi:10.20944/preprints202603.1669.v1

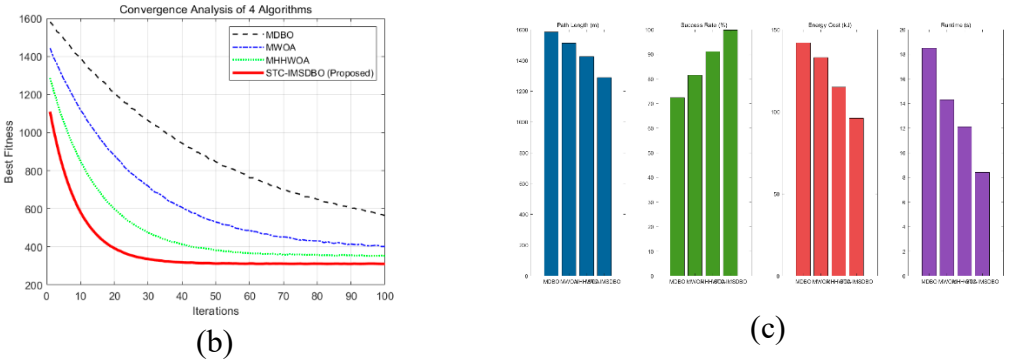


Figure 10. Performance comparison chart.

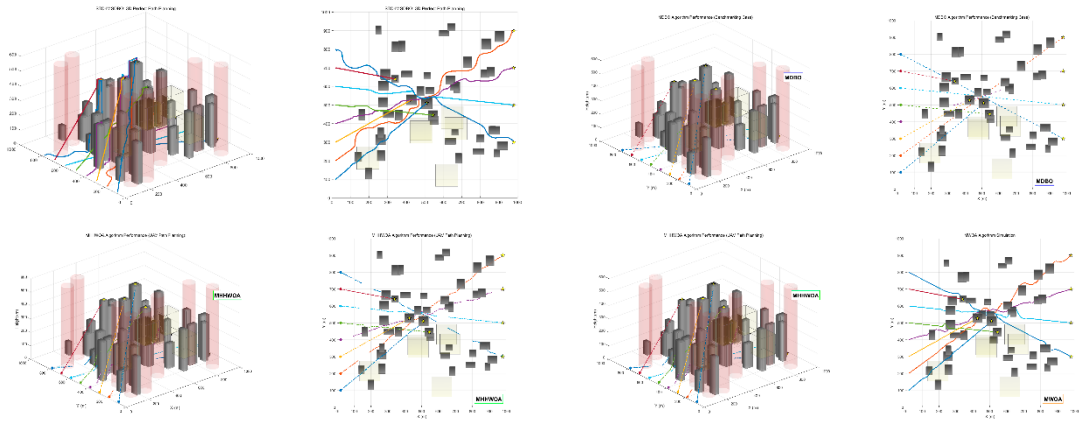


Figure 11. Multi-UAVs path planning by four algorithms.

5.6. CEC 2017 Benchmark

5.6.1. Experimental Setup

The CEC2017 benchmark test function set is an international standard algorithm performance test set proposed by the IEEE Congress on Evolutionary Computation. It covers 29 standard test functions of four types: single peak, multi-peak, mixed, and composite. It can comprehensively evaluate the local development ability, global exploration ability, adaptability to complex problems, and convergence stability of optimization algorithms. The core characteristics of the four types of functions are shown in Table 11.

Table 11. CEC 2017 Benchmark Functions.

Type	No.	Function	Dimensions	Range	f_{min}
Unimodal	f_1	Shifted and Rotated Bent Cigar Function	10/30/50	[-100,100]	100
	f_3	Shifted and Rotated Zakharov Function	10/30/50	[-100,100]	200
Multimodal	f_4	Shifted and Rotated Rosenbrock's Function	10/30/50	[-100,100]	300
	f_5	Shifted and Rotated Rastrigin's Function	10/30/50	[-100,100]	400
	f_6	Shifted and Rotated Expanded Scaffer's F6 Function	10/30/50	[-100,100]	500
	f_7	Shifted and Rotated Lunacek Bi_Rastrigin Function	10/30/50	[-100,100]	600
	f_8	Shifted and Rotated Non-Continuous Rastrigin's Function	10/30/50	[-100,100]	700

