

Article

A Tent-Lévy-Based Seagull Optimization Algorithm for the Multi-UAV Collaborative Task Allocation Problem

Zhao Zhou, Huan Liu * , Yongqiang Dai and Lijing Qin

School of Information Science and Technology, Gansu Agricultural University, Lanzhou 730070, China; zzzzzz20230206@163.com (Z.Z.); dyq@gsau.edu.cn (Y.D.); qinlij@gsau.edu.cn (L.Q.)

* Correspondence: liuh1204@hotmail.com

Abstract: With the rapid advancement of unmanned aerial vehicle technology, the extensive application of multiple unmanned aerial vehicle systems in agriculture has led to significant innovations and benefits. Addressing the challenge of task allocation for multiple unmanned aerial vehicles, the primary objective is to minimize the total time required for unmanned aerial vehicles to return to their starting point after task completion. To tackle this issue, a mathematical model for the multi-constrained multiple unmanned aerial vehicle collaborative task allocation problem is developed. To efficiently solve this model, we propose an enhanced Seagull Optimization Algorithm, which integrates the Tent chaotic mapping strategy and the Lévy flight strategy. The Tent chaotic mapping helps the algorithm avoid becoming trapped in local optima, while the Lévy flight strategy, employed during the seagull attack phase, enhances the algorithm's diversity and its ability to escape local optima. Additionally, the spiral coefficient is refined to balance the coordination between global and local searches. Simulation experiments demonstrate that the proposed algorithm can swiftly and effectively identify a reasonable task allocation scheme for solving the multi-constrained multi-UAV collaborative task allocation problem.

Keywords: multiple unmanned aerial vehicles; collaborative task allocation; tent chaotic mapping; Lévy flight; seagull optimization algorithm



Citation: Zhou, Z.; Liu, H.; Dai, Y.; Qin, L. A Tent-Lévy-Based Seagull Optimization Algorithm for the Multi-UAV Collaborative Task Allocation Problem. *Appl. Sci.* **2024**, *14*, 5398. <https://doi.org/10.3390/app14135398>

Academic Editors: Giovanni Boschetti, Matteo Bottin and Riccardo Minto

Received: 7 May 2024

Revised: 17 June 2024

Accepted: 18 June 2024

Published: 21 June 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Unmanned aerial vehicle (UAV) systems are increasingly valued for their diverse applications in agriculture, aerospace, and military domains [1,2]. Specifically, in agriculture, UAVs have revitalized modern farming practices [3]. Their extensive deployment in this sector not only enhances production efficiency and agricultural product quality but also introduces more intelligent and efficient solutions for various agricultural operations [4]. To overcome the limitations of single UAVs in handling complex tasks, research and practical applications are progressively adopting multi-UAV collaborative approaches to achieve greater overall benefits.

As smart agriculture continues to advance, a single UAV often faces limitations in executing complex agricultural tasks. Consequently, the intricate and diverse nature of agricultural production chains underscores the significance of deploying multi-UAV systems in this field [5]. Addressing the task allocation challenge for multi-UAV systems involves assigning multiple UAVs to collaboratively perform various tasks while adhering to multiple constraints to minimize overall system costs [6]. Thus, the multi-constrained multi-UAV collaborative task allocation problem (MCMUAVCTAP) is classified as a nondeterministic polynomial (NP) problem, necessitating effective management of its inherent complexity. To accurately control multi-UAVs and efficiently formulate task allocation plans, solving the MCMUAVCTAP has become a pressing issue in the coordinated control of multi-UAVs [7].

This study integrates the Tent chaotic mapping strategy and the Lévy flight strategy into the Seagull Optimization Algorithm (SOA) to enhance search diversity and accelerate

convergence. Furthermore, the spiral coefficient within the SOA is refined and applied to the MCMUAVCTAP. To evaluate the effectiveness of the proposed Tent–Lévy Improved Seagull Optimization Algorithm (TLISOA), it was compared with various task allocation algorithms using real agricultural scenarios. The findings confirm the effectiveness of TLISOA in addressing the MCMUAVCTAP. This research aims to support the resolution of MCMUAVCTAP in agriculture, thereby fostering broader application and development of UAV technology in this sector. The main contributions of this study are summarized as follows: (1) establishing a single-objective optimization model for the MCMUAVCTAP; (2) proposing the TLISOA to address the MCMUAVCTAP; (3) conducting simulation experiments in agricultural contexts, demonstrating that the algorithm enhances the response speed and efficiency of multi-UAV systems.

