ITC 2/54

Information Technology and Control
Vol. 54 / No. 2/ 2025
pp. 413-438
DOI 10.5755/j01.itc.54.2.40640

Multi-strategy Hybrid Improved Intelligent Algorithm for Solving UAV-MTSP

Received 2025/02/xx | Accepted after revision 2025/04/29

HOW TO CITE: Wang, Z., Wang, D., Yu, J. (2025). Multi-strategy Hybrid Improved Intelligent Algorithm for Solving UAV-MTSP. *Information Technology and Control,* 54(2), 413-438. https://doi.org/10.5755/j01.itc.54.2.40640

Multi-strategy Hybrid Improved Intelligent Algorithm for Solving UAV-MTSP

Zixin Wang*

School of Communication Engineering, Jilin University, Changchun, 130012, China

Danqing Wang*

School of Economics and Management, Communication University of China, Beijing, 100024, China

Jiguang Yu

Department of Mathematics, University College London, London, WC1E 6BT, United Kingdom

* Zixin Wang and Danqing Wang are first co-authors

Corresponding author: zcahyuc@ucl.ac.uk (Jiguang Yu)

Unmanned aerial vehicles (UAVs) have been increasingly used in fire monitoring and rescue operations, offering flexibility and efficiency. However, determining the shortest path for all UAVs to visit all regions is a crucial issue, known as the Multiple Traveling Salesman Problem (MTSP), which aims to save time and energy. This paper proposes a novel hybrid heuristic algorithm, MCPWOA, to solve MTSP with a focus on UAV path planning applications. The algorithm integrates the Whale Optimization Algorithm (WOA), Crested Porcupine Optimizer (CPO), Chaotic Mapping Strategy (CMS), Arcsine Control Strategy (ACS) and Reverse Learning Strategy (RLS) to diversify the initial population and achieve rapid exploration. The algorithm's performance is evaluated using the CEC2022 benchmark function set and TSPLIB dataset for function minimization and UAV-MTSP experimental solution finding. Results indicate that MCPWOA outperforms existing WOA, CPO, and other advanced algorithms on most tests, showing higher convergence accuracy. Moreover, MCPWOA's effectiveness is demonstrated in actual UAV fire monitoring and rescue path planning, enhancing fire response efficiency through optimized UAV configuration and task allocation.

KEYWORDS: UAV, Path planning, Whale Optimization Algorithm, Crested Porcupine Optimizer, Multiple Traveling Salesman Problem, Fire monitoring and rescue.

1.Introduction

1.1 Background

The advancements in drone technology have rendered them critical for fire monitoring and rescue operations, due to their flexibility and efficiency. Efficiently planning drone flight paths to cover critical fire points is challenging, due to complex terrains and variable fire dynamics. Heuristic algorithms provide a powerful solution to tackle the Multiple Traveling Salesman Problem (MTSP), especially for large-scale problems. These algorithms are well-suited to handle the complexity and scale of MTSP, optimizing routes for multiple drones to minimize total path length and enhance monitoring efficiency. Their ability to provide near-optimal solutions quickly makes them ideal for real-time applications in dynamic environments. As a result, research on applying heuristic algorithms to drone path planning for fire monitoring is an active and significant field.

1.2 Motivation and Related Works

The Traveling Salesman Problem (TSP) is a classic problem in the field of combinatorial optimization. Since the 20th century, its solutions have evolved from exact algorithms to heuristic approaches. In the 20th century, solving the TSP primarily relied on exact algorithms such as the branch and bound method [15], dynamic programming [12], and branch and cut method [10]. However, as the problem scale expanded, the computational complexity and time costs of these methods increased significantly, making them impractical for real-world applications. Consequently, heuristic algorithms have gradually become the mainstream approach to solving the TSP in the 21st century.

In 2019, Liu and Wang [28] proposed a TSP solution based on genetic algorithms, which enhanced global search capabilities and convergence speed through optimized selection and mutation operations. In the same year, Chen et al. [14] developed a time-approximation scheme capable of providing approximate solutions within exponential time. In 2020, Manna et al. [30] employed a clustering method to address the TSP, dividing vertices into multiple clusters and using genetic algorithms to find Hamiltonian paths. Simultaneously, Guo et al. [18] introduced the MEATSP membrane evolutionary algorithm, which mimics biological cell behavior to solve large-scale TSP problems. Additionally, in 2020, Chen et al. [14] devised a sublinear algorithm to estimate the cost of graphical TSP, thereby enhancing computational efficiency.

Entering 2021, Zhong [43] proposed an improved 3-Opt algorithm for solving the TSP, which improved the approximation ratio. In 2023, Akhmetov and Pak [6] conducted a comparative study evaluating various TSP algorithms on standard datasets, suggesting hybrid techniques that combine existing algorithms. Furthermore, Alkafaween et al. [8] introduced an iterative approximation method, optimizing paths through local constant permutation. Behnezhad et al. [13] proposed a path-covering-based sublinear algorithm, enhancing the estimation accuracy for graphical TSP.

Due to the more complex and diverse scenarios in real applications, the Multiple Traveling Salesman Problem (MTSP) was introduced to solve the real-world problems. The Multiple Traveling Salesman Problem (MTSP) is an extended version of the traveling salesman problem (TSP) involving multiple traveling salesmen whose routes need to be optimized to reach a specific goal. In recent years, researchers have proposed a variety of algorithms to solve MTSP. In 2019, Gulcu and Ornek [17] proposed an algorithm based on Particle Swarm Optimization (PSO) for solving MTSP. The performance of the algorithm is improved by introducing the 2-opt algorithm and the path relinking operation. In 2020, Lu et al. [29] assigned ants to task-oriented teams, so that each ant used Max-Min strategy to jointly optimize the solution to enhance the performance of the ant colony optimization algorithm. Wang et al. [39] proposed an improved ant colony optimization algorithm with a pheromone model to solve the MTSP problem with capacity and time window constraints. In 2021, Karabulut et al. [23] proposed an Evolutionary strategy (ES) approach for solving MTSP problems with minimization and maximization objectives. The quality of the solution is optimized by adaptive destruction and reconstruction heuristics and a 3-opt local search. Ren et al. [34] optimized the path of the multi-agent MTSP by improving the penalty function of the simulated annealing algorithm. In 2023, Hu and Yang [21] proposed a local optimization algorithm based on eliminating inclusion and crossing relationships between subpaths for balanced workloads in MTSP problems. Nayak and Rathinam [32] explored the application of learning models in Dubins MTSP and proposed a learning-based heuristic search method.

Although accurate algorithms have advantages in solving small-scale problems, in the face of large-scale and complex problems in reality, heuristic algorithms are gradually being widely used because of their efficiency and flexibility. In recent years, UAV technology has developed rapidly in recent years, becoming an indispensable tool in emergency rescue. Therefore, many scholars have tried to apply the heuristic algorithm to solve the MTSP problem to the actual scene of UAV emergency rescue, and have made many achievements. Sanchz-garcia et al. [35] proposed a distributed exploration algorithm based on Particle Swarm Optimization (PSO), which was applied to UAV networks in disaster scenarios to autonomously deploy and provide communication services. The algorithm significantly improved the efficiency of victim discovery and connecting events. Tang [36] solved the problem of random interference and low efficiency in UAV path planning through improved Genetic Algorithm (IGA), and achieved high coverage and high solution accuracy. Huang et al. [22] proposed a new UAV path planning framework that combines path and motion planning to effectively improve the reliability of data transmission and flight time. Arafat and Moh [11] proposed a swarm intelligence-based localization and clustering algorithm for UAV networks in emergency communications to improve localization accuracy and energy efficiency. Wang et al. [40] optimized the multi-UAV task area allocation problem by combining genetic algorithm and simulated annealing algorithm, which effectively improved the balance of task allocation and the global search ability of the algorithm. Gan and Liu [16] proposed a rapid forest fire response method based on K-means and MTSP algorithm, which improved the fire response efficiency by optimizing the UAV configuration and task allocation. Li et al. [26] proposed a multi-UAV patrol path planning method based on evolutionary multi-objective optimization, which significantly reduces the time and energy costs. Almasoud [9] studied a UAV-based intelligent transportation system for emergency reporting in coverage of wireless network void areas, which greatly reduces the emergency information delivery time. Wang et al. [38] proposed a UAV-assisted dynamic hypergraph coloring method for emergency communication, which reduces interference and improves information diffusion speed by optimizing spectrum sharing. Lin et al. [27] proposed an extended model of the Multi-Armed Slot Machine (MAB) problem to optimize the path planning of UAVs in post-disaster emergency communication, which significantly increased the number of served users. Kyrkou and Theocharides [24] proposed a deep learning-based emergency surveillance system, EmergencyNet, which improves the efficiency of emergency response by optimizing the UAV image classification algorithm. Hong et al. [20] proposed an adaptive hybrid algorithm for optimizing UAV search and rescue missions, which significantly improves the target discovery probability and search efficiency. Qu et al. [33] studied the rapid deployment method of UAVs based on bandwidth resources, which effectively reduced the deployment delay and the number of UAVs. Akter et al. [7] proposed a task offloading and resource allocation strategy for UAV emergency response operation in a multi-access Edge Computing (MEC) environment, which improved the task execution efficiency. Wan et al. [37] proposed an improved multi-objective swarm intelligence algorithm for 3D path planning of UAV disaster emergency response, which optimized the flight path and obstacle avoidance performance. These algorithms not only improve the application effect of UAVs in emergency rescue, but also provide a solid foundation for future emergency response technology, showing the great potential of heuristic algorithms in practical applications.

1.3 Our Contributions

As mentioned above, multi-strategy augmented hybrid heuristics have shown great potential in various optimization domains; However, few studies adopt hybrid optimization strategies for the WOA algorithm and CPO algorithm to optimize the path problem of UAV traversing the fire point. In this paper, the algorithm is studied in depth and its improved methods are reviewed in detail, and the potential application of the algorithm in fire monitoring and rescue is discussed. The main contributions can be summarized as follows:

- Initially, an overview of the multi-strategy hybrid algorithm was provided, covering both the mathematical model and the optimization approach. The original WOA is primarily effective for continuous, unconstrained optimization tasks. To broaden its scope and enhance its performance, various strategies including chaotic mapping strategy (CMS), arcsine control strategy (ACS) and Reverse learning strategy (RLS) have been integrated. By combining WOA with the Crested Porcupine Optimizer (CPO) and these additional strategies, the algorithm is equipped to tackle a diverse array of optimization challenges, delivering high-quality solutions.

- To demonstrate the hybrid algorithm's applicability, its performance was studied on the Multiple Traveling Salesman Problem (MTSP) of UAV path optimization in a fire monitoring scenario. Simulation results show that the hybrid algorithm can achieve better performance compared to other metaheuristic algorithms.
- Several potential applications of hybrid algorithms for complex optimization problems are outlined, particularly for path planning of UAVs in forest fire rescue operations. The effectiveness of the proposed algorithm in addressing large-scale dynamic optimization challenges is demonstrated through the use of cluster transformation and asynchronous composition methods.

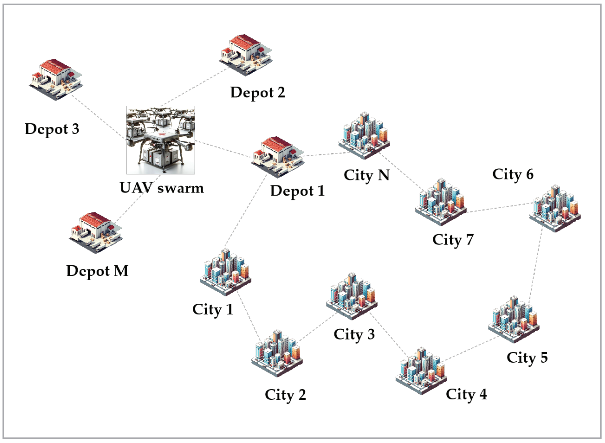
2. MCPWOA: Foundation and Framework

In this section, we introduce the UAV-MTSP and basic framework of MCPWOA, which integrate the Crested Porcupine Optimizer (CPO), Whale Optimization Algorithm (WOA) and other enhancement strategies.

2.1 Introduction to the Multiple traveling Salesman Problem (MTSP)

The Multiple Traveling Salesman Problem (MTSP) is an extension of the Traveling Salesman Problem (TSP), where multiple salesmen are required to visit a set of cities, with each city visited at least once by any one of them and no more than once by the same salesman, as shown in Figure 1. The objective is to determine the most efficient route for each salesman that minimizes the total travel distance. This problem has various real-world applications, including logistics, UAV routing, and traffic management, and often requires the use of integer programming and different optimization techniques to solve effectively.

Figure 1
Example of UAV-MTSP.



In this paper, the path optimization problem of UAVs traversing a fire area is considered. In a disaster emergency rescue scenario, multiple UAVs must start from the depots, traverse the fire area with varying fire conditions, and then return to the original depots. Above all, fire monitoring and rescue task for USV is modeled as a Multi-Depot Multiple Traveling Salesman Problem (MD-MTSP).