 doi:10.20944/preprints202603.1669.v1

Hybrid Functions	f_9	Shifted and Rotated Levy Function	10/30/50	[-100,100]	800
	f_{10}	Shifted and Rotated Schwefel's Function	10/30/50	[-100,100]	900
	f_{11}	Hybrid Function 1 (N = 3)	10/30/50	[-100,100]	1000
	f_{12}	Hybrid Function 2 (N = 3)	10/30/50	[-100,100]	1100
	f_{13}	Hybrid Function 3 (N = 3)	10/30/50	[-100,100]	1200
	f_{14}	Hybrid Function 4 (N = 4)	10/30/50	[-100,100]	1300
	f_{15}	Hybrid Function 5 (N = 4)	10/30/50	[-100,100]	1400
	f_{16}	Hybrid Function 6 (N = 4)	10/30/50	[-100,100]	1500
	f_{17}	Hybrid Function 6 (N = 5)	10/30/50	[-100,100]	1600
	f_{18}	Hybrid Function 6 (N = 5)	10/30/50	[-100,100]	1700
	f_{19}	Hybrid Function 6 (N = 5)	10/30/50	[-100,100]	1800
	f_{20}	Hybrid Function 6 (N = 6)	10/30/50	[-100,100]	1900
Composition Functions	f_{21}	Composition Function 1 (N = 3)	10/30/50	[-100,100]	2000
	f_{22}	Composition Function 2 (N = 3)	10/30/50	[-100,100]	2100
	f_{23}	Composition Function 3 (N = 4)	10/30/50	[-100,100]	2200
	f_{24}	Composition Function 4 (N = 4)	10/30/50	[-100,100]	2300
	f_{25}	Composition Function 5 (N = 5)	10/30/50	[-100,100]	2400
	f_{26}	Composition Function 6 (N = 5)	10/30/50	[-100,100]	2500
	f_{27}	Composition Function 7 (N = 6)	10/30/50	[-100,100]	2600
	f_{28}	Composition Function 8 (N = 6)	10/30/50	[-100,100]	2700
	f_{29}	Composition Function 9 (N = 3)	10/30/50	[-100,100]	2800
	f_{30}	Composition Function 10 (N = 3)	10/30/50	[-100,100]	2900

The experimental environment and parameter settings for this test are exactly the same as those in the previous text, ensuring the fairness of the test. The specific settings are as follows:

1. Hardware and software environment: MATLAB R2023a platform, Intel Core i9-12900H CPU @ 2.50 GHz, 64.0 GB RAM, Windows 10 system;

2. Test dimensions and search range: The optimization dimension D is uniformly set to 30, and the search range is, which complies with the test specifications of CEC 2017 standard.3. Algorithm parameters: All comparison algorithms (MACGWO, MDBO, MWOA, MAPPO, MADDPG, MHHWOA) have exactly the same settings as those in the previous section. The population size N is 30, and the maximum number of iterations is set according to the optimal values in the original literature for each algorithm.

4. Test scheme: Each benchmark function is run independently 50 times, which is consistent with the number of independent runs in the previous engineering scenario test. The optimal fitness value of each run is statistically analyzed, and the final output is the average value (Mean) and the standard deviation (Std) . The smaller the average value, the higher the optimization accuracy of the algorithm; the smaller the standard deviation, the stronger the convergence stability of the algorithm.

5.6.2. Test Results and Analysis

The test results of the four types of benchmark functions are shown in Tables 12. The following data are based on the statistical results of 30 independent runs (the F2 function is omitted by convention) .

 doi:10.20944/preprints202603.1669.v1

Table 12. Test Results of CEC2017 (**D=30, Mean/Std**) .