The structure of this paper is as follows: Section 2 reviews the current state of both domestic and international research on the topic. Section 3 introduces the single-objective optimization model developed for the MCMUAVCTAP. Section 4 describes the TLISOA. Section 5 presents simulation experiments comparing the proposed TLISOA with five other algorithms in multi-UAV collaborative task allocation for rice pest monitoring scenarios and provides a comparative analysis. Finally, Section 6 concludes this paper.

2. Related Work

The MCMUAVCTAP has garnered significant attention from researchers worldwide, resulting in numerous notable advancements. Reference [8] introduced a Clone Selection Algorithm (CSA) designed to optimize four objectives in multi-UAV task allocation: maximizing the number of successfully allocated tasks, maximizing the benefits of executing tasks, minimizing resource costs, and minimizing time costs. Experimental results demonstrated that CSA outperforms the genetic algorithm in multi-objective UAV task allocation. Reference [9] proposed three complementary algorithm variants—Allocation Loop (AL), Sorting and Allocation Loop (SAL), and Limit and Allocation Loop (LAL)—to enhance task assignment performance. Reference [10] developed an Improved Genetic Algorithm (IGA) featuring a logic-based unlocking mechanism for crossover and mutation operations. Simulations indicated that IGA effectively addresses the MCMUAVCTAP. Reference [11] applied dynamic decision theory (DT) to establish a multi-objective task assignment optimization model and introduced an Adaptive Weighted Particle Swarm Optimization (AWPSO) algorithm to solve it. The algorithm's environmental change detection and response strategy ensures population convergence and diversity. Experimental results confirmed the algorithm's strong convergence, diversity, and high decision quality. Reference [12] proposed an improved Wolf Pack Algorithm (WPA), which intelligently modulates the randomness in the traditional wolf pack search to solve the MCMUAVCTAP. Reference [13] developed a dynamic task assignment method for multi-UAVs using a combination of ant colony, bat, and gray wolf algorithms, which significantly reduces computation time and shortens UAV search duration and distance. Reference [14] introduced an unbalanced target allocation model for multi-UAVs, based on situational assessment, solved using a Hungarian fusion genetic algorithm. Reference [15] proposed an improved distributed Ant Colony Optimization (ACO) algorithm incorporating Q-learning to generate task sequences for each UAV. The algorithm uses a colony disorientation strategy to expand the search range and a search transition strategy to prevent premature convergence. Simulation experiments validated the practicality and reliability of this approach. Finally, Reference [16] presented a Competitive Learning Pigeon-Inspired Optimization (CLPIO) algorithm for cooperative dynamic combat problems, integrating distributed swarm motion and centralized target allocation. The algorithm also includes a threshold trigger strategy to switch between sub-tasks, and a comparative analysis demonstrated its superior performance in solving mixed Nash equilibrium problems.

Given that the SOA offers unique advantages over other heuristic algorithms in terms of adaptivity and parallelism, Reference [17] proposed an optimal tidal current calculation method based on the inverse variance SOA. This method balances global search and local

development capabilities, addressing the issues of poor global search ability and slow convergence speed in SOA. Reference [18] introduced an improved SOA, incorporating a natural selection mechanism to expedite the search for optimal solutions. Additionally, it combines the unequal division method to solve dynamic optimization problems, with its feasibility demonstrated through a chemical dynamic optimization example. Reference [19] presented a new Evolutionary Multi-objective Seagull Optimization Algorithm (EMoSQA), which produced highly convergent Pareto optimal solutions in engineering design problems. Reference [20] proposed an initialization method based on Gaussian distribution and a Gaussian distribution-based dimensional sequential variation operator. By integrating non-uniform control vector parameterization with the improved SOA, simulation results indicated that the algorithm achieves similar or even higher solution accuracy.

3. Problem Formulation

3.1. Scenario Description

This paper studies the MCMUAVCTAP in agricultural contexts. The problem is described as follows: In a rice planting area, n UAVs are tasked with monitoring m disease and insect pest monitoring points. All UAVs must perform these tasks. The UAVs depart simultaneously from the agricultural command center, visit and complete their assigned monitoring points in sequence, and finally return to the command center. Each monitoring task requires a different execution time. The objective is for all UAVs to complete the monitoring tasks as quickly as possible and return to the command center, while adhering to practical constraints, that is, in the optimal UAV allocation plan, the time it takes for the latest UAV to arrive at the starting point. The schematic diagram is shown in Figure 1.