2.2 Introduction to Whale Optimization Algorithm (WOA)

The Whale Optimization Algorithm (WOA), introduced by Mirjalili and Lewis in 2016 [31], is an optimization method based on biomimicry. It draws inspiration from the hunting techniques of humpback whales, using "bubble nets" to capture prey. WOA is a global optimization tool, it randomly initializes the dimensions with the following formula:

$$\overrightarrow{X_i} = \vec{L} + \vec{r} \times \left(\vec{U} - \vec{L}\right) \mid i = 1, 2..., N', \tag{1}$$

where N' denotes the number of individuals (i.e., the population size), $\overrightarrow{X_i}$ is the i th candidate solution within the search space, $\vec{L}$ and $\vec{U}$ represent the lower and upper bounds of the search range, respectively, and $\vec{r}$ is a vector initialized within the range [0, 1].

The following equation explains the encircling prey phase in WOA from a mathematical point of view:

$$\overrightarrow{x_i^{t+1}} = \overrightarrow{x_*^t} - \vec{A} \cdot \vec{D} \tag{2}$$

$$\vec{D} = \left| \vec{C} \cdot \overrightarrow{x_*^t} - \overrightarrow{x_i^t} \right| \tag{3}$$

$$\vec{A} = 2\vec{a} \cdot \vec{r} - \vec{a} \tag{4}$$

$$\vec{C} = 2 \cdot \vec{r}, \tag{5}$$

where $\overrightarrow{x_*^t}$ is the current optimal solution, t is the current iteration, $\overrightarrow{x_i^t}$ is the position of the i th individual among population at iteration t, a is a vector which gradually decreases from 2 to 0.

Humpback whales follow a spiral trajectory to approach their prey using the formula:

$$\overrightarrow{D'} = \left| \overrightarrow{x_*^t} - \overrightarrow{x_i^t} \right| \tag{6}$$

$$\overrightarrow{x_i^{t+1}} = \overrightarrow{D'} \cdot e^{bl} \cdot cos(2\pi l) + \overrightarrow{x_*^t}, \tag{7}$$

where b is a constant that determines the spiral shape and l is a random value within the interval [-1, 1].

The algorithm adopts random search strategy. When $\left|\vec{A}\right| > 1$, the formula is:

$$\overrightarrow{x_i^{t+1}} = \overrightarrow{x_r^t} - \vec{A} \cdot \vec{D}, \tag{8}$$

where $\overrightarrow{x_r^t}$ is a randomly chosen whale location.

2.3 Introduction to Crested Porcupine Optimizer (CPO)

The Crested Porcupine Optimizer (CPO) is a metaheuristic algorithm inspired by the crowned porcupine's defense behaviors [2]. It uses four strategies - visual signals, auditory warnings, odor emissions, and physical attacks - to emulate the animal's defense mechanisms. CPO combines exploration and exploitation to tackle large-scale, complex optimization challenges. CPO and WOA use the same population initialization strategy, it introduces the cyclic population reduction technique to preserve the population diversity in addition to accelerating the convergence speed. The equation governing the cyclic reduction of population size is given by:

$$N = N_{min} + \left(N' - N_{min}\right) \times \left(1 - \left(\frac{t\% \frac{T_{max}}{T}}{\frac{T_{max}}{T}}\right)\right), \tag{9}$$

where T represents the variable that dictates the number of cycles, $T_{\max}$ is the maximum number of iterations, % denotes the modulo operator, and $N_{\min}$ is the minimum population size in the newly generated group.

The defense strategy in the first stage is expressed as follows:

$$\overrightarrow{x_i^{t+1}} = \overrightarrow{x_i^t} + \tau_1 \times \left| 2 \times \tau_2 \times \overrightarrow{x_*^t} - \overrightarrow{y_i^t} \right| \tag{10}$$

$$\overrightarrow{y_i^t} = \frac{\overrightarrow{x_i^t} + \overrightarrow{x_r^t}}{2}, \tag{11}$$

where τ_1 is a random number following a normal distribution, and τ_2 is a randomly generated value within the range [0, 1]

The defense strategy in the second stage is expressed as follows:

$$\overrightarrow{x_i^{t+1}} = \left(1 - \overrightarrow{U_1}\right) \times \overrightarrow{x_i^t} + \overrightarrow{U_1} \times \left(\vec{y} + \tau_3 \times \left(\overrightarrow{x_{r1}^t} - \overrightarrow{x_{r2}^t}\right)\right), \tag{12}$$

where r_1 and r_2 are two randomly selected integers within the range [1, N] and τ_3 consists of random values generated between 0 and 1. $\overrightarrow{U_1}$ is a binary vector including 0 and 1 generated randomly

The defense strategy of the third stage is expressed as follows:

$$\overrightarrow{x_i^{t+1}} = \left(1 - \overrightarrow{U_1}\right) \times \overrightarrow{x_i^t} + \overrightarrow{U_1} \times \left(\overrightarrow{x_{r1}^t} + S_i^t \times \left(\overrightarrow{x_{r2}^t} - \overrightarrow{x_{r3}^t}\right) - \tau_3 \times \vec{\delta} \times \gamma_t \times S_i^t\right) \tag{13}$$

$$\vec{\delta} = \begin{cases} +1, if\ \vec{r} \leq 0.5 \\ -1, Else \end{cases} \tag{14}$$

$$\gamma_t = 2 \times r \times \left(1 - \frac{t}{T_{max}}\right)^{\frac{t}{T_{max}}} \tag{15}$$

$$S_i^t = exp\left(\frac{f\left(\overrightarrow{x_i^t}\right)}{\sum_{K=1}^{N} f\left(\overrightarrow{x_k^t}\right) + \varepsilon}\right), \tag{16}$$

where r_3 is a random number within the range [1, N], δ is a parameter controlling the search direction, represents the position of the i th individual at iteration t, and γ_t is the defined defense factor. τ_3 is a random value within the interval [0, 1] and S_i^t is the defined odor diffusion factor. $f\left(\overrightarrow{x_i^t}\right)$ denote the objective function value of the i th individual at iteration t, ε is a small constant to prevent division by zero, rand is a vector of randomly generated values within the range [0, 1], N represents the population size, t is the current iteration number, $T_{\max}$ is the maximum number of iterations, and another rand consists of values randomly generated between 1 and 0. The $\overrightarrow{U_1}$ is used to simulate three different scenarios that may occur within this strategy.

The defense strategy of the fourth stage is expressed as follows:

$$\overrightarrow{x_i^{t+1}} = \overrightarrow{x_*^t} + \left(\alpha\left(1-\tau_4\right)+\tau_4\right)\times \left(\delta\times\overrightarrow{x_*^t}-\overrightarrow{x_i^t}\right)-\tau_5\times\delta\times\gamma_t\times\overrightarrow{F_i^t} \tag{17}$$

$$\overrightarrow{F_i^t} = \overrightarrow{\tau_6}\times\frac{m_i\times\left(\overrightarrow{x_r^t}-\overrightarrow{x_i^t}\right)}{\Delta t} \tag{18}$$

$$m_i = f\left(\overrightarrow{x_i^t}\right)/\,exp\left(\sum_{k=1}^{N} f\left(\overrightarrow{x_k^t}\right)+\varepsilon\right), \tag{19}$$

where τ_4, τ_5 are both random values within the interval [0, 1], Δt is the current number of iterations, represented by a vector that includes randomly generated values between 0 and 1. τ_6 is a vector including random values generated between 0 and 1.

2.4 Enhanced WOA Using Factors and Strategy in CPO (CPWOA)

Whale optimization algorithm has a simple structure and fast computation speed, but it may meet problems like falling into local minima and imbalance between exploration and exploitation when solving continuous and discrete optimization problems [4]. To enhance the performance of the whale optimization algorithm, researchers use many strategies: Wu et al. [42] used the arcsine control strategy to find a balance between exploration and exploitation. Abdel-Basset et al. [1] used a local search strategy and the Lévy flight walks to give a better tradeoff between the diversification and the intensification.

Furthermore, as a new and powerful optimizer, the Crested Porcupine Optimizer has many useful factors to enhance its exploration and exploitation capabilities. In consideration of the fact that WOA simulates the process by which whales hunt their prey and CPO simulates the process by which crested porcupine defense hunters, it is natural to blend WOA and CPO by giving the unique defense methods owned by crested porcupines to the prey hunted by whales. This fusion strategy can increase the game between whales and prey, so that whales will explore more regions instead of staying beside the prey.

The third defense strategy and the fourth defense strategy in CPO simulate the violent reaction when the hunter is too close to the crested population. The existence of the odor diffusion factor S_i^t in the third strategy and the average force F_i^t in the fourth strategy make sure that the optimizer can concentrate exploitation in not only the neighborhood of the global optimal solution but also other solutions in the population, and provide a more comprehensive examination around the global optimal solution so far, which all mean a great capacity to avoid falling into local minima.

As mentioned above, in order to improve the whale optimization algorithm with the help of CPO, firstly the odor diffusion factor S_i^t is added into the encircling prey phase in order to encourage exploitation to cover a larger range rather than focus on the region around the global optimal solution so far, secondly, average force F_i^t is added into bubble-net attacking phase. The fusion strategy can provide WOA a powerful potential to jump out of the local minima, new update equations in whale population after fusion are as follows:

Encircling prey phase:

$$\overrightarrow{x_i^{t+1}} = \left(1-\overrightarrow{U_1}\right)\times\left(\overrightarrow{x_*^t}-A\cdot D\right) +\overrightarrow{U_1}\times\left(\overrightarrow{x_{r_1}^t}+S_i^t\times\left(\overrightarrow{x_{r_2}^t}-\overrightarrow{x_{r_3}^t}\right)-\tau_3\times\vec{\delta}\times\gamma_t\times S_i^t\right) \tag{20}$$

Bubble-net attacking phase:

$$\overrightarrow{x_i^{t+1}} = D'\cdot e^{bl}\cdot cos\left(2\pi l\right)+\overrightarrow{x_*^t}+\left[\alpha\left(1-\tau_4\right)+\tau_4\right] \times\left(\vec{\delta}\times\overrightarrow{x_*^t}-\overrightarrow{x_i^t}\right)-\tau_5\times\vec{\delta}\times\gamma_t\times F_i^t \tag{21}$$

In this paper, the enhanced WOA using factors in CPO is called CPWOA.

2.5 Chaotic Mapping Strategy (CMS)

The quality of the original population significantly influences the algorithm's global searching capacity and convergence speed in metaheuristic algorithms. A good original population and rational distribution can help the algorithm find a better solution at the beginning of iteration and accelerate convergence. In traditional WOA and CPO, they use pseudo-random numbers to initialize the population, which cannot guarantee uniform population distribution and may degrade algorithm performance.

Chaotic mapping theory exhibits good randomness and ergodicity. Numerous chaotic mappings exist, including tent mapping, logistic mapping, Chebyshev mapping, circle mapping, and others. This paper considers six types of mappings and selects the most appropriate one through comparison. Their original distributions in 1000 dimensions are depicted in Figure 2.

By contrast, tent mapping can generate more uniform chaotic sequences and better distribution, which will be helpful to improve population diversity. Tent mapping is a piecewise linear mapping which is widely used in chaotic encryption systems, its definition is as follows,

$$\text{If } x_n \in [0, \alpha) \quad x_{n+1} = x_n / \alpha \tag{22}$$

$$\text{If } x_n \in [\alpha, 1], \quad x_{n+1} = (1 - x_n)/(1 - \alpha) \tag{23}$$

Based on the adequate examination, a chaotic sequence could get a good original distribution when α is close to 0.5. Significantly, a chaotic system will be in a short-period state when α is equal to 0.5, so it is usually set to 0.499.