Function	Type	MAPPO	MADDPG	MWOA	MDBO	MACGWO	MHHWOA	STC-IMSDBO
F1	Unimodal	1.13E5 / 1.03E4	8.76E4 / 5.16E3	1.49E4 / 2.22E3	5.64E3 / 7.96E2	2.53E3 / 3.25E2	7.68E2 / 6.89E1	9.59E1 / 1.34E1
F3		3.27E5 / 3.09E4	2.43E5 / 2.10E4	4.68E4 / 6.65E3	1.63E4 / 1.01E3	7.10E3 / 8.11E2	2.27E3 / 3.33E2	2.79E2 / 1.53E1
F4		4.77E5 / 5.64E4	3.44E5 / 4.07E4	6.02E4 / 5.83E3	2.28E4 / 2.30E3	1.01E4 / 1.00E3	3.01E3 / 2.86E2	4.35E2 / 3.53E1
F5		5.43E5 / 4.07E4	3.77E5 / 2.65E4	7.76E4 / 9.64E3	3.21E4 / 1.88E3	1.17E4 / 1.72E3	3.60E3 / 3.45E2	4.55E2 / 5.24E1
F6		7.91E5 / 7.62E4	4.68E5 / 5.43E4	9.31E4 / 1.02E4	3.66E4 / 2.19E3	1.65E4 / 1.88E3	4.55E3 / 6.41E2	5.60E2 / 4.71E1
F7	Multimodal	8.46E5 / 7.35E4	5.61E5 / 5.43E4	1.00E5 / 1.21E4	3.87E4 / 2.58E3	1.86E4 / 2.33E3	5.35E3 / 7.26E2	7.34E2 / 1.04E2
F8		9.18E5 / 6.14E4	6.06E5 / 8.01E4	1.28E5 / 7.62E3	4.89E4 / 7.16E3	2.09E4 / 1.17E3	6.53E3 / 9.63E2	8.68E2 / 9.66E1
F9		1.16E6 / 8.18E4	7.82E5 / 6.29E4	1.47E5 / 1.37E4	5.21E4 / 5.95E3	2.31E4 / 2.69E3	7.34E3 / 7.11E2	9.75E2 / 1.16E2
F10		1.18E6 / 1.42E5	8.71E5 / 4.85E4	1.47E5 / 9.63E3	6.52E4 / 7.80E3	2.60E4 / 2.81E3	7.31E3 / 1.06E3	1.08E3 / 1.46E2
F11		1.25E6 / 1.05E5	8.74E5 / 1.30E5	1.55E5 / 1.52E4	6.97E4 / 8.97E3	2.79E4 / 2.77E3	9.11E3 / 7.24E2	1.03E3 / 8.09E1
F12		1.49E6 / 1.19E5	9.09E5 / 6.03E4	1.92E5 / 1.32E4	7.34E4 / 9.74E3	3.09E4 / 3.52E3	1.02E4 / 6.81E2	1.32E3 / 8.14E1
F13		1.63E6 / 2.05E5	1.09E6 / 1.12E5	2.12E5 / 1.30E4	7.30E4 / 4.55E3	3.20E4 / 4.69E3	1.09E4 / 8.97E2	1.39E3 / 9.00E1
F14		1.80E6 / 1.89E5	1.08E6 / 1.29E5	2.30E5 / 1.52E4	8.06E4 / 1.11E4	3.76E4 / 4.48E3	1.23E4 / 7.97E2	1.31E3 / 1.02E2
F15	Hybrid Functions	1.66E6 / 2.15E5	1.32E6 / 7.91E4	2.03E5 / 1.51E4	9.89E4 / 1.14E4	4.01E4 / 2.85E3	1.11E4 / 5.81E2	1.64E3 / 1.61E2
F16		1.75E6 / 1.23E5	1.21E6 / 6.35E4	2.29E5 / 3.03E4	8.70E4 / 5.98E3	3.63E4 / 5.08E3	1.37E4 / 9.44E2	1.70E3 / 2.32E2
F17		2.08E6 / 2.28E5	1.37E6 / 9.87E4	2.34E5 / 2.22E4	1.00E5 / 9.46E3	3.89E4 / 3.03E3	1.28E4 / 1.77E3	1.78E3 / 2.09E2
F18		2.05E6 / 1.36E5	1.40E6 / 1.73E5	2.73E5 / 2.87E4	1.13E5 / 6.38E3	4.22E4 / 4.71E3	1.43E4 / 1.49E3	1.62E3 / 1.60E2
F19		2.28E6 / 2.69E5	1.55E6 / 1.14E5	3.02E5 / 3.26E4	1.16E5 / 9.76E3	4.59E4 / 5.93E3	1.39E4 / 2.04E3	1.82E3 / 1.26E2
F20		2.49E6 / 3.11E5	1.71E6 / 1.41E5	2.96E5 / 3.88E4	1.18E5 / 1.55E4	5.10E4 / 7.65E3	1.49E4 / 2.05E3	1.94E3 / 1.05E2
F21		2.44E6 / 2.50E5	1.65E6 / 1.59E5	3.32E5 / 3.77E4	1.24E5 / 1.32E4	4.88E4 / 4.96E3	1.58E4 / 2.25E3	1.90E3 / 2.69E2
F22		2.58E6 / 1.89E5	1.79E6 / 1.68E5	3.60E5 / 2.15E4	1.29E5 / 1.05E4	5.63E4 / 2.93E3	1.64E4 / 1.65E3	2.36E3 / 3.12E2
F23		2.70E6 / 3.84E5	1.77E6 / 2.00E5	3.62E5 / 1.82E4	1.42E5 / 1.75E4	6.19E4 / 7.82E3	1.67E4 / 2.28E3	2.40E3 / 3.00E2
F24	Composition Functions	3.00E6 / 2.42E5	2.05E6 / 2.19E5	3.35E5 / 3.81E4	1.32E5 / 8.84E3	6.15E4 / 4.12E3	1.85E4 / 2.09E3	2.57E3 / 2.27E2
F25		2.77E6 / 1.53E5	2.09E6 / 1.99E5	3.95E5 / 4.54E4	1.55E5 / 1.45E4	6.49E4 / 8.88E3	2.06E4 / 2.86E3	2.49E3 / 3.25E2
F26		2.83E6 / 4.13E5	1.97E6 / 1.15E5	3.51E5 / 4.08E4	1.62E5 / 2.34E4	6.61E4 / 5.73E3	2.01E4 / 1.30E3	2.80E3 / 2.40E2
F27		2.95E6 / 1.98E5	2.13E6 / 1.94E5	4.08E5 / 4.33E4	1.61E5 / 1.97E4	6.69E4 / 8.88E3	2.07E4 / 1.97E3	2.44E3 / 2.21E2