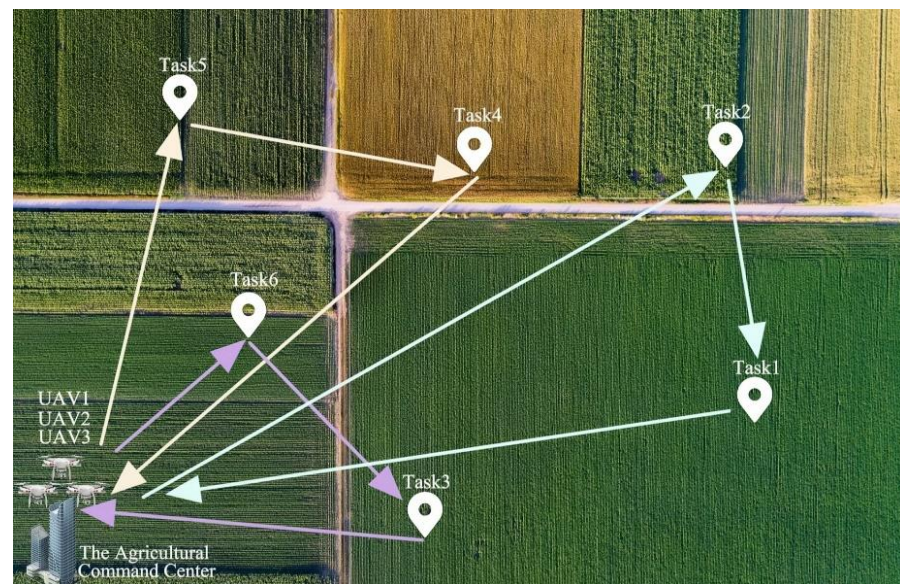


Figure 1. Schematic diagram of task assignment.

In this example, three UAVs—UAV1, UAV2, and UAV3—are required to complete pest monitoring tasks for Task 1 to Task 6, aiming to find the optimal task allocation scheme while satisfying practical constraints. UAV1, UAV2, and UAV3 start simultaneously from the same agricultural command center, perform their assigned tasks, and return to the command center. The total time for the allocation scheme is determined by the UAV that takes the longest to return. Therefore, the optimal scheme minimizes this return time.

3.2. Parameter Definitions

Suppose the set of UAVs is $UAV = \{U_1, U_2, \dots, U_i, \dots, U_n\}$, the initial coordinates of the UAVs are $P = (\alpha, \gamma)$, and the flying speed of the UAVs is V . Each UAV has sufficient power to complete its assigned sequence of pest and disease monitoring tasks

and return to the agricultural command center. The set of tasks that all UAVs need to perform is $Task = \{Task_1, Task_2, \dots, Task_j, \dots, Task_m\}$, the position coordinates that each task is abstracted into are $Q_m = (o, q)$, and the execution times are represented by $Time = \{Time_1, Time_2, \dots, Time_m\}$. The sequence of tasks performed by UAV U_i can be denoted as $Seq^i = \{M_1^i, M_2^i, \dots, M_{N_i}^i\}$, and N_i represents the number of tasks assigned to UAV U_i . Define C_{ij} as a decision variable that indicates the allocation relationship between UAV U_i and task $Task_j$. When task $Task_j$ is assigned to UAV U_i , $C_{ij} = 1$; otherwise, $C_{ij} = 0$.

3.3. Restrictive Conditions

There are the following constraints in the allocation of agricultural tasks:

1. Collaborative constraints for multi-UAV system: Each task point within the pest monitoring set must be completed only once. Consequently, no more than one UAV can perform an operation on the same task point. Formally, this constraint can be expressed as

$$Seq^a \cap Seq^b = \emptyset \quad a, b \in \{1, 2, \dots, n\} \text{ and } a \neq b \quad (1)$$

2. Task completion constraint: All pest monitoring tasks must be completed. This requirement can be formally represented as

$$Seq^1 \cup Seq^2 \cup \dots \cup Seq^n = \{Task_1, Task_2, \dots, Task_j, \dots, Task_m\} \quad (2)$$

3. Maximum UAV deployment constraint: All UAVs must be actively engaged in pest monitoring tasks. Thus, no UAV should remain idle at the agricultural command center. Formally, this constraint can be expressed as

$$Seq^i \neq \emptyset \quad i \in \{1, 2, \dots, n\} \quad (3)$$

3.4. Objective Function

In addressing this type of problem, the voyage or flight time is typically a crucial metric for decision-making. In our study, we select the total time taken by the last UAV to complete its monitoring tasks and return to the agricultural command center. This duration represents the overall time spent for the entire task allocation scheme after all pest monitoring tasks have been completed. The optimal task allocation scheme is defined as the one that minimizes this total time. Thus, the execution time for UAV U_i in a given allocation scheme is represented as