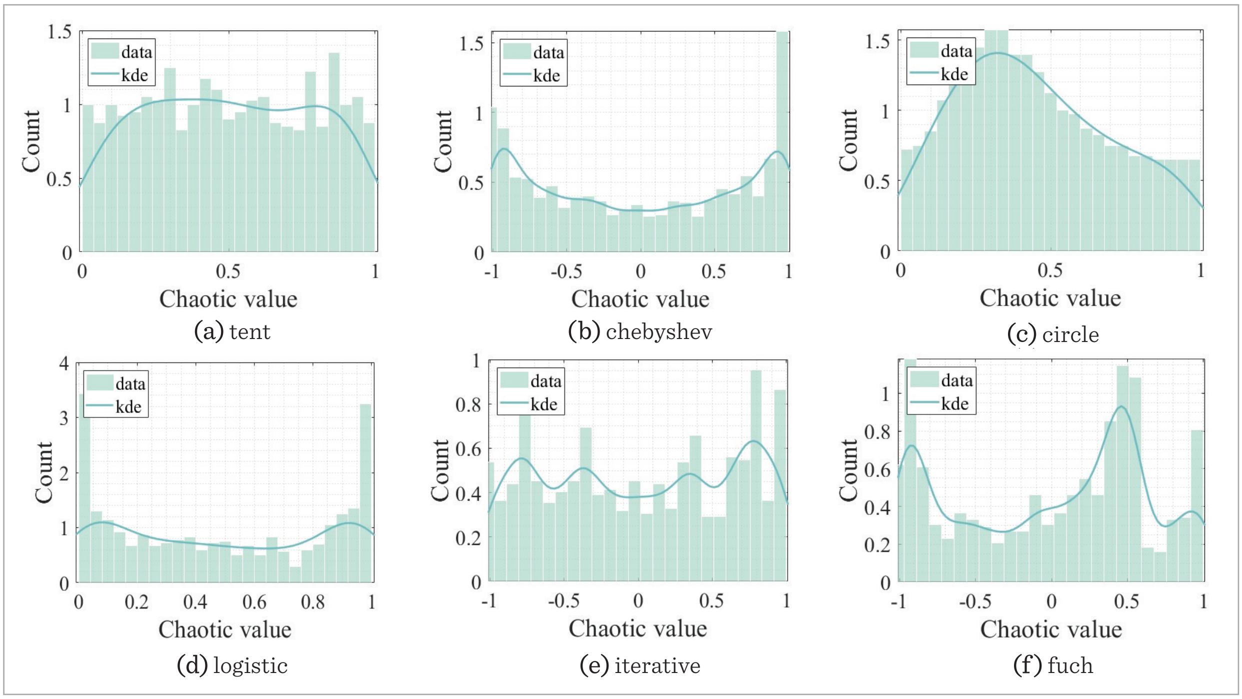
Initializing the population using tent mapping has two steps. The first step is using Equations (22)-(23) to generate a chaotic sequence which has the same dimension as the optimization problem. The second step is replacing the pseudo-random number with the chaotic sequence. The new initialization is as follows:

$$\overrightarrow{X_i} = \vec{L} + \overrightarrow{chaos} \times \left(\vec{U} - \vec{L}\right), \tag{24}$$

where, $\overrightarrow{chaos} = [x_1, x_2, ..., x_n]$, n refers to the dimension of optimization problem.

Figure 2

Original distribution of each mapping.



2.6 Arcsine Control Strategy (ACS)

Besides enhancing exploration and exploitation capacities, the other key to solving a given optimization problem using a metaheuristic mwethod is a good balance between them [41].

Because of that, Wu et al. [42] attempted many nonlinear control strategies to replace traditional constant control strategy or linear control strategy, and found that arcsine function can lead to better performance.

Under this strategy, parameter a in CPWOA and parameter T_f in CPO will decrease along the arcsine function in iterative process. Optimizer will focus more on exploration during early iterations and concentrate on exploiting in the end. The new equation of parameter a is as follows:

$$a = -\frac{2}{arcsin 1} arcsin\left(\frac{t}{T_{max}} - 1\right) \tag{25}$$

$$T_f = -\frac{1}{arcsin 1} arcsin\left(\frac{t}{T_{max}} - 1\right) \tag{26}$$

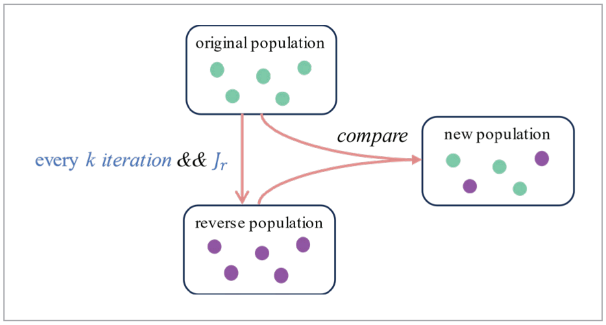
2.7 Reverse Learning Strategy (RLS)

Population evolution is a slow process that takes time. Reverse learning is proposed to boost the learning rate in evolutionary algorithms, simulating a revolution taking place in the population. During the revolution, a reverse population of the same size as the original population is generated under a mapping and interacts with it. Assuming there are N individuals in the original population, N best individuals will survive after interaction. The revolution is a fast process, so reverse learning can not only explore unknown areas in the search space but also accelerate the convergence process.

Both fixed boundary reverse learning and dynamic boundary reverse learning can achieve the mapping from the original population to the reverse population. The former can explore more unknown areas, but may slow down convergence. The latter overcomes the disadvantage that the former has difficulty saving search experience, but will confine the reverse solution to a narrow space. This paper chooses the former to generate a reverse population, which is calculated as follows:

$$\overrightarrow{op_i} = \vec{r} \times \left(\vec{U} + \vec{L}\right) - \overrightarrow{x_i^t} \tag{27}$$

Figure 3
Schematic diagram of reserve learning.



Reverse learning will consume a lot of computing resources when the population has a large size. Given this, jump rate (J_r) is proposed. In the iteration process, execute reverse learning every k iterations, as shown in Figure 3. What is better is that the cyclic population reduction technique can also reduce calculating pressure. Chaotic mapping strategy and reserve learning strategy are usually treated as a joint strategy called chaotic reverse learning because they are based on mappings and helpful in reducing convergence time.

2.8 Introduction to Symbiotic Organisms Search (SOS)

The Symbiotic Organisms Search (SOS) algorithm, based on biological symbiosis theory, uses swarm intelligence to simulate cooperative and competitive behaviors in biological systems to find optimal solutions. It has strong exploitation ability in the mutualism phase, with formulas including:

$$\overrightarrow{x_i^{t+1}} = \overrightarrow{x_i^t} + \vec{r} \times \left(\overrightarrow{x_*^t} - Mutual\_agent\right) \times BF1 \tag{28}$$

$$\overrightarrow{x_j^{t+1}} = \overrightarrow{x_j^t} + \vec{r} \times \left(\overrightarrow{x_*^t} - Mutual\_agent\right) \times BF2 \tag{29}$$

$$Mutual\_agent = \frac{\overrightarrow{x_i^t} + \overrightarrow{x_j^t}}{2}, \tag{30}$$

where denotes the reciprocal intermediate of the i and j individuals in the set of generations, $BF1$ and

*BF*2 are randomly generated interest factors, whose values are 1 or 2, indicating the beneficial degree of the reciprocal relationship between two individuals.

2.9 Multi-strategy Hybrid Algorithm Base on Predation Mechanism (MCPWOA)

In this section, we introduce the three above strategies to improve the convergence precision of CPWOA and original CPO, and the two improved algorithm are integrated into a hybrid search algorithm

Symbiotic Organisms Search (SOS) is renowned for its efficient exploitation capacity during mutualism and commensalism. It utilizes the global optimal solution as a standard, providing a wider search range. Wang et al. [41] introduced the mutualism phase in SOS, combining butterfly optimization algorithm and flower pollination algorithm to achieve strong exploitation capability and fast convergence speed. The symbiotic mechanism aligns with the relationship between butterflies and flowers, rather than whales and their prey. Given this, the Predatory Organisms Search (POS) is proposed by adjusting parameters in SOS to fit the whale-prey relationship.

The predation relationship is characterized by a direct benefit to the predator, who gains energy, nutrients, and other resources from consuming the prey, while the prey suffers harm or death. Enhanced Whale Optimization Algorithm (WOA) and Cuckoo Population Optimization (CPO) are both excellent algorithms for solving optimization problems. To release the full potential of each algorithm and retain their respective advantages, POS is utilized to hybridize the two.

To simulate the predation between whales and prey, the parameter *BF1* in SOS is randomly set to -1 or -2, which refers to the benefit factor of the first member in the ecosystem. A negative value represents that it suffers from harm during the interaction with the second member. Then The parameter *BF2* of the second member is randomly set to 1 or 2 to show it gains benefit, so the first member plays the part of prey when the second playing the part of whales. What is better is that POS could exploit a bigger neighbourhood of the global optimal solution than SOS because *BF1* and *BF2* have opposite symbol.

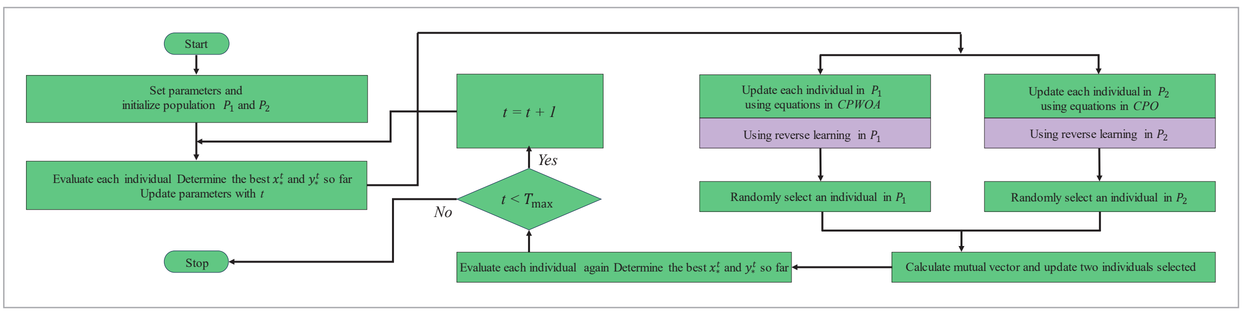
This paper takes four steps to mix WOA and CPO. The first step is to generate two populations P_1 and P_2 to constitute a community. The second step is to update two populations independently, which refers to update P_1 using equations in CPO and update P_2 using equations in enhanced WOA. The third step is to lead interactions between two populations using POS. In this step, every whale selects a prey to hunt randomly and generates mutualism vector *Mutual_agent*. The fourth step is to use *Mutual_agent*. to generate new individuals in each population, so as to increase the exploitation ability of the fusion algorithm and give play to the advantages of each. If the new individuals are better than the old ones, they need to be replaced. Formulas to calculate new individuals are as follows:

$$\overrightarrow{x^{new}} = \overrightarrow{x_i^t} + \vec{r} \times \left(\overrightarrow{x_*^t} - mutual\_agent \right) \tag{31}$$

$$\overrightarrow{y^{new}} = \overrightarrow{y_j^t} + \vec{r} \times \left(\overrightarrow{y_*^t} + Mutual\_agent \right) \tag{32}$$

Above all, the hybrid algorithm using two algorithms and multiple strategies is called MCPWOA. The flow chart of MCPWOA is shown in Figure 4, and the pseudo-code is shown in APPENDIX A.

Figure 4
The flow chart of MCPWOA.



3. Experimental Results on CEC2022 and Discussion

In this section, we conduct test experiments using twelve benchmark functions from CEC2022 to showcase the performance of MCPWOA and demonstrate its superiority. The descriptions and formulations of these functions can be found in the APPENDIX C. All experiments run on Windows 11, with an AMD R7 5800H 16GB processor. Algorithms are programmed using MATLAB R2023a. This section compares the performance of each algorithm across every test function.

The CEC2022 benchmark functions set consists of basic functions F1-F5, hybrid functions F6-F8, and composition functions F9-F12 (Dimensions = 10 and 20). CEC2022 enables evaluation from multiple perspectives using multiple test functions. To eliminate randomness in metaheuristic algorithms, this paper conducted 20 independent experiments on each test function using each algorithm, ensuring fairness by equal conditions. The experiment ran for a maximum of 1000 iterations. The results recorded the average (Avg) of each algorithm across 20 experiments. In optimization, a smaller average indicates higher precision.

This section evaluates the performance of each algorithm by comparing the average value with the theoretical optimal value. MCPWOA is compared with enhanced CPWOA, original WOA, original CPO, and other cutting-edge metaheuristic algorithms including Fungal growth optimizer (FGO), Dream Optimization Algorithm (DOA), Snow Geese Algorithm (SGA) [40-42]. APPENDIX D shows the parameters in WOA, CPO, CPWOA, MCPWOA, other algorithms use the default parameters in original papers. Table 1 shows the results of each algorithm on every test function in 10 dimensions.