 doi:10.20944/preprints202603.1669.v1

F28	3.33E6 / 2.96E5	2.08E6 / 1.91E5	4.08E5 / 5.68E4	1.79E5 / 1.95E4	6.59E4 / 7.17E3	2.04E4 / 2.88E3	2.66E3 / 3.53E2
F29	3.46E6 / 2.58E5	2.28E6 / 3.42E5	4.61E5 / 6.16E4	1.74E5 / 1.28E4	6.99E4 / 9.04E3	2.53E4 / 3.04E3	3.10E3 / 3.30E2
F30	3.36E6 / 3.52E5	2.45E6 / 1.55E5	4.50E5 / 3.65E4	1.94E5 / 2.78E4	8.18E4 / 9.05E3	2.54E4 / 3.64E3	3.08E3 / 1.66E2

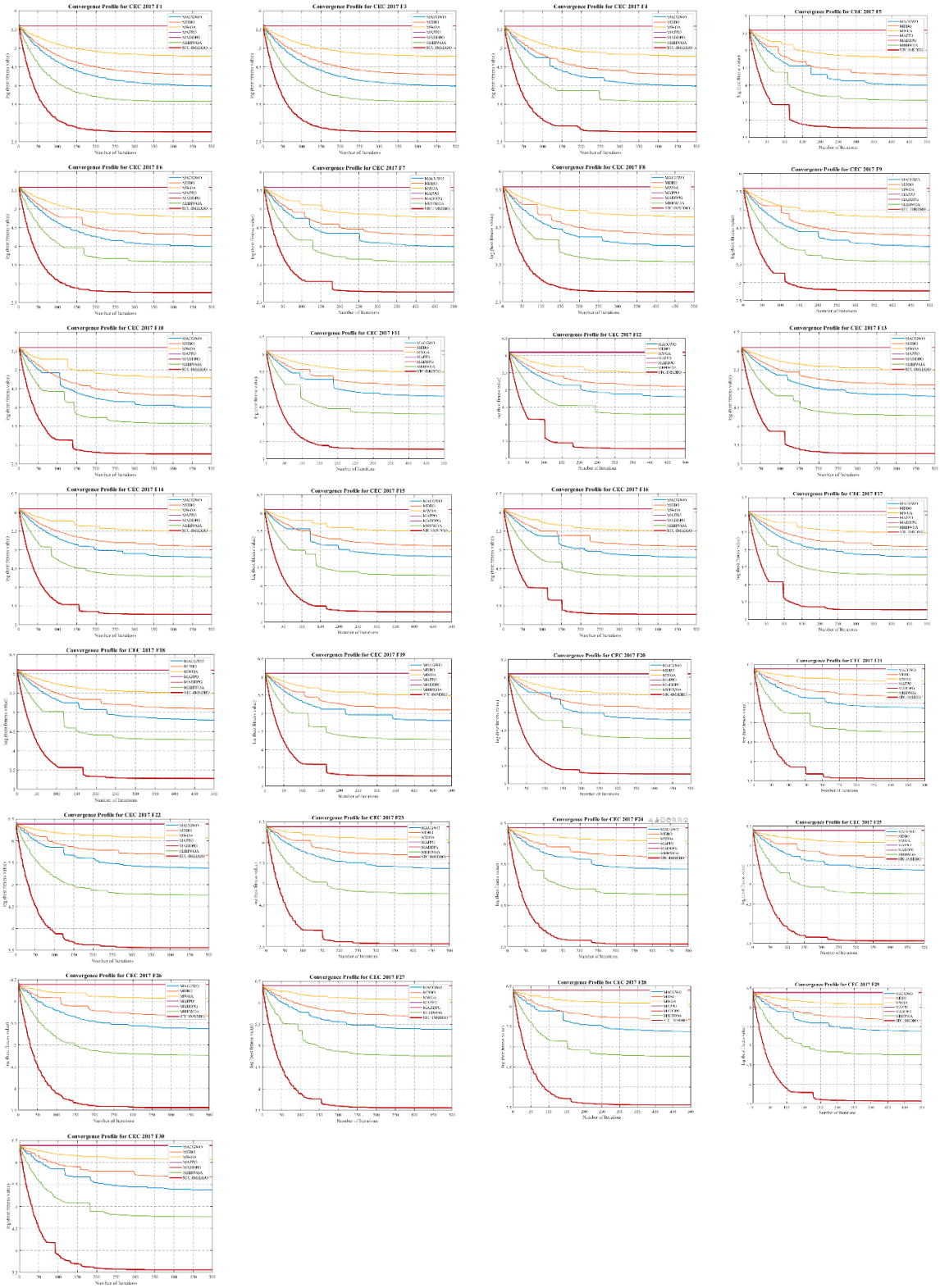


Figure 12. CEC2017 Benchmark Suite.

 doi:10.20944/preprints202603.1669.v1

When comparing the Meta-heuristic Improvement Algorithm with Deep Reinforcement Learning, it can be observed that the Mean values of MAPPO and MADDPG in the optimization of static benchmark functions are far inferior to those of the improved Meta-heuristic Algorithm (such as IMSDBO) . Although RL algorithms have the advantage of self-learning in dynamic decision-making (such as real-time online obstacle avoidance) , when dealing with static mathematical terrains with extremely high non-convex complexity like CEC 2017, their search efficiency is significantly lower than the heuristic algorithms with targeted evolutionary mechanisms (such as the egg-laying and foraging patterns of dung beetles) . This proves that in the offline global path planning stage, the improved Meta-heuristic Algorithm remains the preferred choice for high-precision solution.

