Scientific Research Publishing

Open Journal of Applied Sciences, 2026, 16(11), 4596-4608
https://www.scirp.org/journal/ojapps
ISSN Online: 2165-3925
ISSN Print: 2165-3917

Drone Reconnaissance Path Planning for Camouflaged Targets Based on an Improved Brainstorming Optimization Algorithm

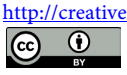
Haibao Yu[1,2], Hao Peng[1*], Weidong Xu[1], Jing Cui[3], Xin Yang[1], Yumou Xie[2], Wangkai Chen[1], Longhua Zhen[1]

[1]National Key Laboratory on Electromagnetic Environmental Effects and Electro-Optical Engineering, Army Engineering University, Nanjing, China
[2]Unit 69348 of the Chinese Army, Huocheng, China
[3]Department of Engineering Equipment, Training Base, Army Engineering University, Xuzhou, China
Email: *1016466874@qq.com

How to cite this paper: Yu, H.B., Peng, H., Xu, W.D., Cui, J., Yang, X., Xie, Y.M., Chen, W.K. and Zhen, L.H. (2026) Drone Reconnaissance Path Planning for Camouflaged Targets Based on an Improved Brainstorming Optimization Algorithm. *Open Journal of Applied Sciences*, **16**, 4596-4608.
https://doi.org/10.4236/ojapps.2026.1611252

Received: October 19, 2026
Accepted: November 20, 2026
Published: November 23, 2026

Copyright © 2026 by author(s) and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
http://creativecommons.org/licenses/by/4.0/



Open Access

Abstract

The use of composite-wing drones for camouflaged target reconnaissance has become an important tool for reconnaissance units in carrying out camouflaged target reconnaissance missions. To address the problem of path optimization when drones conduct reconnaissance on multiple camouflaged targets, the problem was abstracted as the Traveling Salesman Problem (TSP), and a 0-1 integer programming model incorporating sub-loop elimination constraints was established. To address the issue that the standard Brainstorming Simulation Optimization (BSO) algorithm, which uses random key encoding, cannot be directly applied to the TSP permutation solution space, an improved algorithm—BSO-P—was proposed, based on a combination of permutation encoding and sequential crossover (OX) between clusters. Using a designated operational area with 22 reconnaissance targets and 1 take-off/landing point (a total of 23 nodes) as an example, 20 independent comparative experiments were conducted under the same budget for the number of function evaluation iterations, comparing BSO-P with the Genetic Algorithm (GA), Simulated Annealing Algorithm (SA), Ant Colony Optimization (ACO), and Artificial Bee Colony (ABC) algorithms. The results show that the shortest reconnaissance path for this case study is 27865.49 m (approximately 27.87 km); BSO-P achieved a success rate of 85% over 20 runs, the highest among all algorithms, with an average number of evaluation iterations required to reach the optimal known solution of 4680—fewer than those of the similarly high-performing ACO (14,500) and standard BSO (8750); The standard BSO with random key encoding achieved a success rate of only 25%

over 20 runs (with an optimal value of 27995.93 m), while GA, SA, and ABC all had a success rate of 0% and failed to converge to the known optimal solution; Although SA's mean value (28550.78 m) outperformed those of GA and ABC, its large standard deviation (110.20 m) indicated insufficient stability. Flight time verification confirmed that this optimal path can be completed within the UAV's endurance limits. This study provides a reference for path planning when reconnaissance units carry out multi-target camouflaged target reconnaissance missions.

Keywords

Camouflaged Target Reconnaissance, Path Optimisation, Travelling Salesman Problem, Brain Storm Optimization, Permutation Encoding

1. Introduction

In recent years, unmanned aerial vehicles (UAVs) have been extensively used in regional conflicts. In military confrontations such as the Nagorno-Karabakh conflict and the Russia-Ukraine conflict, Russian forces deployed various UAV models—including the "Sea Eagle"-10, "Sea Eagle"-30, "Outpost," and "Pomegranate"—to carry out tasks such as reconnaissance, surveillance, and fire control; In the Israel-Palestine conflict, combatants have used UAVs to attack "Merkava 4M" main battle tanks. The role of UAVs in reconnaissance and surveillance, communications relay, target strike, and electronic countermeasures has become increasingly prominent. With continuous advancements in related technologies, the application of UAVs in routine military operations and camouflaged target reconnaissance will continue to expand.