Initial comparative analysis between CPWOA and WOA, as evidenced in Table 2, demonstrates CPWOA's superior performance across all benchmark functions except F4 and F10. Notably, it exhibits enhanced exploitation capabilities relative to WOA, particularly evident in F6 results. The two algorithms demonstrate equivalent performance on F10. This superiority originates from CPWOA's integration of two strategic components: 1) CPO-derived defense mechanisms embedded in WOA's core phases (encircling prey and bubble-net attacking), and 2) a cyclic population reduction protocol that

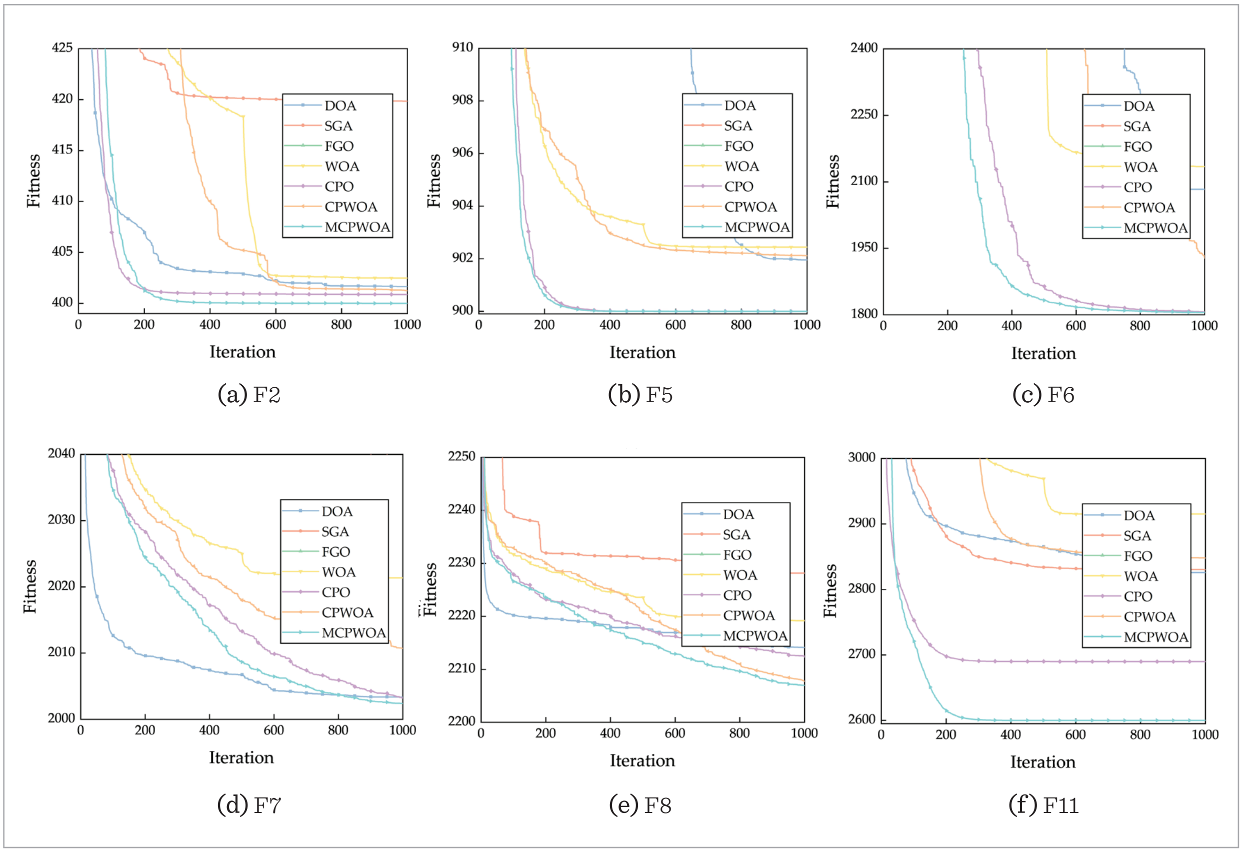
Table 1

The results of each algorithm on every test function in 10 dimensions.

	DOA	SGA	FGO	WOA	CPO	CPWOA	MCPWOA
F1	3.0000E +02	3.0130E +02	1.7855E +04	3.7500E +02	3.0000E +02	3.0021E +02	3.0002E +02
F2	4.0165E +02	4.1985E +02	7.3175E +02	4.0248E +02	4.0087E +02	4.0125E +02	4.0000E +02
F3	6.0000E +02	6.1266E +02	6.4787E +02	6.0112E +02	6.0000E +02	6.0022E +02	6.0000E +02
F4	8.1811E +02	8.2561E +02	8.7039E +02	8.0910E +02	8.1796E +02	8.1174E +02	8.1061E +02
F5	9.0195E +02	1.0193E +03	1.8927E +03	9.0244E +02	9.0000E +02	9.0211E +02	9.0000E +02
F6	2.0830E +03	4.2985E +03	5.8439E +07	2.1344E +03	1.8071E +03	1.9279E +03	1.8050E +03
F7	2.0034E +03	2.0401E +03	2.1195E +03	2.0213E +03	2.0032E +03	2.0107E +03	2.0024E +03
F8	2.2141E +03	2.2281E +03	2.2572E +03	2.2192E +03	2.2123E +03	2.2077E +03	2.2070E +03
F9	2.5293E +03	2.5315E +03	2.7141E +03	2.5404E +03	2.5293E +03	2.5293E +03	2.5293E +03
F10	2.4328E +03	2.5260E +03	2.5632E +03	2.5004E +03	2.5113E +03	2.5004E +03	2.5003E +03
F11	2.8257E +03	2.8303E +03	1.9056E +04	2.9152E +03	2.6900E +03	2.8481E +03	2.6000E +03
F12	2.8643E +03	2.8643E +03	2.9819E +03	2.8690E +03	2.8652E +03	2.8687E +03	2.8643E +03

Figure 5

Convergence curve of all algorithms on chosen functions in CEC2022.

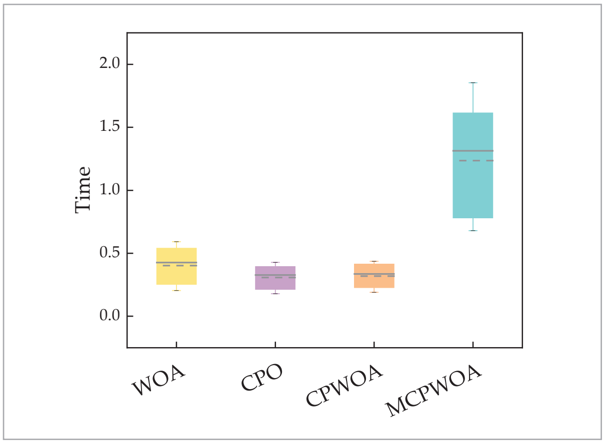


enhances swarm diversity. Furthermore, the arcsine control strategy (ACS) maintains effective exploration-exploitation equilibrium.

Table 2 further reveals MCPWOA's exceptional competitiveness across 10-dimensional CEC2022 test functions. The algorithm achieves first-rank performance in all categories except F1, F4 and F10, consistently outperforming counterparts in basic, hybrid, and composition function benchmarks. While demonstrating superiority over WOA, CPO and CPWOA on F10, it trails the well-established DOA algorithm, which is also a normal and powerful optimization algorithm. To visualize the process of searching for the optimal value and solution and showcase MCPWOA's convergence capacity, 20 additional independent experiments are conducted on each test function. The convergence curves of all algorithms regarding average fitness are presented in Figure 5. From these curves, it is evident that MCPWOA's convergence capacity is better than other metaheuristic algorithms. MCPWOA's superior convergence precision on CEC2022 indicates its potential to solve the MTSP problem. The superiority of MCPWOA is attributed to POS combining two algorithms with powerful exploitation capacity and undergoing a secondary exploitation. The effect of POS is obvious on F6, F7, F8 and F10, F11, F12, including three hybrid functions and three composition functions. When the performance of WOA and CPWOA differs significantly, MCPWOA can always adopt the better-performing algorithm because it integrates the two populations from both WOA and CPWOA. In addition, secondary exploitation can unlock capabilities that are not inherently possessed by WOA and CPWOA themselves. Moreover, CMS creates a more diverse original population,

Figure 6

Time cost of four algorithms.



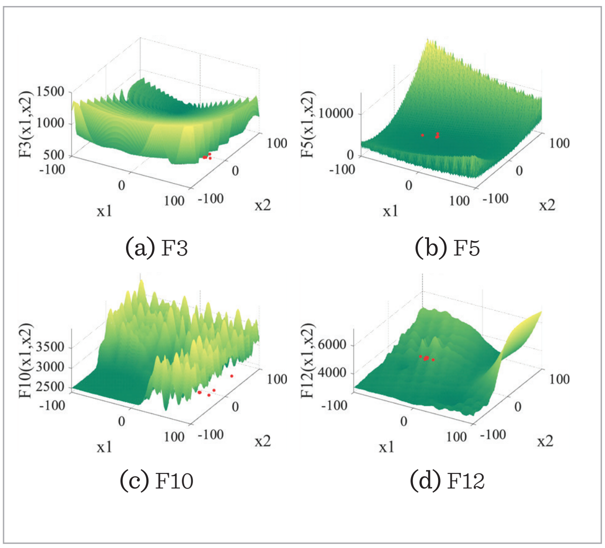
and RLS generates revolutions to open up unknown areas, leading to more opportunities to obtain better solutions.

In order to quantify the computational time costs across algorithms, the average time consumption per test function was recorded over 20 independent experimental trials. The boxplot in Figure 6 provides detailed visualization of experimental results, revealing that MCPWOA incurs higher computational costs. This phenomenon is attributable to its hybrid architecture: as an integration of two distinct algorithms, MCPWOA's update process simultaneously maintains dual populations (prey and whale populations), resulting in approximately double the iteration time compared to standalone CPWOA or CPO implementations. Furthermore, the reverse population generation and subsequent population re-update mechanisms in RLS introduce additional computational overhead, as evidenced by time complexity analysis.

Further experiments provide a more detailed demonstration of the CEC2022 function set landscapes while visualizing the search process of MCPWOA in two-dimensional space. The experimental results are presented in Figures 7-10. Red dots in Figure 7 denote the solutions generated during each iteration. In Figure 8, a progressive color gradient denotes the search process, where darker shades indicate proximity to the end of the iteration. The abrupt color transition observed suggests MCPWOA's remarkable convergence capability in the two-dimensional objective function. Figures 9 and 10 respectively demonstrate the average population positions along the x1 and x2 dimensions. In Figure 9, the region delineated by an orange outline substantiates MCPWOA's superior capabilities. Subplot (b) illustrates how the ACS ensures exploratory capacity in the algorithm's early-stage, facilitating comprehensive region search. Subplot (c) visually confirms the RLS strategy's efficacy in circumventing local optima traps.

Figure 7

Visualization of solutions on every function.



(a) F3 (b) F5 (c) F10 (d) F12

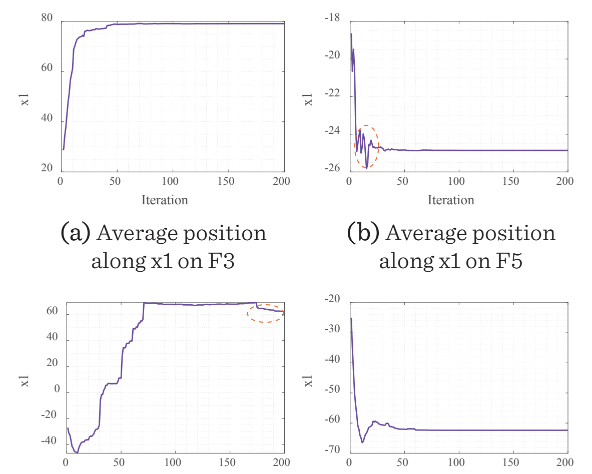
Figure 8

Search history on every function.

(a) Search history on F3 (b) Search history on F5 (c) Search history on F10 (d) Search history on F12

Figure 9

Average population positions along x1.



(a) Average position along x1 on F3

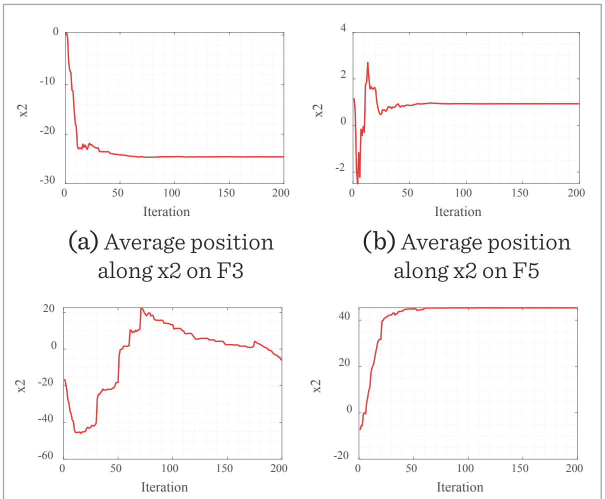
(b) Average position along x1 on F5

(c) Average position along x1 on F10

(d) Average position along x1 on F12

Figure 10

Average population positions along x2.



(a) Average position along x2 on F3

(b) Average position along x2 on F5

(c) Average position along x2 on F10

(d) Average position along x2 on F12

4. Experimental Results on CEC2022 and Discussion

In this section, we evaluate the capability of MCP-WOA in solving the Multiple Traveling Salesman Problem (MTSP) using a clustering-based approach on TSPLIB dataset, and design an Asynchronous Combinatorial Method (ACM) to mitigate the sensitivity of k-means clustering to the selection of initial cluster centers, then apply it to real bushfires in Australia, a country with frequent bushfires. Case studies are conducted.