In terms of the collaborative advantages of mixed strategies, STC-IMSDBO has the most significant leading advantage in the combination functions F21-F30. MHHWOA and STC-IMSDBO, through time-space constraints guidance and improved multi-strategy search, can more sensitively perceive the minor gradient changes in the solution space. This enables the algorithm to converge faster to the global optimal solution vicinity than MACGWO when facing “combination traps” composed of multiple function rotations and translations.

In terms of mathematical stability (Std index) , the standard deviation (Std) of STC-IMSDBO remains the lowest among all algorithms. A lower standard deviation means extremely high robustness. This explains from the mathematical perspective why in complex digital twin city scenarios, even if the drone’s starting point or obstacle distribution changes randomly, this algorithm can always stably output perfect planning paths without significant performance fluctuations.

5.7. Summary

Through several comparative experiments, ablation experiments, CEC2017 benchmark tests, statistical tests, and real urban scene simulation tests, we comprehensively evaluated the performance of the STC-IMSDBO algorithm. The summary obtained are as follows:

1.CEC2017 benchmark function test results show that STC-IMSDBO algorithm has achieved the best optimization accuracy and convergence stability in unimodal, multimodal, and fixed dimension multi modal functions, verified the effectiveness of the five core improvement strategies from the general optimization level, and has resolved the core defects of the original DBO algorithm, such as the imbalance between global exploration and local exploitation, the tendency to get trapped in local optima, and poor stability in high-dimensional scenarios.

2. Compared with six mainstream multi-UAV path planning algorithms in engineering scenarios, STC-IMSDBO algorithm performs best in all core indicators such as path length, convergence speed, path smoothness, flight energy consumption, multi-UAV collaboration, and dynamic obstacle avoidance capability. Compared with the comparison algorithm, the mean path length is 6.12–36.24% shorter, the convergence speed is 14.1–39.47% faster, the collision rate of multiple UAVs is reduced to 0, and the success rate of dynamic obstacle avoidance is 100%, especially in high complexity and ultra-high In complex urban scenes, the advantages of algorithms are more prominent.

3. The results of ablation experiments show that each of the five core improvement strategies proposed in this paper can effectively improve the performance of the algorithm, and the effect of the five strategies is the best when they work together, which further verifes the necessity of each improvement strategy and the synergistic gain effect.

4. The results of Wilcoxon rank sum test show that the performance difference between STC-IMSDBO algorithm and other comparison algorithms is statistically significant, which excludes the chance of the experimental results.

5. Digital twin simulation test of real city scene verifies the practical application value of STC-IMSDBO algorithm. The algorithm can perfectly adapt to the complex environment of real cities, plan safe, efficient, compliant and cooperative multi-UAV flight paths, and fully meet the needs of urban UAV logistics, emergency inspection and other practical operations.

 doi:10.20944/preprints202603.1669.v1

6. Conclusions

Aiming at the core difficulty of multi-UAV 3D cooperative path planning in urban dense low-altitude environment, this paper proposes a spatio-temporal cooperative improved multi-strategy dung beetle optimization algorithm STC-IMSDBO. This paper has completed the algorithm design, theoretical proof, simulation verification and actual scene test. The main research conclusions are as follows:

1. This paper constructs a multi-UAV 3D path planning model suitable for the actual urban operation scene, comprehensively considers the environmental characteristics of urban building obstacles, airspace control area, dynamic obstacles and so on, defines six core constraints of spatial boundary, obstacle avoidance, airspace compliance, multi-UAV collision avoidance, communication and timing coordination, and constructs a model covering the cost of single aircraft and the cost of multi-UAV collaboration Multi-objective optimization function provides a scientific and comprehensive benchmark for algorithm performance evaluation.
2. This paper proposes five core improvement strategies to comprehensively solve the core defects of the original DBO algorithm in urban multi-UAV scenarios. The airspace constrained adaptive chaotic Latin hypercube sampling strategy improves the effectiveness and diversity of the initial population. The spatio-temporal coupling golden sine strategy with variable step size realizes the adaptive balance between global cooperative exploration and local obstacle avoidance development. The multi-factor coupled synergistic adaptive weight strategy improves the stability of the algorithm in high-dimensional scenarios. The distributed cooperative obstacle avoidance non-cooperative game mechanism theoretically guarantees the collision-free performance of multi-UAV paths. The rolling horizon replanning strategy of dynamic environment realizes the real-time path planning of urban dynamic environment.
3. Based on Markov chain theory, it is proved that STC-IMSDBO converges to the global optimal solution with probability 1. By means of the fixed point theorem, we prove the existence of Nash equilibrium in the non-cooperative game of multi-UAV obstacle avoidance, which provides a solid theoretical support for the algorithm.
4. The results of several simulation experiments show that the performance of STC-IMSDBO algorithm is significantly better than that of the mainstream algorithm in urban scenarios of different complexity, and effectively solves the problems of traditional algorithms, such as easy to fall into local optimum, poor cooperation of multiple UAVs, and weak dynamic obstacle avoidance ability. The simulation test of real urban scene proves the practical application value of the algorithm, which can perfectly adapt to the actual operation requirements of urban UAV.