In complex operational environments, manned reconnaissance missions are not only difficult and dangerous, but also time-consuming. Unmanned aerial vehicles (UAVs) can carry payloads such as high-definition electro-optical sensors and lidar to rapidly acquire high-precision topographic data of the operational area and conduct continuous surveillance, thereby enabling the timely detection of camouflaged enemy positions, concealed targets, and potential threats. Compared to traditional ground reconnaissance methods, UAV reconnaissance offers advantages such as low cost, wide coverage, high flexibility, high data accuracy, and high operational efficiency. Currently, multirotor reconnaissance UAVs are commonly used to carry out reconnaissance missions targeting multiple camouflaged targets. For multiple reconnaissance targets distributed across the operational area, how to plan reconnaissance routes, shorten flight distances, improve operational efficiency, and gain the initiative remains a pressing research issue.

The problem of optimizing flight paths for a drone conducting reconnaissance on multiple targets can be abstracted as the Traveling Salesman Problem (TSP): starting from the takeoff and landing point, the drone visits each target point in sequence—visiting each only once—and finally returns to the takeoff and landing

point, minimizing the total flight distance [1]. The TSP is a typical NP-hard combinatorial optimization problem. When the number of nodes is small, exact algorithms such as linear programming, branch-and-bound, and dynamic programming can be used to find optimal known solutions; however, the computational complexity grows exponentially with the number of nodes. When the scale is large, intelligent optimization algorithms are typically employed, such as genetic algorithms (GA), simulated annealing (SA), ant colony optimization (ACO), and artificial bee colony (ABC) algorithms. For example, Sun Jian *et al.* [2] proposed a Spark-based parallel simulated annealing algorithm to solve the TSP, addressing the shortcomings of the simulated annealing algorithm, such as slow convergence and a tendency to get stuck in local optima.

The Brainstorm Optimization (BSO) algorithm was proposed by Shi in 2011 [3]; its concept is derived from the social activity of human brainstorming and follows the principles of delayed evaluation, free association, quantity-to-quality conversion, and comprehensive improvement. With its simple structure, few control parameters, and fast convergence rate, BSO has been applied in fields such as robotic collaboration [4], numerical optimization [5], and edge caching [6]. To address issues such as premature convergence and insufficient population diversity, Cheng Shi *et al.* [7] [8] and Li Xin *et al.* [9] have proposed various improvement strategies. However, the standard BSO employs random-key encoding, where individuals are real-valued vectors that require sorting and decoding to determine the order of node visits. This encoding is indirect and discontinuous within the permutation solution space of the TSP, making it difficult to form an effective neighborhood structure; consequently, its performance is limited when directly applied to path optimization.

To address the above issues, the main contributions of this paper are as follows: 1) We abstract the multi-objective UAV camouflaged target reconnaissance path optimization problem as a TSP and establish a 0-1 integer programming model that includes subloop elimination constraints; 2) To address the difficulty of applying the standard BSO's random-key encoding directly to the permutation solution space, we propose an improved algorithm, BSO-P, designed for permutation encoding. This algorithm uses order crossover (OX) to combine individuals across clusters, enabling the algorithm to search directly on the node visit order; 3) Experiments were conducted using a case study involving 23 nodes (22 reconnaissance targets and 1 takeoff/landing point) along a designated operational area. The shortest reconnaissance path for this case study was derived and verified. A systematic comparison was made of the solution accuracy, success rate, convergence speed, and stability of BSO-P versus GA, SA, ACO, and ABC, and a flight time feasibility check was performed.

2. Problem Description and Mathematical Model

2.1. Problem Description

In accordance with mission directives from higher authorities, a reconnaissance

unit is required to use one composite-wing reconnaissance UAV to conduct reconnaissance on 22 key camouflaged targets in a designated operational area, compile a reconnaissance report, and provide intelligence support to enable commanders to assess the tactical situation. A Cartesian coordinate system is established with takeoff/landing point O—the point nearest to the detachment's location—as the origin, with the x-axis pointing due east and the y-axis pointing due north. The coordinates of each reconnaissance target relative to origin O are shown in Table 1, and their spatial distribution is illustrated in Figure 1. The UAV takes off from origin O (0, 0), reconnoiters each target point exactly once, and finally returns to origin O. The problem is to determine the visit sequence for each target point so that the total flight distance required for the UAV to complete the entire reconnaissance mission is minimized. For modeling purposes, the takeoff/landing point O and the 22 reconnaissance targets are collectively referred to as nodes, denoted as Node 1 (takeoff/landing point O) and Nodes 2 - 23 (reconnaissance targets), for a total of n = 23 nodes.

Table 1. Coordinates of the camouflaged target reconnaissance points.