4.1 K-means Clustering Algorithm

Clustering transformation is a usual method to solve MTSP problems, which refers to dividing all cities on the map into several groups using a clustering algorithm and assigning to every traveling salesman to cover. Then the traversal problem of multiple fire nodes problem in this research is transferred to MD-MTSP problems. Clustering algorithms are primarily used to divide objects in a dataset into several clusters based on criteria such as similarity or distance.

K-means is a classic partitioning clustering algorithm that iteratively optimizes cluster centroids to minimize within-cluster sum of squared errors. K-means belongs to unsupervised learning. Initially, K-means selects K original centers of clustering and iterates. In each iteration, every data point is assigned to the nearest center, and then centers are updated using the mean of each cluster. The iteration stops when it reaches a limit or when changes in centers fall below a given precision.

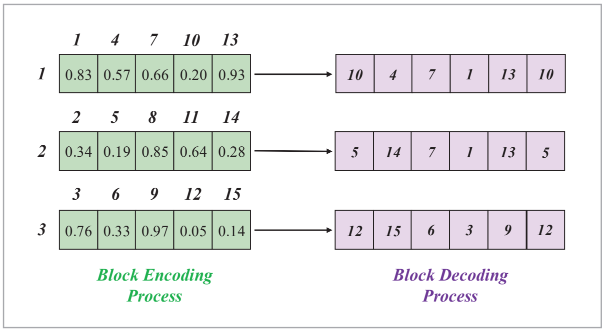
4.2 City block Encoding and Block Decoding

Solving TSP problems after clustering requires the use of city encoding, a general method that maps solutions to TSP problems and positions of individuals in metaheuristic algorithms. After encoding, cities can be treated as individuals and updated using MCPWOA. Decoding involves determining the order of city visits for the traveling salesman and calculating the total distance as fitness. Based on this distance, MCPWOA updates the codes. The encoding-decoding cycle is the core process.

K-means partitions all cities into M groups, enabling block encoding and decoding. For instance, consider three traveling salesmen visiting fifteen cities. After K-means, the first salesman visits cities 1, 4, 7, 10, and 13, the second visits 2, 5, 8, 11, and 14, and the third visits 3, 6, 9, 12, and 15. Distribute a random number from 0 to 1 to every city as its code, constituting the encoding process. In the decoding process, each salesman visits cities in order of smallest to largest code, returning to the first city after visiting the last. Thus, order updates due to code updates, allowing MCPWOA to search for the shortest path via the encoding-decoding cycle. Figure 11 illustrates encoding and decoding process.

Figure 11

Process of city block encoding and block decoding.



What is better is that the encoding and decoding process above can perfectly satisfy constraints in MTSP problems. In this process, each node can only be visited once by one UAV and every UAV will return to the starting node finally. In addition, if a UAV visits a certain node, it will leave that node.

4.3 Evaluation Using MCPWOA on TSPLIB

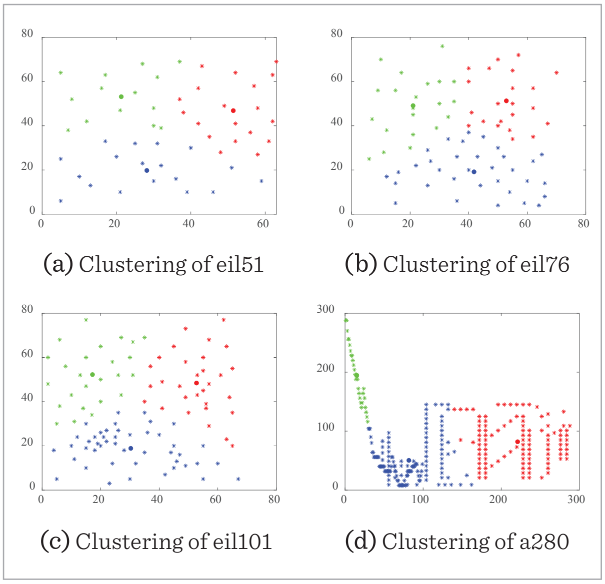
The TSPLIB (Traveling Salesman Problem Library) is a dedicated repository for storing and sharing instance data of the Traveling Salesman Problem (TSP) and related problems. In this study, we select eil51, eil76, eil101 and a280 as instances and conduct experiments in two aspects.

First, the four algorithms above (WOA, CPO, CPWOA, MCPWOA) are used to solve MTSP based on K-means clustering in the four instances to prove that MCPWOA is still powerful through comparison. During this process, the parameters in all algorithms are the same as those in APPENDIX C. For fairness, all experiments are based on the same original centers and clustering results using specific random number seeds. At the same time, to reduce randomness, this paper takes 10 independent experiments on each instance.

The evaluation results are presented in Table 2. The initial clustering results and the final optimized paths planned by MCPWOA are presented in Figures 12-13, respectively. Meanwhile, the average convergence curves, which show the total cost during iteration, and box plots of all algorithms across 10 independent experimental trials are illustrated in Figures 14-15. The results indicate that CPWOA remains more competitive than WOA, and MCPWOA

Figure 12

The initial clustering results.

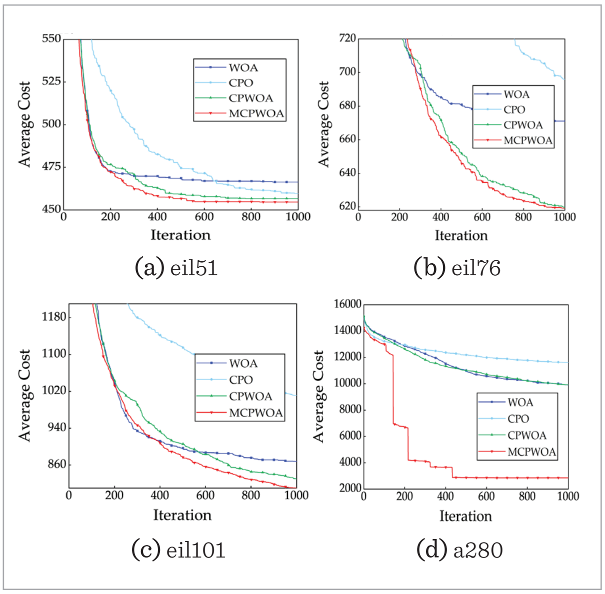


(a) Clustering of eil51 (b) Clustering of eil76

(c) Clustering of eil101 (d) Clustering of a280

Figure 13

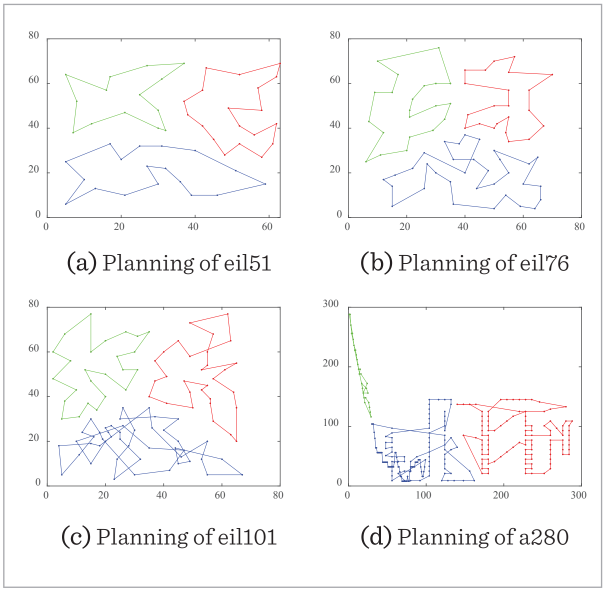
Average convergence curves showing total cost.



(a) eil51 (b) eil76

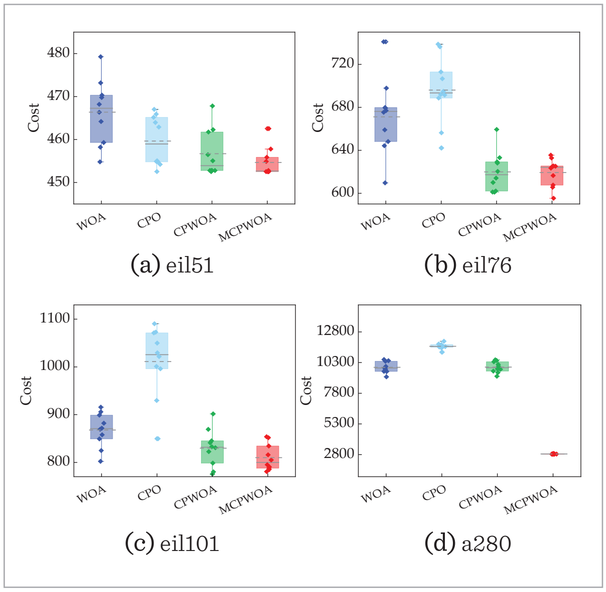
(c) eil101 (d) a280

Figure 14
Final optimized paths planned by MCPWOA.



(a) Planning of eil51 (b) Planning of eil76
(c) Planning of eil101 (d) Planning of a280

Figure 15
Box plots across 10 independent experimental trials.

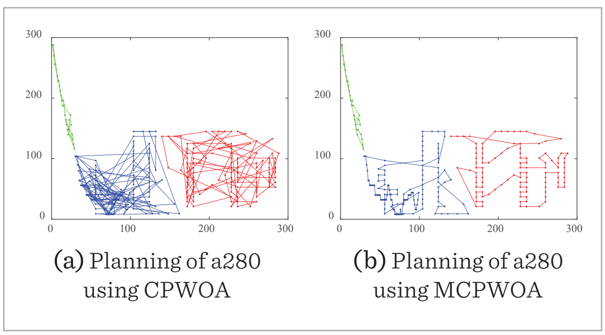


(a) eil51 (b) eil76
(c) eil101 (d) a280

is the most competitive among all algorithms for MTSP problems, achieving the minimum average cost across all algorithms. It is crucial to emphasize that MCPWOA exhibits exceptional performance on the a280 dataset — a structured map containing numerous nodes with regular spatial distribution.

As visualized in Figure 16, the final paths generated by CPWOA and MCPWOA reveal striking contrasts: MCPWOA successfully constructs an orderly traversal route through this regularized environment, whereas CPWOA produces chaotic path configurations. This compelling evidence highlights MCPWOA's enhanced exploration capacity during early optimization stages and its superior ability to escape local optima.

Figure 16
Highlight on a280.



(a) Planning of a280 using CPWOA (b) Planning of a280 using MCPWOA

4.4 Traditional and Clustering-Driven MTSP Solutions

A classic method for solving the MTSP is to encode all cities using a Genetic Algorithm (GA) without clustering them. The chromosome encoding still adopts a decimal encoding scheme. Each position in the encoding directly represents the sequence number of the city to be visited, thus eliminating the need for a decoding process. Unlike the TSP problem, each individual (solution) has a corresponding set of chromosome separation points. The purpose of dividing the chromosome separation points. The purpose of dividing the chromosome is to distinguish the path of each traveler. This method is illustrated in Figure 17. A comparison of GA method and Clustering-Driven method on eil101 is presented in Figures 18-19.

We compare the two methods based on the intersection characteristics of planned paths and the equilibrium of total path lengths per traveler. Figure 18 demonstrates the final paths generated by both methods on the eil101 map, where the cluster-based approach evidently avoids intersections between different travelers' routes. In practical UAV path planning scenarios, non-intersecting paths mitigate collision risks between drones. Figure 19 illustrates the convergence accuracy of both methods and the

Table 2

The evaluation results on TSPLIB.

		WOA	CPO	CPWOA	MCPWOA
eil51 (M=3)	Best	4.5481E +02	4.5481E +02	4.5253E +02	4.5253E +02
	Worst	4.7918E +02	4.7265E +02	4.6388E +02	4.5505E +02
	Average	4.6328E +02	4.6119E +02	4.5775E +02	4.5327E +02
eil76 (M=3)	Best	6.4702E +02	6.3438E +02	5.9916E +02	5.9180E +02
	Worst	7.2371E +02	7.7000E +02	6.5798E +02	6.3511E +02
	Average	6.8071E +02	7.2032E +02	6.2035E +02	6.1820E +02
eil101 (M=3)	Best	8.5195E +02	9.3915E +02	7.8065E +02	7.9006E +02
	Worst	9.5580E +02	1.1391E +03	9.4050E +02	8.4486E +02
	Average	9.0167E +02	1.0181E +03	8.3635E +02	8.1482E +02
a280 (M=3)	Best	9.1371E +03	1.1148E +04	9.1856E +03	2.8285E +03
	Worst	1.0544E +04	1.2036E +04	1.0527E +04	2.8875E +03
	Average	9.9124E +03	1.1621E +04	9.9287E +03	2.8518E +03

Figure 17

An illustration of the classical GA method.