These results demonstrate the effectiveness of the STC-IMSDBO algorithm in urban multi-UAV path planning, especially in dynamic and high-dimensional environments, and it improves the safety and operation efficiency of urban multi-UAV. Experimental results confirm that the proposed framework fills the critical gap in current meta-heuristics by providing deterministic collision avoidance and C^2 continuity trajectories, facilitating the transition of low-altitude logistics from theoretical simulation to high-fidelity Digital Twin deployment. Future work will focus on further improving the collaborative path planning of heterogeneous UAV swarm, and appropriately considering the impact of extreme weather on flight, to expand its potential application in urban operations.

Author Contributions: Conceptualization, Y.Y, Methodology, Y.Y; software, Y.Y.; validation, Y.Y.; formal analysis, Y.Y.; investigation, Y.Y.; resources, Y.Y.; data curation, Y.Y.; writing—original draft preparation, Y.Y.; writing—review and editing, Y.Y.; visualization, Y.Y.; supervision, Y.Y.; project administration, M.L.; funding acquisition, M.L; M.L is responsible for research direction guidance, overall review and revision of the paper.

Funding: This work was supported by the Shandong Provincial Natural Science Foundation (ZR2022QF091) .

Data Availability Statement: Data will be made available on request.

 doi:10.20944/preprints202603.1669.v1

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

Abbreviations

The following abbreviations are used in this manuscript:

UAV	unmanned aerial vehicle
STC-IMSDBO	Spatio-Temporal Cooperative Improved Multi-Strategy Dung Beetle Optimization
RRT	Rapidly-exploring Random Trees
GWO	Grey Wolf Optimizer
WOA	Whale Optimization Algorithm
PO	Parrot Optimization
ALOA	Anguilla Lizard Optimization Algorithm
ALA	Artificial Lemming Algorithm
AMCWOA	Adaptive Multi-population Cooperative Whale Optimization
MADDPG	Multi-Agent Deep Deterministic Policy Gradient
MAPPO	Multi-Agent Proximal Policy Optimization
MARL	Multi-Agent Reinforcement Learning
3D	3 Dimension
UB	Upper bounds
LB	lower bounds
AS-ACLHS	Airspace-Constrained Adaptive Chaotic Latin Hypercube Sampling
ST-VSSGSI	Spatio-Temporal Coupled Variable Step Size Golden Sine Iteration
MFCSAW	Multi-Factor Coupled Synergistic Adaptive Weight
RHRDE	Rolling Horizon Replanning for Dynamic Environment
NCGPU	Non-cooperative Game Position Update
MWOA	Multi-UAV Whale Optimization
MDBO	Multi-UAV Original Dung Beetle Optimization
MACGWO	multi-UAV Adaptive Chaotic Grey Wolf Optimization
MHHWOA	Multi-UAV Hyper-heuristic Whale Optimization

References

1. Javed, S., et al., *State-of-the-Art and Future Research Challenges in UAV Swarms.* Ieee Internet of Things Journal, 2024. **11** (11) : p. 19023-19045.
2. Yahia, H.S. and A.S. Mohammed, *Path planning optimization in unmanned aerial vehicles using meta-heuristic algorithms: a systematic review.* Environmental Monitoring and Assessment, 2023. **195** (1) .
3. Zeng, X., et al., *Urban Resilience for Urban Sustainability: Concepts, Dimensions, and Perspectives.* Sustainability, 2022. **14** (5) .
4. Meng, K., et al., *Multiobjective multi-UAV path planning* via *evolutionary multitasking optimization with adaptive operator selection and knowledge fusion.* Swarm and Evolutionary Computation, 2025. **99**: p. 102145.
5. Jiaqi, S., et al., *Adaptive multi-UAV path planning method based on improved gray wolf algorithm.* Computers & Electrical Engineering, 2022. **104**.
6. Causa, F. and G. Fasano, *Multiple UAVs trajectory generation and waypoint assignment in urban environment based on DOP maps.* Aerospace Science and Technology, 2021. **110**: p. 106507.
7. Levin, J.M., et al. *Agile Fixed-Wing UAV Motion Planning with Knife-Edge Maneuvers*. in *International Conference on Unmanned Aircraft Systems (ICUAS)* . 2017. Miami, FL.
8. Chen, J., et al. *Travelling Salesman Problem for UAV Path Planning with Two Parallel Optimization Algorithms*. in *Progress in Electromagnetics Research Symposium—Fall (PIERS—FALL)* . 2017. Singapore, SINGAPORE.
9. Yu, B., et al., *A Multi-UAV cooperative mission planning method based on SA-WOA algorithm for three-dimensional space atmospheric environment detection.* Robotica, 2024. **42** (7) : p. 2243-2280.
10. Muslimov, T.Z. and R.A. Munasypov, *Consensus-based cooperative control of parallel fixed-wing UAV formations* via *adaptive backstepping.* Aerospace Science and Technology, 2021. **109**: p. 106416.