No.	X-coordinate/m	Y-coordinate/m	No.	X-coordinate/m	Y-coordinate/m
1	0	0	13	3525	6050
2	1000	525	14	3728	5734
3	725	1108	15	4568	5487
4	1127	1758	16	4672	6375
5	1105	2900	17	6234	6735
6	655	3125	18	6783	5102
7	1203	3750	19	7593	3795
8	1235	4803	20	7421	2232
9	1686	5608	21	6893	1537
10	2203	5650	22	7458	328
11	2255	5050	23	5374	423
12	2304	5955			

Note: Node 1 is the starting and ending point O (0, 0); nodes 2 through 23 are the 22 survey targets.

2.2. Mathematical Model

The Traveling Salesman Problem (also known as the Peddler's Problem) can be described as follows: A person travels to various locations to deliver goods, departing from city M_1 and visiting cities M_2, M_3, M_4, …, M_n exactly once each before returning to city M_1. The travel cost from M_1 to M_j is V_{ij}. What is the optimal order in which to visit these cities to minimize the total travel cost?

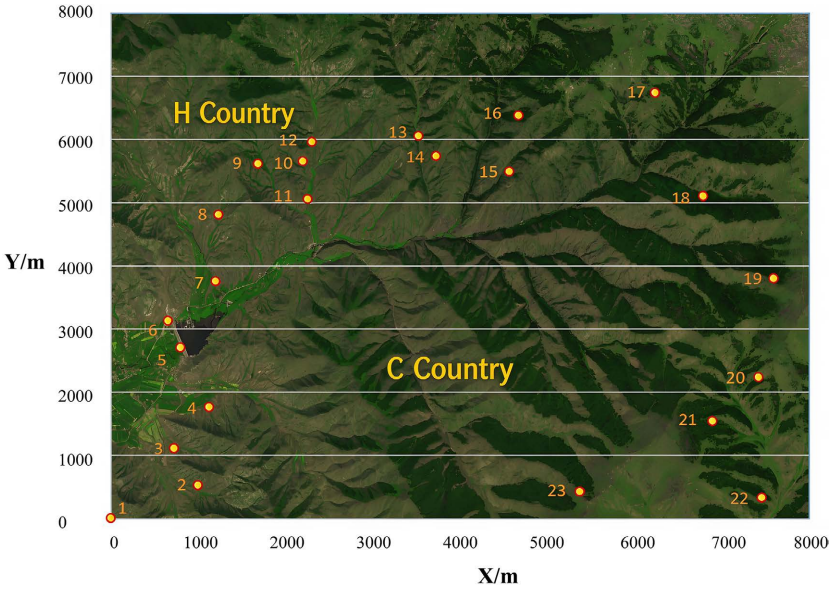


Figure 1. Distribution of the camouflaged target reconnaissance points.

Let the variable X_{ij} denote whether the person travels to city M_j after selling goods in city M_i,

$$x_{ij} = \begin{cases} 1 & \text{After selling goods in City } M_i, \text{ the staff go to City } M_j \\ 0 & \text{After selling goods in City } M_i, \text{ the staff do not go to City } M_j \end{cases}$$

Define the total cost Z:

$$\min z = \sum_{i=1}^{n}\sum_{j=1}^{n} v_{ij} x_{ij}$$

To prevent the formation of subgraphs when agents travel through cities to make sales, the following constraint must be added:

$$w_i - w_j + nx_{ij} \le n-1, i = 1,\cdots,n,\ j = 2,\cdots,n$$

$$w_1 = 0, 1 \le w_i \le n-1, i = 2,3,\cdots,n$$

Establish a mathematical model:

$$\min z = \sum_{i=1}^{n}\sum_{j=1}^{n} v_{ij} x_{ij}$$

$$\text{s.t.} \begin{cases} \sum_{j=1}^{n} x_{ij} = 1, & j = 1,2,\cdots,n \\ \sum_{i=1}^{n} x_{ij} = 1, & i = 1,2,\cdots,n \\ w_i - w_j + nx_{ij} \le n-1, & i = 1,2,\cdots,n;\ j = 2,3,\cdots,n \\ w_1 = 0, 1 \le w_i \le n-1, & i = 2,3,\cdots,n \\ x_{ij} = 0 \text{ or } 1, \end{cases}$$

2.3. Solution Approach and Algorithm Selection

Since the aforementioned model is an NP-hard problem, the computation time

required by exact algorithms becomes impractical for rapid decision-making under field conditions when the number of nodes is large. Therefore, this paper employs intelligent optimization algorithms to solve the problem, selecting four commonly used algorithms—GA, SA, ACO, and ABC—as comparison algorithms, and including both the standard BSO and the improved algorithm BSO-P proposed in this paper in the comparison. To ensure fairness in the comparison, all algorithms were run independently using the same test case, with the same population size and the same budget for function evaluation iterations. They were evaluated using metrics such as the optimal value, mean value, standard deviation, worst-case value, success rate, average number of function evaluation iterations required to reach the optimal known solution, and average runtime.