$$T_i = T_{tot_dis} + T_{tot_task} \quad i \in \{1, 2, \dots, n\} \quad (4)$$

$$T_{tot_dis} = T_0^{M_1^i} + \sum_{M_1^i \in Seq^i} T_{M_k^i}^{M_{k+1}^i} + T_{M_{N_i}^i}^0 \quad (5)$$

$$T_{tot_task} = \sum_{k=1}^{N_i} t_k^i \quad (6)$$

where T_{tot_dis} in the equation denotes the time taken by UAV U_i to execute the voyage it has passed through in the sequence of agricultural tasks to which it has been assigned, and T_{tot_task} denotes the sum of the execution lengths required by UAV U_i to execute the individual tasks in the sequence of agricultural tasks to which it has been assigned. $T_0^{M_1^i}$ in Equation (5) denotes the time required for UAV U_i to fly from the start point to the first task point in the assigned sequence, $\sum_{M_1^i \in Seq^i} T_{M_k^i}^{M_{k+1}^i}$ denotes the time required for UAV U_i to fly from the k th task point to the $k + 1$ st task in the assigned sequence, and $T_{M_{N_i}^i}^0$ denotes the time required for the UAV U_i to fly from the last task point in the allocation sequence to the start point.

In summary, the longest execution time of a UAV sequence in a given distribution scheme is expressed as

$$T_{max} = \max(T_i) \quad U_i \in U \quad (7)$$

Then, the optimization objective can be expressed as

$$\min(T_{max}) \quad (8)$$

4. An Improved Seagull Optimization Algorithm Incorporating Tent Chaos Mapping and Lévy Flight

Compared with other chaotic mappings, Tent chaotic mapping [21] shows great advantages in ergodicity. Ergodicity means that the algorithm can comprehensively explore the solution space, thereby increasing the possibility of finding the global optimal solution. Secondly, the values of the Tent chaos map are relatively evenly distributed in the [0, 1] interval, which is conducive to maintaining the diversity of the population when generating the initial population, and avoids concentrating on a certain sub-region of the solution space in the early stages of the algorithm, thereby improving search efficiency. Moreover, the iteration speed of Tent chaos mapping is fast, which means that a high-quality initial population can be generated in a short time and accelerate the convergence process of the algorithm. Therefore, this study incorporates Tent chaos mapping into the standard SOA. The introduction of this method aims to effectively deal with the problem that the algorithm is prone to falling into premature and local optimal values, so as to significantly improve the convergence speed and optimization accuracy of the algorithm. When the standard SOA updates the position of candidate solutions, it may cause the search process to focus on certain areas of the search space and ignore other areas, especially areas far away from the current optimal solution. This central tendency may slow down the convergence of the algorithm or even cause the algorithm to fall into a local optimal solution. Therefore, the Lévy flight strategy [22] was also introduced during the seagull attack phase. This strategy uses the Lévy distribution to generate step sizes. This distribution is characterized by its long tail characteristics and can generate large step sizes under low-probability conditions, which means that the algorithm is more likely to skip those local optimal solutions, thereby increasing the algorithm's exploration capabilities. In this way, the Lévy flight strategy can help the SOA cover the solution space more extensively during the search process, improve search efficiency, and accelerate convergence to the global optimal solution. At the same time, the spiral coefficient in the SOA is adjusted, to more effectively balance the coordination relationship between global and local searches. In view of the above improvement scheme, the flowchart of the algorithm is shown in Figure 2, where the improved parts are highlighted in this figure.

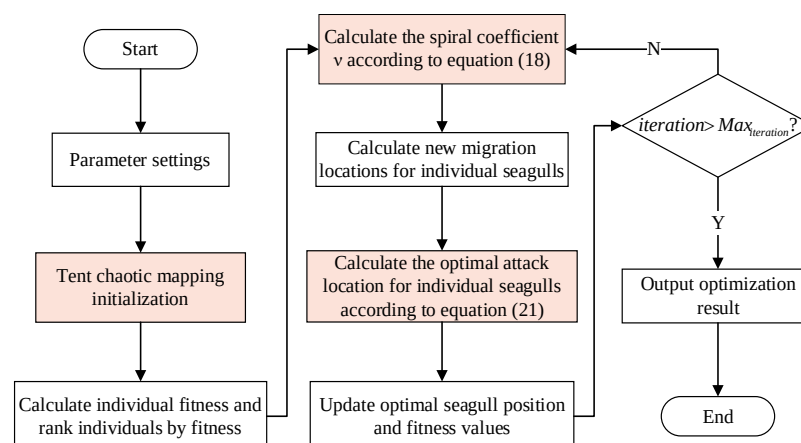


Figure 2. Improved Seagull Optimization Algorithm flowchart.