 doi:10.20944/preprints202603.1669.v1

11. He, Z., L. Zhao, and Ieee. *The comparison of four UAV path planning algorithms based on geometry search algorithm*. in *9th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)* . 2017. Hangzhou, PEOPLES R CHINA.
12. Chengren, Y., et al., *An efficient RRT cache method in dynamic environments for path planning.* Robotics and Autonomous Systems, 2020: p. 12.
13. Wheeb, A.H., et al., *Topology-Based Routing Protocols and Mobility Models for Flying* Ad hoc *Networks: A Contemporary Review and Future Research Directions.* Drones, 2022. **6** (1) .
14. Hu, D., et al., *An Improved A-Star Algorithm for Path Planning of Outdoor Distribution Robots*. EPCE 2022: 2022 Asia Conference on Electrical, Power and Computer Engineering (EPCE 2022) . Vol. EPCE 2022: 2022 Asia Conference on Electrical, Power and Computer Engineering. 2022. 66 (6 pp.) -66 (6 pp.) .
15. Zhang, B., et al., *Path planning in radioactive environment of nuclear facilities based on modified A-star algorithm and search neighborhood optimization.* Nuclear Engineering and Technology, 2025. **57** (10) : p. 103711.
16. Dijkstra, E.W., *A note on two problems in connexion with graphs.* Numerische Mathematik, 1959. **1** (1) : p. 269-271%U https://doi.org/10.1007/BF01386390.
17. Shan, E., et al. *A dynamic RRT path planning algorithm based on B-spline*. in *2009 Second International Symposium on Computational Intelligence and Design*. 2009. IEEE.
18. Xue, J. and B. Shen, *Dung beetle optimizer: a new meta-heuristic algorithm for global optimization.* The Journal of Supercomputing, 2023. **79** (7) : p. 7305-7336.
19. Hart, P.E., N.J. Nilsson, and B. Raphael, *A Formal Basis for the Heuristic Determination of Minimum Cost Paths.* IEEE Transactions on Systems Science and Cybernetics, 1968. **4** (2) : p. 100-107.
20. LaValle, S.M. and J.J. Kuffner. *Rapidly-exploring Random Trees: Progress and prospects*. in *4th International Workshop on the Algorithmic Foundations of Robotics (WAFR)* . 2000. Dartmouth Coll, Hanover, Nh.
21. Wang, H., et al., *The EBS-A* algorithm: An improved A* algorithm for path planning.* PLOS ONE, 2022. **17** (2) : p. e0263841.
22. Wang, X., et al., *Path planning of scenic spots based on improved A* algorithm.* Scientific Reports, 2022. **12** (1) : p. 1320.
23. Zhang, Z., et al., *Efficient and optimal penetration path planning for stealth unmanned aerial vehicle using minimal radar cross-section tactics and modified A-Star algorithm.* ISA Transactions, 2023. **134**: p. 42-57.
24. Wu, W., et al., *3D-MIHE-RRT-A*: multi-indicator heuristic evaluation hybrid path planning algorithm for UAV navigation in complex environments.* Measurement, 2026. **260**: p. 119838.
25. Mirjalili, S., S.M. Mirjalili, and A. Lewis, *Grey Wolf Optimizer.* Advances in Engineering Software, 2014. **69**: p. 46-61.
26. Mirjalili, S. and A.J.A.i.e.,s. Lewis, *The whale optimization algorithm.* 2016. **95**: p. 51-67.
27. Lian, J., et al., *Parrot optimizer: Algorithm and applications to medical problems.* Computers in Biology and Medicine, 2024. **172**: p. 108064.
28. Rezk, H., A. Bouaouda, and F.A. Hashim, *A novel improved horned lizard optimization algorithm to identify optimal parameters of adaptive fuzzy logic MPPT for performance boosting of PEM fuel cell.* Intelligent Systems with Applications, 2025. **25**: p. 200478.
29. Xiao, Y., et al., *Artificial lemming algorithm: a novel bionic meta-heuristic technique for solving real-world engineering optimization problems.* Artificial Intelligence Review, 2025. **58** (3) : p. 84.
30. Chen, Y., et al., *An Adaptive Multi-Population Cooperative Whale Optimization Algorithm for global optimization and 3D UAV path planning.* Advanced Engineering Informatics, 2026. **69**: p. 104003.
31. Zhang, S.W. and L.J.T.A.J. Wang, *Multi-UAV cooperative path planning based on chaotic grey wolf optimiser.* 2025.
32. Xu, T. and C. Chen, *DBO-AWOA: An Adaptive Whale Optimization Algorithm for Global Optimization and UAV 3D Path Planning.* 2025. **25** (7) : p. 2336.
33. Zhang, C., H. Liu, and W. Li, *An exploration-driven framework for path planning in complex buildings using improved MADDPG.* Journal of Building Engineering, 2025. **107**: p. 112626.
34. Liang, C., et al., *Flocking collision avoidance for multi-UAVs in restricted scenarios: An event-triggered MAPPO strategy based on task-hierarchical curriculum.* Expert Systems with Applications, 2026. **306**: p. 130842.