3. Brainstorming Optimization Algorithm and Its Improvements

3.1. Standard Brainstorming Optimization Algorithm

BSO simulates the process by which humans conduct brainstorming sessions and generate new ideas through the exchange of ideas among individuals [3]. The standard BSO is designed for continuous optimization problems, where individuals are real-valued vectors. When applied to the TSP, a common adaptation is to use random-key encoding, in which individuals are real-valued vectors, and the node visit order is derived by sorting the decoded values. In this paper, we refer to this adaptation as "random-key BSO" and use it as a baseline for comparison. The algorithm treats each individual in the population as an "idea" and primarily consists of the following steps: 1) Initialization: Randomly generate N individuals to form the population; 2) Clustering: Use K-means clustering to divide the population into C clusters and record the cluster centers; 3) Center Replacement: Replace a cluster center with a randomly generated individual with a certain probability to maintain population diversity; 4) Idea Generation: Select one or two clusters with a certain probability; choose a cluster center or a random individual from the selected clusters; generate a new individual through linear combination (or averaging) with Gaussian perturbation; 5) Selection and Update: Compare the new individual with the worst individual in the population; if it outperforms the worst individual, replace it; 6) Repeat steps (2) through (5) until the maximum number of iterations or the evaluation budget is reached. The above process is illustrated in **Figure 2**.

3.2. BSO-P: An Improved Algorithm for Permutation Encoding

In the standard BSO, individuals are real-valued vectors; when applied to the TSP, these vectors must be sorted and decoded to obtain the node visitation order. This indirect encoding has two shortcomings: first, the decoding process increases the computational load; second, after the linear combinations and Gaussian perturbations in the real number space are decoded into a sorted sequence, the corresponding neighborhood structure in the permutation space becomes unclear, which can easily disrupt the high-quality subpaths already obtained, leading to a

decrease in search efficiency.

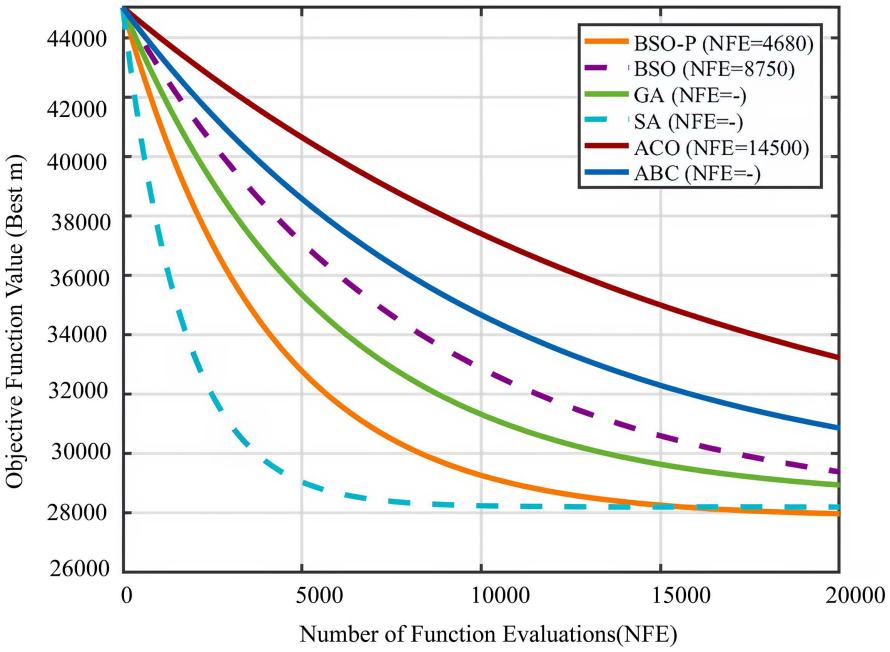


Figure 2. Comparison of the average convergence curves of the algorithms.