4.1. Seagull Optimization Algorithm

The Seagull Optimization Algorithm (SOA) is a novel meta-heuristic intelligent optimization algorithm proposed by Gaurav Dhiman et al. in 2019. It simulates the migratory and aggressive behavior of seagulls to solve optimization problems. Migration refers to the seasonal movement of animals in search of abundant food sources to acquire sufficient energy. During migration, each seagull occupies a distinct location to prevent collisions. Within a flock, seagulls can adjust their positions towards the optimal direction. During the attack phase, seagulls exhibit a spiral movement pattern. A schematic diagram illustrating the migration and attack behavior of seagulls is presented in Figure 3.

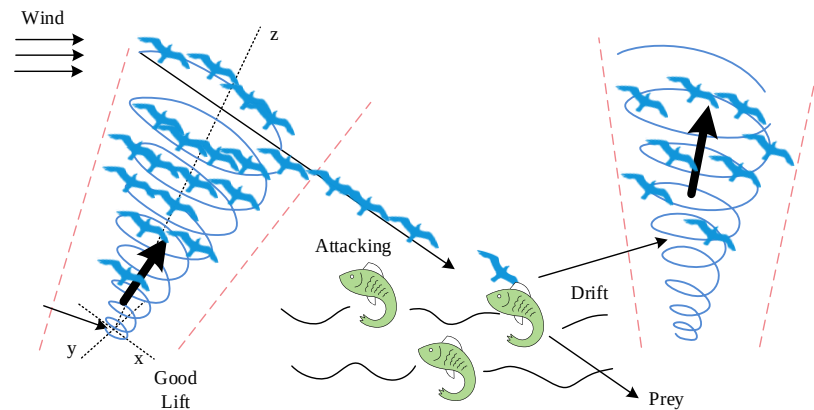


Figure 3. Migration and attacking behaviors of seagulls.

4.1.1. Migration Phase of Seagulls

In the migration process of gulls, the SOA simulates how the seagull colony moves from one location to another. At this stage, the seagulls should fulfill the condition of collision avoidance. In order to avoid collision with other neighboring gulls, the algorithm uses an additional variable A to calculate the new position of the seagulls, i.e.,

$$C_s(t) = A \times P_s(t) \quad (9)$$

where $C_s(t)$ denotes the new position that does not have positional conflict with other seagulls; $P_s(t)$ is the current position of the seagulls; t denotes the current number of iterations; and A denotes the movement behavior of the seagulls in the given search space, which can be expressed as

$$A = f_c - (t \times (f_c / \text{Max}_{\text{iteration}})) \quad (10)$$

where f_c indicates that the frequency of change in variable A can be controlled, and its value decreases linearly from 2 to 0; $\text{Max}_{\text{iteration}}$ indicates the maximum number of iterations; and t indicates the current number of iterations.

During migration, after avoiding overlapping with other seagulls' positions, the seagulls will first calculate the direction of the best position and move to the best position as shown in Equation (11).

$$M_s(t) = B \times (P_{bs}(t) - P_s(t)) \quad (11)$$

where $M_s(t)$ denotes the direction of the best position; $P_{bs}(t)$ denotes the current best position of the seagulls; $P_s(t)$ denotes the current position of the seagulls; and B is the random number that balances the global search and the local search, i.e.,

$$B = 2 \times A^2 \times r_d \quad (12)$$

where r_d is a random number in the interval $[0, 1]$.

After obtaining the direction of the optimal position, the seagulls will move towards the optimal position and reach the new position. The process is represented by Equation (13).

$$D_s(t) = |C_s(t) + M_s(t)| \quad (13)$$

where $D_s(t)$ denotes the new position of the seagulls; $C_s(t)$ denotes the position where there is no positional conflict with other seagulls; and $M_s(t)$ denotes the direction in which the best position is taken.

4.1.2. Seagulls Attacking Prey Stage

Seagulls can constantly change the angle and speed of attack during migration while using their wings and weight to maintain altitude, and when seagulls attack their prey, they engage in spiral motion in the air. Describe the motion behavior of seagulls in the x , y , and z plane as

$$\begin{cases} x = r \times \cos\theta \\ y = r \times \sin\theta \\ z = r \times \theta \\ r = u \times e^{\theta v} \end{cases} \quad (14)$$

where r is the radius of each helix; θ is a random angle value in the interval; u and v are constants associated with the shape of the helix; and e is the base of the natural logarithm.