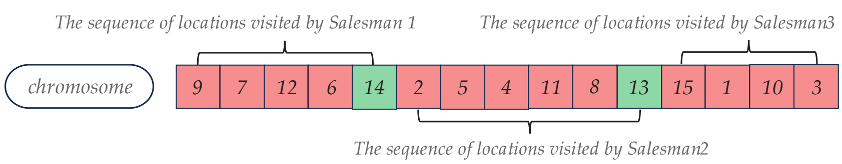
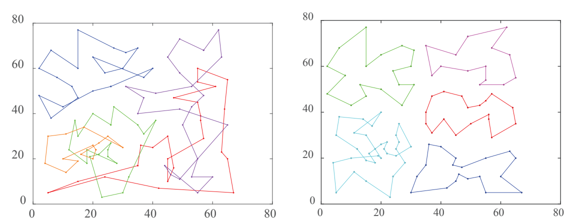


Figure 18

Final paths generated by both methods on the eil101 map.

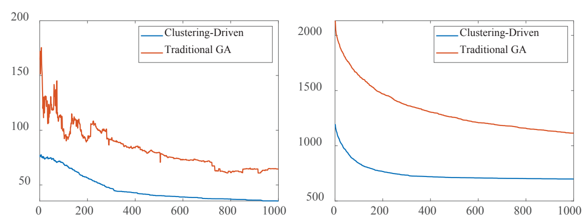


(a) Planning of eil101 based on GA

(b) Planning of eil101 based on Clustering

Figure 19

Standard deviation and convergence comparison.



(a) Standard deviation comparison

(b) Convergence comparison

standard deviation of all travelers' path lengths, with the latter metric reflecting planning equilibrium. Due to the bucket effect (where system performance is constrained by the weakest component), more balanced path distributions may lead to shorter overall traversal times — under the assumption of identical individual speeds, the total duration is determined by the longest path.

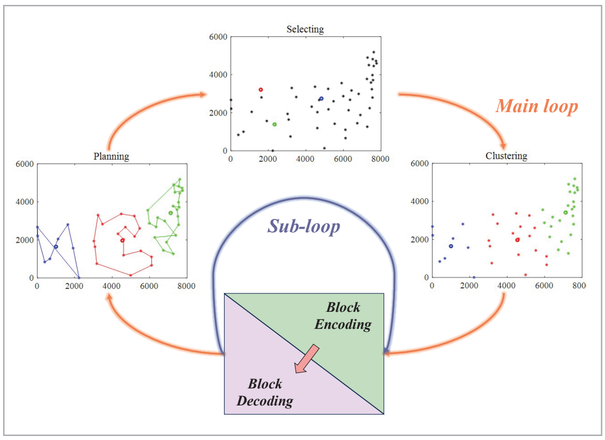
Experimental results demonstrate that the cluster-based MTSP approach exhibits comprehensive superiority over the traditional GA method, achieving both higher convergence accuracy and improved equilibrium in path length distribution. However, given the inherent sensitivity of clustering methods to initial conditions, we propose an asynchronous combinatorial approach to effectively search for favorable initial cluster centers.

4.5 Asynchronous Combinatorial Method (ACM) Base on MCPWOA

K-means is highly sensitive to initial center selection, with different initializations potentially leading to vastly different results. Thus, a single run of K-means cannot guarantee optimal clustering performance. A common approach to addressing this limitation is to run K-means multiple times and select the best result. This paper presents an asynchronous combinatorial method that iteratively updates both the initial centers and the order of the Traveling Salesman Problem (TSP). The method's framework consists of a main loop and a sub-loop, with the former updating initial centers and the latter modifying the TSP order. This technique is called "combinatorial" because it iterates over both aspects simultaneously and "asynchronous" because the main loop and sub-loop do not iterate at the same time. In each main loop iteration, a complete sub-loop process is required. The asynchronous combinatorial method's process is depicted in Figure 20, and its pseudocode is shown in APPENDIX B. It is clear that this method requires significant computational resources. Considering that both population interaction and reverse leaning processes significantly increase computation, the main loop only uses traditional CPO to update initial centers, while the sub-loop still employs hybrid MCPWOA to update the order, thereby reducing computation.

Figure 20

An illustration of ACM



5. Simulation Application on Real Bushfire Scenes in the Australia

In Australia, bushfires have emerged as a significant natural calamity, with the haunting aftermath of the 2019-20 Black Summer deeply etched in the nation's collective memory. The current method of bushfire detection in Australia heavily relies on public reports to emergency services. And the unchecked minor fire incidents have the potential to escalate into massive forest infernos.

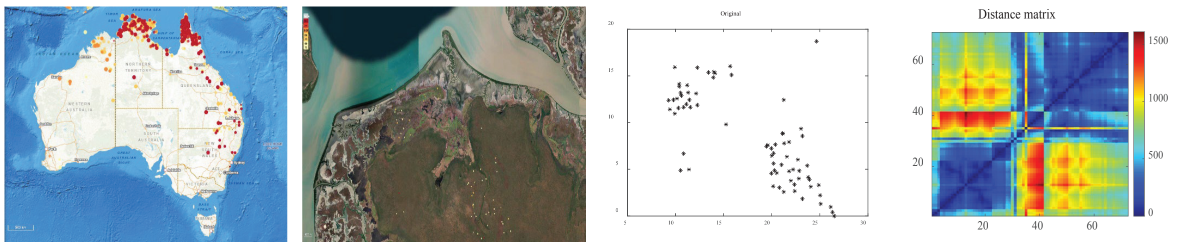
To verify our above algorithm's ability to solve MTSP and its practical applications, we utilized information from hotspots in Australia. Data from thermal infrared sensors on Earth-orbiting satellites aid in detecting fire 'hotspots,' which are areas of the land surface that are significantly warmer than their surroundings. While hotspots do not always indicate an active fire, they provide valuable information about the locations and directions of potential fires and fire risks. During bushfire outbreaks, numerous fire points emerge within regions influenced by airflow, facilitating the rapid spread of bushfires. Each fire point represents an early stage of a bushfire, emphasizing the critical need to identify the shortest path to efficiently cover and extinguish these fires promptly.

5.1 Case 1: Northern Australia

To verify the algorithm's ability to solve MTSP and its practical applications, information from Sentinel hotspots in the Northern Territory of Australia was utilized [12]. Data from thermal infrared sensors on Earth-orbiting satellites assist in detecting fire 'hotspots,' which are areas of the land surface that are significantly warmer than their surroundings. While hotspots do not always indicate a fire, they provide accurate information about the locations and directions of possible fires and fire risks. Figure 21(a) shows how many dense hotspots there are in the Northern Territory and northern Queensland of Australia. During a bushfire outbreak, numerous fire points emerge within areas influenced by airflow. The chosen region can facilitate the rapid spread of bushfires. All fire points represent early stages of the bushfires, emphasizing the critical need to identify the shortest path to efficiently cover and extinguish these fires promptly. In Australia, bushfires have emerged as a significant natural calamity, with the haunting aftermath of the 2019-20 Black Summer deeply etched in the nation's collective memory. The current method of bushfire detection in Australia heavily relies on public reports to emergency services. And the unchecked minor fire incidents have the potential to escalate into massive forest infernos.

Figure 21

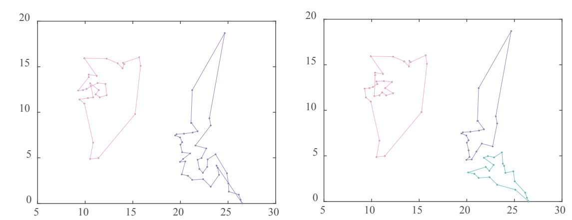
Case of northern Australia: hotspots distribution in Australia and the region chosen.



(a) (b) (c) (d)

Figure 22

Planned paths when M=2 and M=3.

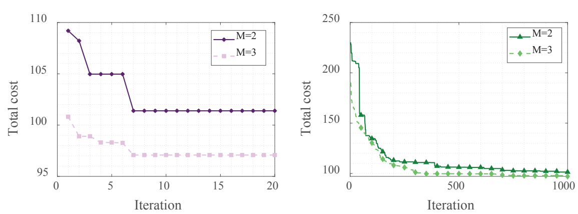


(a) Planned paths when M=2

(b) Planned paths when M=3

Figure 23

Convergence curve for both main loop and sub-loop in asynchronous combinatorial method.

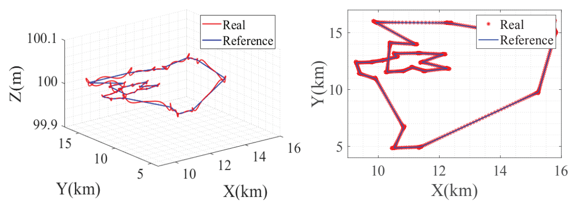


(a) Main loop

(b) Sub-loop

Figure 24

Tracking in three-dimensional space and two -dimensional space.

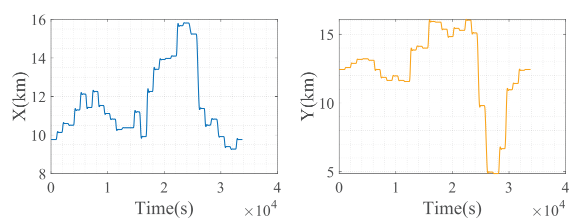


(a) Tracking in three-dimensional space

(b) Tracking in two-dimensional space

Figure 25

Positional evolution in both x- and y-coordinates.



(a) Trajectory in X coordinate

(b) Trajectory in Y coordinate

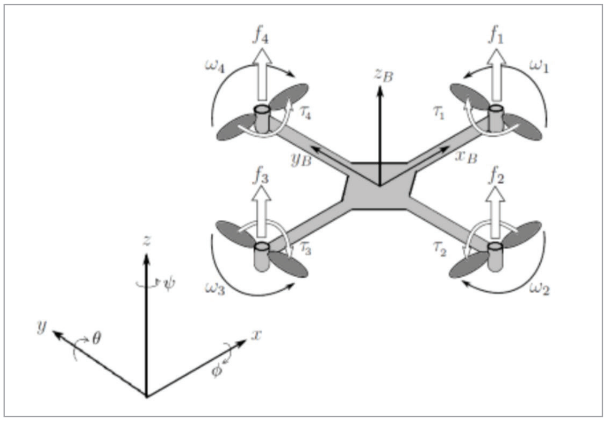
Figure 21(a) depicts the number of dense hotspots in the Northern Territory and northern Queensland of Australia. In contrast to other administrative divisions within Australia, the Northern Territory falls short of the prerequisites for statehood, remaining classified as a "territory" due to its sparse population. The region faces constant threat of minor fires escalating into catastrophes. In recent years, the Northern Territory has endured severe bushfire damage, notably the Barkly region inferno in September 2023, which ravaged 9,300 square kilometres of land. For research purposes, we selected a 330 square kilometre area situated on the north side of the South Alligator River, 190 kilometres from Darwin, the Northern Territory's capital. This locale is proximate to Kakadu National Park, a designated World Heritage Site renowned for its rich ecological diversity, housing a plethora of habitats, flora, fauna, and significant human historical remnants. Moreover, the region harbours substantial mineral resources, with the Ranger Uranium Mine within its vicinity ranking among the world's most prolific uranium extraction sites. Despite its ecological and historical significance, the area is characterized by abundant dry grass fuels and understorey vegetation, necessitating early detection and suppression of fires to mitigate the risk of large-scale conflagrations. The distribution of hotspots in this region at 13:00 on July 21st, 2014, is depicted in Figure 21(b), with the processed coordinates illustrated in Figure 21(c), with measurements in kilometres. Distance matrix of the map is shown in Figure 21(d).

The path planning results for the case of northern Australia using the asynchronous combinatorial method proposed are shown in Figure 22. This paper takes different scenes with different numbers of UAVs. When there are two UAVs, the optimal total cost searched is 101.391 km and its value is 97.082 km when there are three UAVs. Main loop process and sub-loop process in the asynchronous combinatorial method are both shown in Figure 23. It is obvious that the main loop could indeed search for optimal original clustering centres, which proves that the method proposed is more competitive than the traditional method consisting of K-means and metaheuristic algorithms.

This paper investigates a UAV-enabled Multiple Traveling Salesman Problem (UAV-MTSP) in the context of forest fire rescue operations. Unlike conventional MTSP, the proposed formulation explicitly incorporates UAV endurance constraints and dwell time requirements at each target location. During these dwell periods, UAVs must complete critical fire assessment and rescue procedures, such as delivering emergency supplies. To realistically simulate UAV behavior in our UAV-MTSP framework, we adopt a standard quadrotor UAV model with comprehensive flight dynamics and implement a nonlinear model predictive controller (NMPC) for trajectory tracking.

Figure 26

Model of quadrotor UAV.