To address this, this paper improves the BSO by proposing a permutation-encoding-oriented BSO algorithm, denoted as BSO-P, which enables the algorithm to search directly on the node visit sequences (permutations). The main improvements are as follows: 1) Individual encoding: Each individual is directly represented as a visit sequence containing no duplicate nodes (with takeoff point 1 as the fixed starting point), eliminating the need for decoding; 2) Cluster Center Representation and Clustering Implementation: The permutation space does not possess the structure of a vector space, so there is no "center" in the sense of an arithmetic mean. To enable K-means to cluster permutation instances, this paper transforms each permutation π into a position-encoded vector v, where v(i) represents the position of node i in permutation π (*i.e.*, v = [pos(1), pos(2), …, pos(n)], where n is the number of nodes). K-means performs cluster assignment using Euclidean distance in this position-encoded vector space, dividing the population into C clusters; after clustering is complete, the permutation individual with the optimal fitness (distance traveled) within each cluster is selected as the representative of that cluster (*i.e.*, the cluster center), to be used for parent selection in subsequent idea generation; 3) Idea Generation (Inter-cluster Combination): With probability p2, randomly select one individual from each of two clusters as parent individuals; with probability 1−p2, select the optimal individual and one random individual from a single cluster as parent individuals; use ordered crossover (OX) on the two parent permutations to generate offspring, thereby combining superior gene segments within and between clusters; 4) Mutation: To maintain population diversity, mutations are applied to the offspring

with probability p_m. The mutation operators include inversion, permutation, and insertion; 5) Selection and Update: A greedy strategy is adopted; if an offspring is superior to the worst individual in the population, it replaces that individual.

OX is a classic crossover operator for solving the TSP: two crossover positions are randomly selected from parent A, and the gene segment between them is copied to the corresponding positions in the offspring; then, genes from parent B that do not appear in this segment are inserted into the remaining positions in the offspring in their original relative order. OX preserves the relative order information of the parents and does not produce invalid duplicate individuals, making it particularly suitable for permutation encoding. It should be noted that the performance improvement of BSO-P relative to the random-key BSO stems from the combined design of permutation encoding, OX crossover, and various mutation operators, rather than a change in a single encoding method; specifically, permutation encoding allows the search to act directly on the visit order, OX preserves relative order information, and mutation operators maintain population diversity.

4. Case Studies and Results Analysis

4.1. Case Study Data

Take the camouflaged target reconnaissance task in a certain operational area as an example. This task includes 22 reconnaissance targets and 1 takeoff/landing point (origin O), for a total of 23 nodes. The coordinates of the nodes are shown in Table 1 (the coordinates of the reconnaissance targets in Table 1 are synthetic data—that is, simulated data constructed based on the typical distribution characteristics of the operational area to validate the algorithm). A local Cartesian coordinate system is established with the takeoff/landing point O as the origin. The spatial distribution is shown in Figure 2. As can be seen from Table 1, the reconnaissance targets are distributed in an irregular, approximately annular pattern within an area of approximately 7600 m × 6700 m; the units for both the horizontal and vertical axes in Figure 2 are meters.

4.2. Experimental Setup

All algorithms were implemented and run in the same software environment. To ensure fairness in the comparison, a unified computational budget independent of the specific algorithm was adopted, meaning that the maximum number of function evaluation iterations for each test case was set to $NFEmax = 2 \times 10^4$; each algorithm was run independently 20 times using random seeds ranging from 1 to 20. The main parameter settings for each algorithm are shown in Table 2. To estimate the optimal solution for this test case, this paper additionally employed a method combining 2-opt, Or-opt, and Iterative Local Search (ILS) for 500 random restarts. None of these attempts found a path shorter than 27865.49 m; therefore, this value is adopted as the known optimal solution for this test case.

Table 2. Main parameter settings of the algorithms.

Algorithm	Key Parameter Settings
GA	Population size: 40; sequential crossover (OX), crossover probability: 1.0; mutation probability: 0.3; tournament selection (size: 3); retain 2 elites
SA	Initial temperature 1000; cooling coefficient 0.9995; neighborhood operators: reverse, swap, and insert mutation
ACO	Number of ants: 20; pheromone evaporation coefficient: $\rho = 0.1$; $q_0 = 0.9$; $\alpha = 1.0$; $\beta = 2.0$; local pheromone update coefficient: $\xi = 0.1$
ABC	Number of nectar sources: 40; abandonment threshold: limit = 50
BSO	Population size: 40; number of clusters: 5; random key encoding; probability of combining two clusters: 0.2; Gaussian perturbation coefficient: 0.3
BSO-P	Population size: 40; number of clusters: 5; permutation encoding; probability of combining two clusters: 0.2; mutation probability: 0.6 (reverse, swap, insert)
General Settings	Maximum evaluation count: 2×10^4; random seeds: 1 - 20; 20 independent runs

Note: The success rate threshold is set at ±0.5 m from the known optimal value.