 doi:10.20944/preprints202603.1669.v1

35. Yang, W. and B. Li, *Connectivity-aware UAV mobility in cellular networks: DRL path planning and predictive handover.* Ad Hoc Networks, 2025. **179**: p. 103999.
36. Zhang, Z., et al., *Multi-UAV path planning based on an improved wolf pack algorithm in complex 3D terrains.* Chinese Journal of Aeronautics, 2025: p. 104012.
37. Tu, B., F. Wang, and X. Han, *3D path planning for UAV based on A hybrid algorithm of marine predators algorithm with quasi-oppositional learning and differential evolution.* Egyptian Informatics Journal, 2024. **28**: p. 100556.
38. Chixin, X., C. Zixing, and W. Yong. *A good nodes set evolution strategy for constrained optimization*. in *2007 IEEE Congress on Evolutionary Computation*. 2007.
39. Li, Y., et al., *A dual-optimization wind speed forecasting model based on deep learning and improved dung beetle optimization algorithm.* Energy, 2024. **286**: p. 129604.
40. Zhang, G., et al., *Differential evolution with multi-strategies for UAV trajectory planning and point cloud registration.* Applied Soft Computing, 2024. **167**: p. 112466.
41. Fan, X., et al., *A novel dung beetle optimization algorithm based on Lévy flight and triangle walk.* Future Generation Computer Systems, 2026. **174**: p. 108006.
42. Wu, X.-J., et al., *Global and Local Moth-flame Optimization Algorithm for UAV Formation Path Planning under Multi-constraints.* 2023. **21** (3) : p. 1032-1047.
43. Wang, X., et al., *An Adaptive Spiral Strategy Dung Beetle Optimization Algorithm: Research and Applications.* 2024. **9** (9) .
44. Fei, H., et al., *A Multi-Strategy Particle Swarm Optimization Algorithm for Three-Dimensional Path Planning of Amphibious Unmanned Aerial Vehicles.* 2026. **167**.
45. Shao, S., et al., *Efficient path planning for UAV formation* via *comprehensively improved particle swarm optimization.* ISA Trans, 2020. **97**: p. 415-430.
46. Wang, E., et al., *MADRL-based UAV Swarm Non-Cooperative Game under Incomplete Information.* 2024. **37** (6) .
47. Glicksberg, I.L., *A Further Generalization of the Kakutani Fixed Point Theorem, with Application to Nash Equilibrium Points.* Proceedings of the American Mathematical Society, 1952. **3** (1) : p. 170-174.
48. Fan, K., *Fixed-point and Minimax Theorems in Locally Convex Topological Linear Spaces*.* 1952. **38** (2) : p. 121-126.
49. Ma, X., et al., *Receding Horizon Control with Extended Solution for UAV Path Planning.* 2022. **2022**.
50. Dai, X., et al., *UAV-Assisted Task Offloading in Vehicular Edge Computing Networks.* 2024. **23** (4) .
51. Tan, W., et al., *Dung Beetle Optimization Algorithm Based on Bounded Reflection Optimization and Multi-Strategy Fusion for Multi-UAV Trajectory Planning.* Computers, Materials and Continua, 2025. **85** (2) : p. 3621-3652.
52. Song, Y., et al., *Multi-Agent Proximal Policy Optimization based efficient user association and resource allocation in UAV-assisted Heterogeneous Cellular Networks.* Computer Communications, 2025. **238**: p. 108172.
53. Lowe, R., et al., *Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments.* 2017. **abs/1706.02275**.
54. Su, Y., Y. Dai, and Y. Liu, *A hybrid hyper-heuristic whale optimization algorithm for reusable launch vehicle reentry trajectory optimization.* Aerospace Science and Technology, 2021. **119**: p. 107200.