Set the UAV dwell time to 1000 seconds (1000s) at each target location. Assuming a UAV flight speed of 15 m/s and an altitude of 100 m, we analyze the trajectory tracking performance under mission configuration M=3. As shown in Figure 24, the planned trajectory of UAV 1 exhibits precise tracking behavior. Its positional evolution in both x- and y-coordinates is detailed in Figure 25, where the horizontal plateaus in the trajectory correspond to dwell phases at target locations for fire assessment and payload delivery. The standardized quadrotor UAV platform used in this study is illustrated in Figure 26.

Several UAVs can cover and monitor all high-risk fire hotspots along the planned path with optimal distance cost, significantly aiding fire prevention.

5.2 Case 2: Southern Australia

To assess the algorithm's capacity to handle a greater number of coordinate points, a map featuring 476 fire points is introduced. This study examines bushfire incidence in southern Australia between October

1st, 2019, and November 1st, 2019, with a focus on major fires that occurred during this period around the Gearys Flat region, resulting in the identification of 476 fire points. The search region is located approximately at latitude 31.15 and longitude 152.51. The fire archive and the original fire count map are depicted in Figure 27, with measurements in miles.

The path planning outcomes for the southern Australia scenario are illustrated in Figure 28, showcasing diverse scenarios with varying numbers of Unmanned Aerial Vehicles (UAVs). In this context, a higher number of UAVs is utilized to navigate this expansive map. With seven and ten UAVs, the optimal total search distances are calculated.

Main loop process and sub-loop process are both shown in Figure 29. Based on the primary loop convergence curve, it is evident that the proposed method retains robust efficacy when handling extensive coordinate points. Along the planned path, multiple UAVs can effectively cover and monitor all fire points to prevent them from further developing into bushfires. Main loop process and sub-loop process are both shown in Figure 29.

Further experiments quantified the time cost of the proposed method, which are shown in Figure 30. It can be clearly observed that as the number of UAVs and map scale increase, the algorithm's runtime shows significant growth. This stems from the dual-loop structure of the asynchronous combinatorial method and the dual-population architecture of MCPWOA. Compared with traditional algorithms, the advantages of our proposed method lie in convergence accuracy and path planning consistency, while effectively avoiding potential collision risks between UAVs.

Figure 27

Case of southern Australia: fire points distribution in chosen region and fire points map.

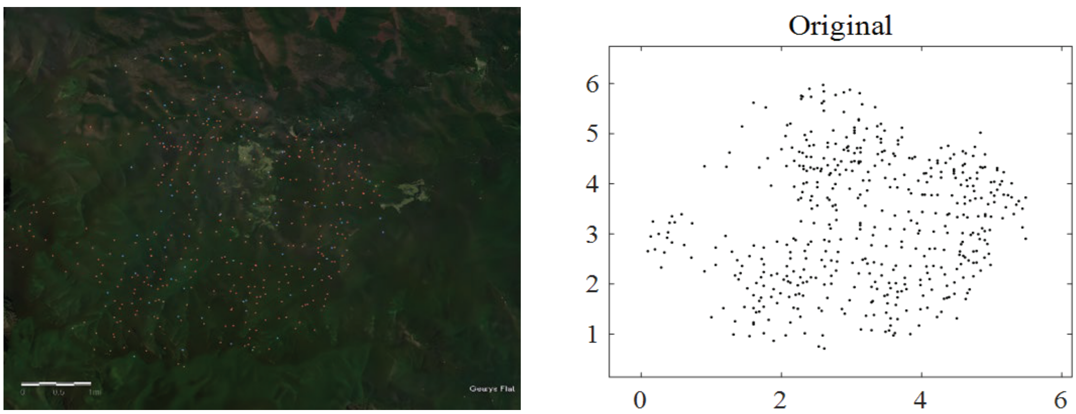
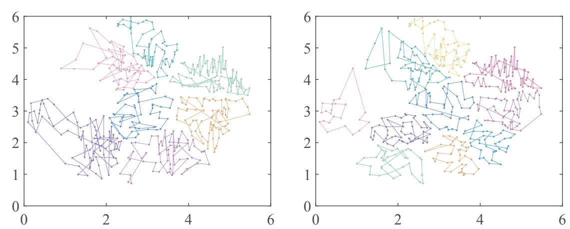


Figure 28

Planned paths when M=7 and M=10.

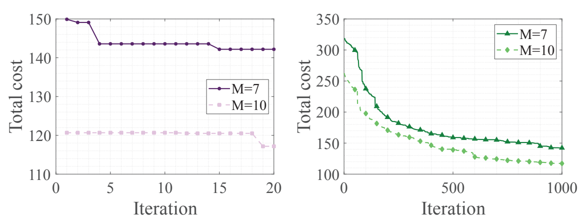


(a) Planned paths when M=7

(b) Planned paths when M=10

Figure 29

Convergence curve for two loops.

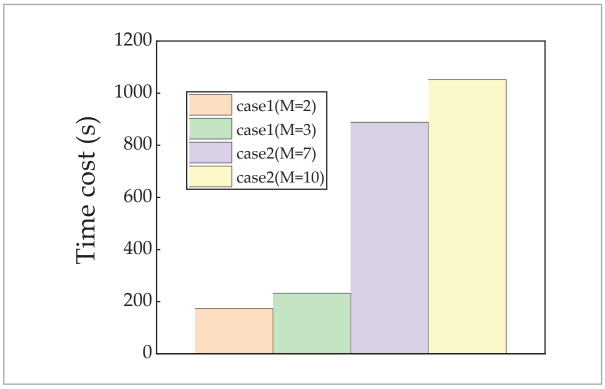


(a) Main loop

(b) Sub-loop

Figure 30

Time cost for two cases.



6. Conclusion

This paper proposes MCPWOA, a multi-strategy improved metaheuristic algorithm base on both whale optimization algorithm and crested porcupine optimizer, to solve the path planning problem of UAV in bushfire prevention and rescue process.

In response to existing problems in the original whale optimization algorithm, such as a tendency to fall into local minima, an improved algorithm called CPWOA is proposed. CPWOA introduces unique defence methods from an advanced crested porcupine optimizer and a cyclic population reduction technique to increase population diversity and enhance exploration and exploitation capabilities of the whale optimization algorithm. To further obtain an algorithm with better performance, MCPWOA is proposed. Firstly, predatory organisms search is used to integrate CPWOA and CPO to simulate the predation relationship between the whale population and the prey population. Predatory organisms search can perform second-order exploitation on the community consisting of the two populations. Secondly, tent chaotic mapping enhances the quality of the original population, offering more possibilities for the algorithm to find the optimal solution. Thirdly, the arcsine control strategy is used to make parameters controlling the exploration and exploitation process decrease through the iteration process, balancing those two processes. Finally, the reverse learning strategy generates revolution during convergence, providing a chance to learn again for individuals performing worse and accelerating convergence. Benchmark tests using advanced CEC2022 indicate that CPWOA performs better than the original WOA on most functions, and MCPWOA owns better convergence speed and precision than other algorithms, proving MCPWOA is competitive when dealing with optimization problems.

To solve MTSP problems, the K-means clustering algorithm and city encoding-decoding method are proposed to split an MTSP problem into M TSP problems, which can then be solved by a metaheuristic algorithm. To prove that MCPWOA is still competitive on MTSP problems, four TSPLIB instances are introduced. Experimental results show that it still performs at the top level in terms of average total cost. Given that the original clustering centres have a strong impact on clustering results, an asynchronous combinatorial method is proposed to search for the optimal distribution of original centres using two coupled cycles. Experiments on the same instances demonstrate that the asynchronous combinatorial method can find better original centres. Regarding simulation application to real bushfire scenes, this paper introduces two typical bushfire scenes that occurred in Australia. The simulation experiment shows that the proposed method is useful during the path planning process for UAVs to monitor hotspots or rescue fire points.

Data Sharing Agreement

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

Appendix A: Pseudocode of the MCPWOA

Algorithm 1: MCPWOA

1:// Set parameters in CPO and CPWOA:

Set parameters α, T_f in CPO;
Set parameters a, b in CPWOA;

2:// Set control parameters:

Set control parameters N, N_{min}, T, T_{max}, k, J_r;

3:// Generate chaotic sequence:

Using Eq (22) and Eq (23)

4:// Initialize two populations:

Initialize whale population P_1 using Eq (24)
Initialize prey population P_1 using Eq (24)

5:// Update and integrate two populations:

while(t<=T_{max}):
$P_1 = \{\overrightarrow{x_1^t}, \overrightarrow{x_2^t}, \dots, \overrightarrow{x_N^t}\}$, $P_2 = \{\overrightarrow{y_1^t}, \overrightarrow{y_2^t}, \dots, \overrightarrow{y_N^t}\}$;
Evaluate fitness values for each individual Determine the best x_*^t and y_*^t so far;
Update model parameters using Eq (9), Eq (25) and Eq (26);
Update P_1 using Eq (2) or Eq (20) or Eq (21);
Reverse learning process: generate reverse population OP_1 using Eq (27) and select the best N individuals to constitute P_*;
$P_1 = P_*$
Update P_2 using Eq (10) or Eq (12) or Eq (13) or Eq (17);
Reverse learning process: generate reverse population OP_2 using Eq (27) and select the best N individuals to constitute P_*;
$P_2 = P_*$
Randomly select an individual $\overrightarrow{x_i^t}$ and an individual $\overrightarrow{y_j^t}$;
Calculate mutual vector using Eq (30);
Calculate $\overrightarrow{x^{new}}$ and $\overrightarrow{y^{new}}$ using Eq (31) and Eq (32);
if $\overrightarrow{x^{new}}$ ***or*** $\overrightarrow{y^{new}}$ ***is better:***
Update the two populations;
Update the best x_*^t and y_*^t so far;
end if
$t=t+1$;
end while

6:// Return the solutions:

Return the better one between x_*^t and y_*^t;

Appendix B: Pseudocode of the ACM

Algorithm 2: ACM

1:// Set parameters in CPO and MCPWOA:

2:// Set control parameters:

Set control parameters M, $T_1 = 5$, $T_2 = 1000$;

3:// Initialize the population of centres. (Size=5)

4:// Main loop:

while(i<=T_1):
for every original centres' distribution:
Use K-means to divide cities into M clusters;
for every cluster:
Initialize the population of the cluster in the form of block encoding;

5:// Sub-loop:

while(j<=T_2):
Decode and get visit order, then calculate cost;
Update city code using MCPWOA;
$j=j+1$;
end while

Determine the optimal cost and solution of this cluster;
end for
Calculate the optimal total cost and determine the optimal global solution according to this centres' distribution;
Update original centers using CPO;
end for
Determine the optimal total cost, the optimal global solution and the optimal distribution among all original centres' distribution;
$i=i+1$;
end while

6:// Return:

Return the optimal total cost, the optimal global solution and the optimal original centres' distribution;

Appendix C:
CEC 2022 Benchmark Test Functions

Type	No.	Description	Formulation	Range	Dimension	f_{min}
Unimodal	F1	Shifted and full Rotated Zakharov Function	$f_1(x)=\sum_{i=1}^{n}x_i^2+\left(\sum_{i=1}^{n}0.5x_1\right)^2+\left(\sum_{i=1}^{n}0.5x_i\right)^4$	[-100,100]	10/20	300
Multimodal	F2	Shifted and full Rotated Rosenbrock's Function	$f_2(x)=\sum_{i=1}^{n-1}\left[100\cdot\left(x_{i+1}-x_i^2\right)^2+\left(x_i-1\right)^2\right]$	[-100,100]	10/20	400
Multimodal	F3	Shifted and full Rotated Rastrigin's Function	$f_3(x)=\sum_{i=1}^{n}\left[x_i^2-10\cdot cos\left(2\pi x_i\right)+10\right]$	[-100,100]	10/20	600
Multimodal	F4	Shifted and full Rotated Non-Continuous Rastrigin's Function	$f_4(x)=\sum_{i=1}^{n}\left[y_i^2-10\cdot cos\left(2\pi y_i\right)+10\right]$	[-100,100]	10/20	800
Multimodal	F5	Shifted and full Rotated Levy Function	$f_5(x)=sin^2(\pi w_1)+\sum_{i=1}^{n-1}(w_i-1)^2\left[1+10sin^2(\pi w_i+1)\right]+(w_n-1)^2\left[1+sin^2(2\pi w_n)\right]$	[-100,100]	10/20	900
Hybrid	F6	Hybrid Function 1 (N=3)		[-100,100]	10/20	1800
Hybrid	F7	Hybrid Function 2 (N=6)		[-100,100]	10/20	2000
Hybrid	F8	Hybrid Function 3 (N=5)		[-100,100]	10/20	2200
Composition	F9	Composition Function 1 (N=5)		[-100,100]	10/20	2300
Composition	F10	Composition Function 2 (N=4)		[-100,100]	10/20	2400
Composition	F11	Composition Function 3 (N=5)		[-100,100]	10/20	2500
Composition	F12	Composition Function 4 (N=6)		[-100,100]	10/20	2700