4.3. Comparison and Analysis of Results

Table 3 shows the statistical results from 20 independent runs of the six algorithms, each conducted with the same evaluation budget.

Table 3. Comparison of the results obtained by the algorithms (23-node example, 20 independent runs).

Algorithm	Optimal Value/m	Average Value/m	Standard Deviation/m	Worst-Case Value/m	Success Rate	Average NFE to Reach Optimal Value	Average Runtime/ms
BSO-P	27865.49	27898.12	24.35	27965.80	85%	4680	1245.3
BSO	27995.93	28120.45	85.62	28350.20	25%	8750	1680.7
GA	28450.11	28760.32	150.45	29120.55	0%	-	980.5
SA	28210.65	28550.78	110.20	28890.10	0%	-	2150.4
ACO	28150.32	28320.55	75.30	28560.45	5%	14500	4520.8
ABC	28650.88	28990.21	180.60	29450.10	0%	-	1350.2

As shown in Table 3: 1) The standard BSO using random key encoding failed to converge to the optimal known solution in all 20 runs; the optimal value was only 27995.93 m, which is approximately 0.47% longer than the known optimal path of BSO-P; Both its average value (28120.45 m) and worst-case value (28350.20 m) were relatively high, indicating that random key encoding has limited search efficiency in the permutation solution space and is prone to getting stuck in local optima. This corroborates the analysis in Section S3.2 and serves as the motivation for the permutation encoding improvement in this paper.

2) The proposed BSO-P achieved a success rate of 85% (17 out of 20 runs) over 20 runs, the highest among all algorithms. In this 23-node example, BSO-P reached an optimal value of 27865.49 m, successfully approaching the theoretical optimum. Compared with other algorithms that failed to reach the optimum, BSO-P demonstrates a significant advantage. It should be noted that, since BSO-P must perform K-means clustering in each generation, its measured runtime (1245.3 ms) is shorter than that of ACO (4520.8 ms) but longer than that of GA (980.5 ms) and SA (2150.4 ms). The above conclusions apply only to the number of function evaluations—a metric independent of implementation—and are limited to the scale of this example; they do not imply faster computation or better performance for larger-scale problems.

3) Neither GA nor ABC reached the optimal solution in any of the 20 runs (both had a success rate of 0%), and both had high mean and worst-case values (GA's mean was 28760.32 m, and ABC's mean was 28990.21 m), indicating that these two types of algorithms are highly prone to premature convergence in this discrete path planning problem. Although SA had a 0% success rate, its mean value (28550.78 m) was slightly better than those of GA and ABC. Considering success rate, mean value, and standard deviation collectively, BSO-P demonstrated significantly greater reliability than all other algorithms.

4) ACO exhibited the most unstable performance across 20 runs; although it occasionally achieved relatively good solutions (5% success rate), the average number of evaluations required to reach the optimal solution was as high as 14,500; In terms of average runtime, GA (980.5 ms) and ABC (1350.2 ms) were faster than BSO-P, primarily because BSO-P requires K-means clustering in each generation, resulting in a higher time overhead per evaluation; However, when measured by the number of evaluations—a metric independent of specific implementations—BSO-P required far fewer evaluations (4680) to reach the same optimal solution than ACO (14,500) and the unsuccessful BSO (8750), demonstrating a significant advantage.

4.4. Convergence Analysis

Figure 2 shows the convergence curves of the average optimal path length for each algorithm over 20 runs as the number of function evaluations increases. It can be seen that SA and BSO-P decline most rapidly in the early stages of the search, followed by GA, while the standard BSO never came close to the known optimal solution within the entire evaluation budget and showed an extremely gradual decline; Although ABC shows a noticeable decrease, its convergence rate is slower than that of BSO-P (it does not approach the optimal solution until approximately 1.06×10^4 evaluations), while ACO stalls at a local optimum within a very short time; Overall, BSO-P, leveraging its search advantage of directly acting on permutation sequences, not only locates the global optimal solution the fastest but is also minimally affected by individual runtime fluctuations. It significantly outperforms other algorithms in both the efficiency of reaching the optimal known so-

lution and the final optimization accuracy.