When $u = 1$, $v = 0.1$, θ increases from 0 to 2π , and a coordinate system is established with x , y , and z ; the trajectory of a seagull is shown in the Figure 4.

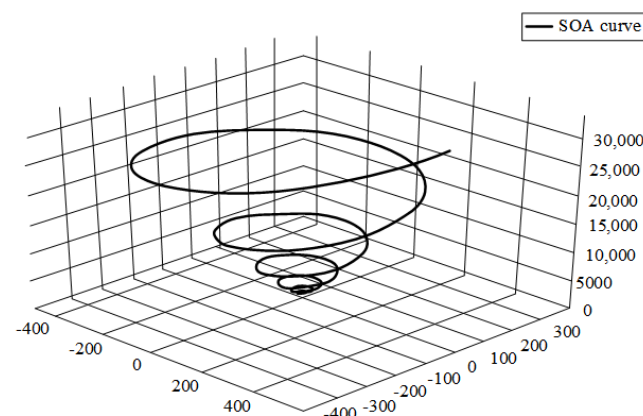


Figure 4. Seagull movement trajectory.

The position of the seagulls after attacking their prey is represented by Equation (15).

$$P_s(t) = D_s(t) \times x \times y \times z + P_{bs}(t) \quad (15)$$

where $P_s(t)$ denotes the position of the seagulls after attacking the prey; $P_{bs}(t)$ denotes the current optimal position of the seagulls.

4.2. Tent Chaotic Mapping for Population Initialization

During the execution process of the SOA, the random generation of the initial population stands as a pivotal step. To enhance the diversity of the initial population and prevent premature convergence or local optima, chaotic mapping can be employed for population initialization. Chaotic mapping exhibits superior properties in traversal and randomness compared to traditional random generation methods. Specifically, Tent chaotic mapping, a classical approach, is widely adopted in the initial population generation of population intelligence algorithms due to its randomness, ergodicity, and flexibility. This method not only improves the algorithm's convergence speed and accuracy but also ensures a more

uniform distribution of initial solutions in the search space, thereby enhancing the SOA's performance.

The functional expression for the Tent chaotic mapping is shown in Equation (16).

$$x_{t+1}^{\xi} = \begin{cases} 2x_t^{\xi}, & 0 \leq x_t^{\xi} \leq 1/2 \\ 2(1 - x_t^{\xi}), & 1/2 < x_t^{\xi} \leq 1 \end{cases} \quad (16)$$

where $\xi = 1, 2, \dots, N$ denotes the number of populations; $t = 1, 2, \dots, h$ denotes the spatial dimension.

Selecting h initial values according to Equation (16), h chaotic sequences x_t^{ξ} are obtained, and these sequences are mapped to the search space through Equation (17), from which the initial population is generated.

$$y_t^i = c_i + (d_i - c_i)x_t^i \quad (17)$$

where c_i and d_i are the lower and upper bounds of the x_t^i search, respectively.

4.3. The Adaptive Spiral Coefficient ν

The spiral coefficient ν in the seagull attack phase of the SOA is a constant ($\nu = 1$) that cannot be adaptively adjusted as the number of iterations increases, which will make the gulls fly with a larger spiral radius in the later phase, which will cause the algorithm to hover near the optimal solution, and it cannot be converged quickly. Therefore, the spiral coefficient ν is improved in this paper:

$$\nu = 1 - 2 \times (\text{iteration} / \text{Max}_{\text{iteration}}) \quad (18)$$

where iteration is the current iteration number and $\text{Max}_{\text{iteration}}$ is the maximum iteration number.

As the number of iterations increases, the improved spiral coefficient ν decreases linearly from 1 to -1 . The spiral radius of the seagull is formulated as $r = u \times e^{\theta\nu}$, so the spiral radius of the seagull's flight is larger in the early stage, which enables the algorithm to explore the approximate location of the optimal solution more efficiently, while in the later stage, the radius of the seagull's flight is smaller, which enables the algorithm to find the exact location of the optimal solution more quickly.