Appendix D:
CEC 2022 Benchmark Test Functions

WOA	Parameter a	Decrease linearly from 2 to 0
	Parameter b	1
CPO	Parameter N_{min}	80
	Parameter T	2
	Parameter α	0.1
	Parameter T_f	0.3
CPWOA	Parameter b	Same to WOA
	Parameter N_{min}, T, α	Same to CPO
	Parameter a	Decrease from 2 to 0 along arcsine curve
	Original Population size in all algorithms above	130

MCPWOA	Parameter b, N_{min}, T, α	Same to WOA and CPO
	Parameter a	Decreas from 2 to 0 along arcsine curve
	Parameter T_f	Decreas from 1 to 0 along arcsine curve
	Parameter β in CMS	0.499
	Parameter k, J_r in RLS	10
	Parameter J_r in RLS	0.8
	Original Population size	130

References

1. Abdel-Basset, M., El-Shahat, D., Sangaiah, A. K. A Modified Nature-Inspired Meta-Heuristic Whale Optimization Algorithm for Solving 0-1 Knapsack Problem. International Journal of Machine Learning and Cybernetics, 2019, 10, 495-514. https://doi.org/10.1007/s13042-017-0731-3
2. Abdel-Basset, M., Mohamed, R., Abouhawwash, M. Crested Porcupine Optimizer: A New Nature-Inspired Metaheuristic. Knowledge-Based Systems, 2024, 284, 111257. https://doi.org/10.1016/j.knosys.2023.111257
3. Abdel-Basset, M., Mohamed, R., Abouhawwash, M. Fungal Growth Optimizer: A Novel Nature-Inspired Metaheuristic Algorithm for Stochastic Optimization. Computer Methods in Applied Mechanics and Engineering, 2025, 437, 117825. https://doi.org/10.1016/j.cma.2025.117825
4. Abdel-Basset, M., Mohamed, R., Jameel, M., Abouhawwash, M. Nutcracker Optimizer: A Novel Nature-Inspired Metaheuristic Algorithm for Global Optimization and Engineering Design Problems. Knowledge-Based Systems, 2023, 262, 110248. https://doi.org/10.1016/j.knosys.2022.110248
5. Ai-Qing, T., Liu, F. F., Lv, H. X. Snow Geese Algorithm: A Novel Migration-Inspired Meta-Heuristic Algorithm for Constrained Engineering Optimization Problems. Applied Mathematical Modelling, 2024, 126, 327-347. https://doi.org/10.1016/j.apm.2023.10.045
6. Akhmetov, I., Pak, A. TSP Review: Performance Comparison of the Well-Known Methods on a Standardized Dataset. In Proceedings of the 2023 19th International Asian School-Seminar on Optimization Problems of Complex Systems (OPCS), IEEE, August 2023, pp. 1-6. https://doi.org/10.1109/OPCS59592.2023.10275749
7. Akter, S., Kim, D. Y., Yoon, S. Task Offloading in Multi-Access Edge Computing Enabled UAV-Aided Emergency Response Operations. IEEE Access, 2023, 11, 23167-23188. https://doi.org/10.1109/ACCESS.2023.3252575
8. Alkafaween, E., Elmougy, S., Essa, E., Mnasri, S., Tarawneh, A. S., Hassanat, A. IAM-TSP: Iterative Approximate Methods for Solving the Travelling Salesman Problem. International Journal of Advanced Computer Science & Applications, 2023, 14(11). https://doi.org/10.14569/IJACSA.2023.0141143
9. Almasoud, A. M. UAV-Based Intelligent Transportation System for Emergency Reporting in Coverage Holes of Wireless Networks. Sensors, 2021, 21(19), 6371. https://doi.org/10.3390/s21196371
10. Applegate, D., Bixby, R., Cook, W., Chvátal, V. On the Solution of Traveling Salesman Problems. 1998. (Publication details incomplete-please provide journal name or conference for full citation)
11. Arafat, M. Y., Moh, S. Localization and Clustering Based on Swarm Intelligence in UAV Networks for Emergency Communications. IEEE Internet of Things Journal, 2019, 6(5), 8958-8976. https://doi.org/10.1109/JIOT.2019.2925567
12. ArcGIS StoryMap. (n.d.). https://storymaps.arcgis.com/stories/9ebbe1b54dc847f2a7dd01917c9f307
13. Behnezhad, S., Roghani, M., Rubinstein, A., Saberi, A. Sublinear Algorithms for TSP via Path Covers. arXiv preprint, 2023. arXiv:2301.05350
14. Chen, Y., Kannan, S., Khanna, S. Sublinear Algorithms and Lower Bounds for Metric TSP Cost Estimation. arXiv preprint, 2020. arXiv:2006.05490
15. Dantzig, G., Fulkerson, R., Johnson, S. Solution of a Large-Scale Traveling-Salesman Problem. Journal of the Operations Research Society of America, 1954, 2(4), 393-410. https://doi.org/10.1287/opre.2.4.393
16. Gan, F., Liu, D. Forest Fire Fast Response Method Based on Optimized UAV Algorithm. In Proceedings of the 2022 7th International Conference on Intelligent Computing and Signal Processing (ICSP), IEEE, April 2022, pp. 47-51. https://doi.org/10.1109/ICSP54964.2022.9778521
17. Gulcu, S. D., Ornek, H. K. Solution of Multiple Travelling Salesman Problem Using Particle Swarm Optimization Based Algorithms. International Journal of Intelligent Systems and Applications in Engineering, 2019, 7(2), 72-82. https://doi.org/10.18201/ijisae.2019252784
18. Guo, P., Hou, M., Ye, L. MEATSP: A Membrane Evolutionary Algorithm for Solving TSP. IEEE Access, 2020, 8, 199081-199096. https://doi.org/10.1109/ACCESS.2020.3035058
19. Held, M., Karp, R. M. A Dynamic Programming Approach to Sequencing Problems. Journal of the Society for Industrial and Applied Mathematics, 1962, 10(1), 196-210. https://doi.org/10.1137/0110015

20. Hong, L., Wang, Y., Du, Y., Chen, X., Zheng, Y. UAV Search-and-Rescue Planning Using an Adaptive Memetic Algorithm. Frontiers of Information Technology & Electronic Engineering, 2021, 22(11), 1477-1491. https://doi.org/10.1631/FITEE.2000632

21. Hu, Z., Yang, G. A Local Optimization Algorithm Based on Eliminating the Inclusion and Intersection Relations Between Sub Tours for Multi-Traveling Salesman Problem. In Proceedings of the International Conference on Electronic Information Engineering and Computer Science (EIECS 2022), SPIE, Vol. 12602, April 2023, pp. 261-271. https://doi.org/10.1117/12.2668134

22. Huang, Z., Chen, C., & Pan, M. Multiobjective UAV Path Planning for Emergency Information Collection and Transmission, IEEE Internet of Things Journal, 2020, 7(8), 6993-7009. https://doi.org/10.1109/JIOT.2020.2979521

23. Karabulut, K., Öztop, H., Kandiller, L., Taşgetiren, M. F. Modeling and Optimization of Multiple Traveling Salesmen Problems: An Evolution Strategy Approach. Computers & Operations Research, 2021, 129, 105192. https://doi.org/10.1016/j.cor.2020.105192

24. Kyrkou, C., Theocharides, T. EmergencyNet: Efficient Aerial Image Classification for Drone-Based Emergency Monitoring Using Atrous Convolutional Feature Fusion. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2020, 13, 1687-1699. https://doi.org/10.1109/JSTARS.2020.2969809

25. Lang, Y., Gao, Y. Dream Optimization Algorithm (DOA): A Novel Metaheuristic Optimization Algorithm Inspired by Human Dreams and Its Applications to Real-World Engineering Problems. Computer Methods in Applied Mechanics and Engineering, 2025, 436, 117718. https://doi.org/10.1016/j.cma.2024.117718

26. Li, L., Zhou, S., Wang, Z. A Novel Multi-UAV Patrol Path Planning Method Based on Evolutionary Multi-Objective Optimization. In Proceedings of the 2022 4th International Conference on Intelligent Information Processing (IIP), IEEE, October 2022, pp. 332-335. https://doi.org/10.1109/IIP57348.2022.00076

27. Lin, Y., Wang, T., Wang, S. UAV-Assisted Emergency Communications: An Extended Multi-Armed Bandit Perspective. IEEE Communications Letters, 2019, 23(5), 938-941. https://doi.org/10.1109/LCOMM.2019.2906194

28. Liu, R., Wang, Y. Research on TSP Solution Based on Genetic Algorithm. In Proceedings of the 2019 IEEE/ACIS 18th International Conference on Computer and Information Science (ICIS), IEEE, June 2019, pp. 230-235. https://doi.org/10.1109/ICIS46139.2019.8940186

29. Lu, L. C., Yue, T. W. Mission-Oriented Ant-Team ACO for Min-Max MTSP. Applied Soft Computing, 2019, 76, 436-444. https://doi.org/10.1016/j.asoc.2018.11.048

30. Manna, A., Maity, S., Roy, A. A Heuristic Approach for Cluster TSP. In Recent Advances in Intelligent Information Systems and Applied Mathematics, Springer International Publishing, 2020, 43-52. https://doi.org/10.1007/978-3-030-34152-7_4

31. Mirjalili, S., Lewis, A. The Whale Optimization Algorithm. Advances in Engineering Software, 2016, 95, 51-67. https://doi.org/10.1016/j.advengsoft.2016.01.008

32. Nayak, A., Rathinam, S. Heuristics and Learning Models for Dubins MinMax Traveling Salesman Problem. Sensors, 2023, 23(14), 6432. https://doi.org/10.3390/s23146432

33. Qu, H., Zhang, W., Zhao, J., Luan, Z., Chang, C. Rapid Deployment of UAVs Based on Bandwidth Resources in Emergency Scenarios. In Proceedings of the 2020 Information Communication Technologies Conference (ICTC), IEEE, May 2020, pp. 86-90. https://doi.org/10.1109/ICTC49638.2020.9123274

34. Ren, T., Yang, J., Li, J. The Improvement of Simulated Annealing Algorithm on the Penalty Function in Multi-Agent Traveling Salesman Problem. In Proceedings of the 2021 European Symposium on Software Engineering, November 2021, pp. 142-149. https://doi.org/10.1145/3501774.3501795

35. Sánchez-García, J., Reina, D. G., Toral, S. L. A Distributed PSO-Based Exploration Algorithm for a UAV Network Assisting a Disaster Scenario. Future Generation Computer Systems, 2019, 90, 129-148. https://doi.org/10.1016/j.future.2018.07.048

36. Tang, Y. UAV Detection Based on Clustering Analysis and Improved Genetic Algorithm. In Proceedings of the 2021 International Conference on Electronic Communications, Internet of Things and Big Data (ICEIB), IEEE, December 2021, 4-9. https://doi.org/10.1109/ICEIB53692.2021.9686421

37. Wan, Y., Zhong, Y., Ma, A., Zhang, L. An Accurate UAV 3-D Path Planning Method for Disaster Emergency Response Based on an Improved Multiobjective Swarm Intelligence Algorithm. IEEE Transactions on Cybernetics, 2022, 53(4), 2658-2671. https://doi.org/10.1109/TCYB.2022.3170580

38. Wang, B., Sun, Y., Sun, Z., Nguyen, L. D., Duong, T. Q. UAV-Assisted Emergency Communications in Social IoT: A Dynamic Hypergraph Coloring Approach. IEEE Internet of Things Journal, 2020, 7(8), 7663-7677. https://doi.org/10.1109/JIOT.2020.2988445

39. Wang, M., Ma, T., Li, G., Zhai, X., Qiao, S. Ant Colony Optimization with an Improved Pheromone Model for Solving MTSP with Capacity and Time Window Constraint. IEEE Access, 2020, 8, 106872-106879. https://doi.org/10.1109/ACCESS.2020.3000501

40. Wang, Y., Shi, Y., Liu, Y. Research on Improved Genetic Simulated Annealing Algorithm for Multi-UAV Cooperative Task Allocation. In Journal of Physics: Conference Series, IOP Publishing, 2022, 2246(1), 012081. https://doi.org/10.1088/1742-6596/2246/1/012081

41. Wang, Z., Luo, Q., Zhou, Y. Hybrid Metaheuristic Algorithm Using Butterfly and Flower Pollination Base on Mutualism Mechanism for Global Optimization Problems. Engineering with Computers, 2021, 37, 3665-3698. https://doi.org/10.1007/s00366-020-01025-8

42. Wu, X., Zhang, S., Xiao, W., Yin, Y. The Exploration/Exploitation Tradeoff in Whale Optimization Algorithm. IEEE Access, 2019, 7, 125919-125928. https://doi.org/10.1109/ACCESS.2019.2938857

43. Zhong, X. On the Approximation Ratio of the 3-Opt Algorithm for the (1, 2)-TSP. Operations Research Letters, 2021, 49(4), 515-521. https://doi.org/10.1016/j.orl.2021.05.012