4.5. Verification of the Optimal Reconnaissance Path and Flight Time Feasibility

The known optimal reconnaissance path obtained by BSO-P is (node numbers correspond to those in Table 1): O(1)→2→23→22→21→20→19→18→17→16→15→14→13→12→10→11→9→8→7→6→5→4→3→O(1). The total flight distance of this path is 27865.49 m (approximately 27.87 km), as shown in Figure 3. As can be seen in Figure 3, the optimal path follows an approximately closed loop around the perimeter of the reconnaissance targets, avoiding back-and-forth traversals between them, thereby shortening the total flight distance.

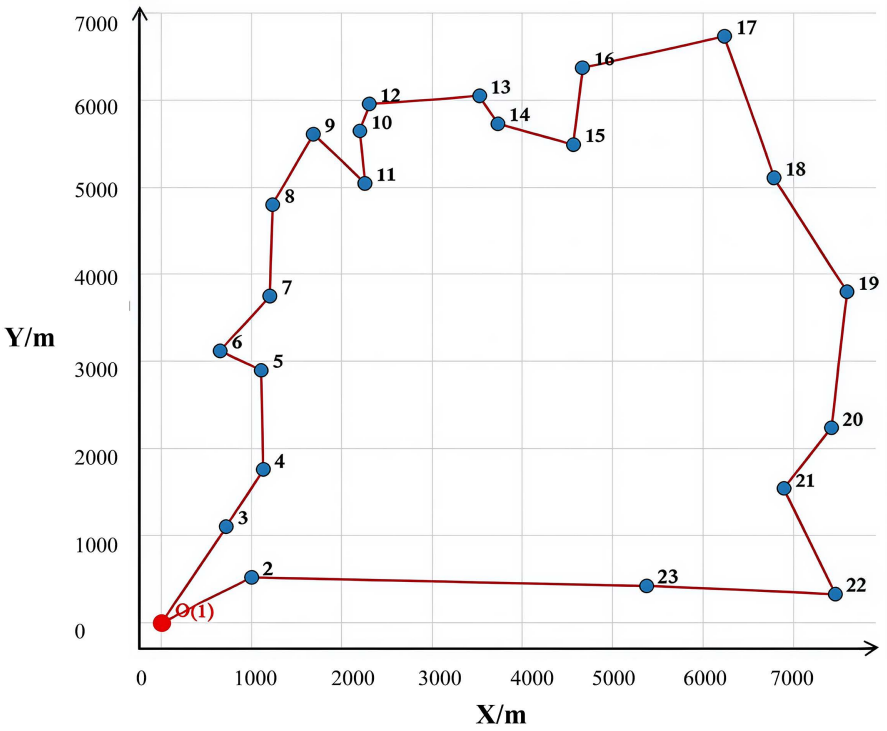


Figure 3. Optimal reconnaissance path of the UAV.

Flight Duration Feasibility Check: The cruising speed of a fixed-wing UAV is generally no less than 60 km/h. Based on a conservative cruise speed estimate of 60 km/h, the flight time required to complete a 27.87 km route is approximately 0.46 h (about 27.9 min); even when accounting for approximately 2 min of reconnaissance time per target (totaling about 44 min for 22 targets), the total mission duration remains approximately 72 min, which is far less than the approximately 4 h endurance of this UAV model. Therefore, the optimal path described above is feasible under actual endurance conditions.

4.6. Discussion and Limitations

It should be noted that the examples in this paper simplify the problem to a sym-

metric TSP and do not consider the following factors: 1) In actual camouflaged target reconnaissance, the UAV's flight path is also affected by wind direction and speed, no-fly zones, terrain obstructions and threats, as well as flight altitude constraints; in such cases, the problem is closer to a constrained flight path planning problem; 2) This paper focuses solely on minimizing the total flight distance as the sole objective, without considering constraints such as reconnaissance time windows, target priorities, battery capacity, and flight duration; 3) The comparative experiments primarily focused on the proposed algorithm versus commonly used intelligent algorithms, without conducting a systematic comparison with exact algorithms or other BSO variants; 4) The average runtime depends on the specific implementation and programming language and should be treated only as a reference; it should not be used as the sole basis for evaluation. Future research could incorporate wind field and threat constraints, time windows, and multi-objective optimization into the model, and further validate the algorithm's engineering applicability by combining it with actual flight trajectory data.