4.4. Lévy Flight

Lévy flight is a stochastic process that emulates flight using a probability distribution of stride lengths with heavy tails. Typically, it involves predominantly short steps interspersed with occasional long strides. This characteristic renders Lévy flight well suited for the exploration phase in population-based optimization algorithms, mirroring the alternation between global and local search processes within the population. In the context of the SOA, incorporating Lévy flights during the search for optimal attack positions effectively broadens the search scope, facilitating escape from local optima. The step sizes of Lévy flights are randomly generated and simulated using Mantegna's algorithm, for which the relevant mathematical formulas are provided below:

$$\text{Levy}(\beta) \sim \frac{\mu}{|\eta|^{1/\beta}} \quad (19)$$

In Equation (19), $Levy(\beta)$ denotes the *Levy* random search path; $\beta = 1.5$; and both μ and η obey Gaussian distributions as follows:

$$\begin{cases} \mu \sim N(0, \sigma_\mu^2), \eta \sim N(0, \sigma_\eta^2) \\ \sigma_\mu = \left\{ \frac{\Gamma(1+\beta) \times \sin(\pi \times \frac{\beta}{2})}{\Gamma[\frac{1+\beta}{2} \times \beta \times 2^{\frac{\beta-1}{2}}]} \right\}^{1/\beta} \\ \sigma_\eta = 1 \end{cases} \quad (20)$$

where σ_μ and σ_η are the variances of μ and η , respectively, and Γ is the gamma function.

Adding the Lévy flight strategy to the seagull attack phase improves the formula as follows:

$$P_s(t) = (D_s(t) \times x \times y \times z) \times Levy(\beta) + P_{bs}(t) \quad (21)$$

5. Simulation Experiments

5.1. Experimental Design

In order to verify the effectiveness of the TLISOA for solving the MCMUAVCTAP, simulation experiments are conducted for two experimental scenarios to verify the superiority of the algorithm for solving the MCMUAVCTAP. In experimental scenario 1, the UAVs are set to be six, the total number of agricultural tasks to be performed is 20, and the agricultural tasks are performed in a 2 km $\times$ 2 km farmland. The initial position of the UAVs is (0, 0), and the position range of each agricultural task is from (100, 100) to (2000, 2000) (unit m); 20 points are randomly selected as the positions of the 20 agricultural tasks within this range. The length of time required for each agricultural task is in the range of (5, 15) (unit min), and the speed of all UAVs is constant at 300 m/min. Six UAVs are set up in experimental scenario 2, and the total number of agricultural tasks to be performed is 50, and the rest of the conditions are consistent with experimental scenario 1. The multi-UAV cooperative task allocation algorithms based on CAM-GA [23], A-QCDPSO [24], SOA [25], PSO-AWOA [26], CESMA [27], and TLISOA are selected for comparison, and each algorithm's population size is $population = 30$, and the maximum iteration is $Max_{iteration} = 500$; in order to reduce the impact of randomness in the algorithm, each algorithm is run independently 20 times, and the simulation environment is shown in Table 1.

Table 1. Simulation environment.

Item	Description
Processor	Intel® Core(TM) i5-8300H CPU @ 2.30GHz 2.30 GHz (Intel, Santa Clara, CA, USA)
RAM	8 GB
OS	Windows 11 (64-bit)
Python version	Python 3.9

5.2. Sensitivity Analysis of Parameters

The performance of the proposed TLISOA is highly dependent on the parameter settings within the algorithm, specifically f_c , u , and v . Different parameter configurations can lead to significantly different optimization results, affecting both the solution quality and computational efficiency. In the context of multi-UAV task allocation, optimization algorithms must operate efficiently under complex task constraints and multi-objective requirements to achieve rational task distribution and optimal resource utilization. Incorrect parameter settings may cause the algorithm to become trapped in local optima during the search process or incur unnecessary computational costs, thereby reducing the practical effectiveness of the algorithm. Given that TLISOA incorporates certain improvements to the spiral coefficient v , it is necessary to conduct a sensitivity analysis on the remaining

two parameters, f_c and u . This analysis will be performed using scenario 1, with the experimental design detailed in Table 2, where each group of experiments is run independently for 20 times, and the experimental results are shown in Figure 5. The algorithm exhibits superior optimization performance and faster convergence when parameters are set to $f_c = 2$ and $u = 1$; therefore, these values are adopted (Table 3).

Table 2. Example of parameter sensitivity experiment.

Experiment ID	f_c	u
1	3	1.5
2	3	1
3	3	0.5
4	2	1.5
5	2	1
6	2	0.5
7	1	1.5
8	1	1
9	1	0.5