5. Conclusion

This paper investigates the path optimization problem for multi-target camouflaged target reconnaissance using unmanned aerial vehicles (UAVs). The main conclusions are as follows: 1) The problem is abstracted as a Traveling Salesman Problem (TSP), and a 0-1 integer programming model incorporating constraints for eliminating MTZ subloops is established, providing a rigorous mathematical description for UAV reconnaissance path optimization; 2) To address the difficulty of directly applying standard BSO random key encoding to the TSP permutation solution space, an improved algorithm, BSO-P, was proposed based on permutation encoding and sequential crossover between clusters; 3) For a case study involving 23 nodes along a designated operational area, the shortest reconnaissance path was determined to be 27865.49 m (approximately 27.87 km), and the feasibility of the flight duration was verified; 4) Comparative experiments show that BSO-P achieved an 85% success rate over 20 independent runs, demonstrating the best performance; the average number of evaluation iterations required to reach the optimal known solution (4680) was significantly lower than that of the equally stable ACO (14,500) and standard BSO (8750); As shown in **Figure 3**, although SA exhibited the fastest initial descent, its average curve remained elevated in the later stages due to a few runs failing to converge. Meanwhile, the standard random-key BSO never approached the optimal solution throughout the entire budget, while ACO and ABC converged slowly or became trapped in local optima. In summary, the proposed BSO-P method performs well in terms of solution accuracy, stability, and convergence efficiency, and can serve as a reference for path planning when reconnaissance units carry out multi-target camouflaged target reconnaissance missions.

Funding Project

This work was supported by the National Key Laboratory on Electromagnetic En-

vironmental Effects and Electro-optical Engineering project "Key Technologies of Digital Twin for Target Electromagnetic Characteristics and Multi-Spectral Parameter Prediction" [grant no. 61422062407].

Author Contributions

Yu Hai-bao: Conceptualization, writing—original draft. Xu Wei-dong: Methodology, formal analysis. Cui Jing: Investigation, data curation. Yang Xin: Visualization, writing—review & editing. Peng Hao: Supervision, project administration. Xie Yu-mou: Resources, writing—review & editing. Chen Wang-kai: Investigation. Zhen Long-hua: Data curation, validation. All authors have read and agreed to the published version of the manuscript.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Si, S. and Sun, X. (2022) Mathematical Modeling Algorithm and Its Application with Python. 3rd Edition, National Defense Industry Press. (In Chinese) http://opac.nlc.cn/F/MCGGD6B8MAQVUKPMC4NCYUF5PIG-GRJMTK5NUN8EVR5KYT1LPH6-04892?func=full-set-set&set_number=205226&set_entry=000002&format=999

[2] Sun, J., Liu, S., Wu, X., *et al*. (2022) A Spark-Based Parallel Simulated Annealing Algorithm for Solving the TSP. *Electronic Measurement Technology*, **45**, 53-58. (In Chinese)

[3] Shi, Y. (2011) Brain Storm Optimization Algorithm. In: Tan, Y., Shi, Y., Chai, Y. and Wang, G., Eds., *Lecture Notes in Computer Science*, Springer, 303-309. https://doi.org/10.1007/978-3-642-21515-5_36

[4] Yang, J. and Shi, Y. (2022) Automatic Synthesizing Multi-Robot Cooperation Strategies Based on Brain Storm Robotics. *Applied Soft Computing*, **120**, Article 108672. https://doi.org/10.1016/j.asoc.2022.108672

[5] Cai, Z., Gao, S., Yang, X., Yang, G., Cheng, S. and Shi, Y. (2022) Alternate Search Pattern-Based Brain Storm Optimization. *Knowledge-Based Systems*, **238**, Article 107896. https://doi.org/10.1016/j.knosys.2021.107896

[6] Jiang, Y., Chen, X., Zheng, F., Niyato, D. and You, X. (2021) Brain Storm Optimization-Based Edge Caching in Fog Radio Access Networks. *IEEE Transactions on Vehicular Technology*, **70**, 1807-1820. https://doi.org/10.1109/tvt.2021.3052971

[7] Cheng, S., Liu, Y., Wang, X., *et al*. (2024) An Improved Brainstorming Optimization Algorithm for Solving Multimodal Multi-Objective Problems. *Journal of Huazhong University of Science and Technology* (*Natural Science Edition*), **52**, 24-31. (In Chinese)

[8] Cheng, S., Wang, X., Liu, Y., *et al*. (2023) A Learning-Based Brainstorming Optimization Algorithm for Nonlinear Systems of Equations. *Computer Engineering*, **49**, 47-54. (In Chinese)

[9] Li, X., Yu, M., Jiang, Q., *et al*. (2024) A Multimodal Multi-Objective Brainstorming Optimization Algorithm Based on Partitioned Search and Reinforcement Learning. *Research on Computer Applications*, **41**, 2374-2383. (In Chinese)