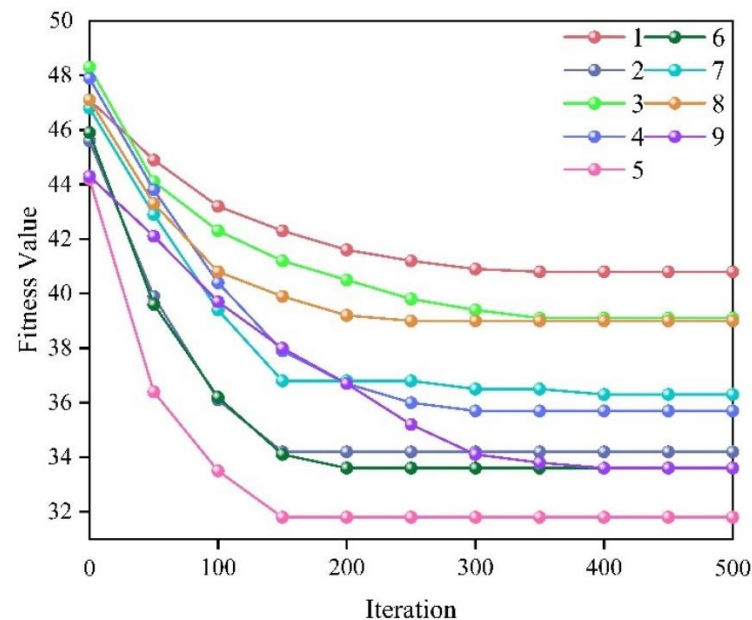


Figure 5. Sensitivity analysis of parameters.

Table 3. Parameter settings of the algorithms.

Algorithms	Parameters
TLISOA	$f_c = 2; u = 1; v = 1 - 2 \times (\text{iteration} / \text{Max}_{\text{iteration}})$
A-QCDPSO	$\omega = 0.6; c1 = c2 = 1.5$
CAM-GA	$p1 = 0.65, p2 = 0.02$
SOA	$f_c = 2; u = 1; v = 1$
PSO-AWOA	$b = 1; c1 = 2; k = 2.2; \omega = \left[1 - \sin\left(\frac{\pi}{2} \times \frac{\text{iteration}}{\text{Max}_{\text{iteration}}}\right)\right]^k$
CESMA	$z = 0.03$

5.3. Comparison of Algorithm Performance in Different Scenarios

The convergence curves of the six algorithms SOA, TLISOA, CAM-GA, A-QCDPSO, PSO-AWOA, and CESMA for MCMUAVCTAP were obtained by executing the programs in PyCharm under the two experimental scenarios, as shown in Figures 6 and 7. Tables 4 and 5

list the optimal task allocation schemes designed by these algorithms for the two experimental scenarios. Tables 6 and 7 present the algorithmic performance comparison between TLISOA and the other five algorithms under the two experimental scenarios in terms of optimal, mean, median, worst, standard deviation, and average convergence time. Box plots of the results from 20 independent runs of each of the six algorithms are provided in Figures 8 and 9, respectively.

Figures 6 and 7 reveal a notable discrepancy in the iteration speeds among the six algorithms, all of which gradually converge toward the optimal value throughout the iteration process. Notably, TLISOA exhibits a swifter iteration pace. This can be attributed to its larger spiral coefficient during the early stages, facilitating rapid optimization progress. Additionally, the combined effect of population initialization via Tent chaotic mapping and Lévy flight enables TLISOA to swiftly converge to the global optimal solution without succumbing to local optima. Compared to the other algorithms, TLISOA demonstrates superior fitness values within the same number of iterations, underscoring its enhanced global search capability and faster convergence rate. Consequently, TLISOA yields optimization results more aligned with the multi-UAV task allocation requirements.

Tables 6 and 7, and Figures 8 and 9, show the excellent performance of TLISOA compared to the other five algorithms in two different experimental scenarios. Although the standard deviation of TLISOA slightly lags behind that of PSO-AWOA, TLISOA's optimality finding ability is more significantly improved compared to the other five algorithms in terms of the optimal, mean, median, and worst values. In addition, TLISOA has a shorter average convergence time, which suggests that it has the ability to find a better allocation scheme in a shorter time. In conclusion, the proposed algorithm has significant advantages in handling MCMUAVCTAP.

Table 4. Task allocation results of experimental scenario 1.

UAV	Algorithms					
	SOA	TLISOA	CAM-GA	A-QCDPSO	PSO-AWOA	CESMA
U_1	5-6-16-19	4-13-3	19-15-18	14-16-18-9-2	4-3-20-7	6-14-15-7
U_2	9-8-17-3	10-5-7	5-1	6-1-8-7	14-1-11-9	12-18-8
U_3	15-13-7-12	9-19-8-20	3-12-18-4	13-19-3-11	13-6-16-12	20-3-9
U_4	11-14-18	15-1-14-12	14-7-16-2	4-20-15	10-5-18	2-1-13-11
U_5	2-4-1	2-6-16	17-13-8-20	12-10	2-19-8	4-17-16
U_6	10-20	17-11-18	6-2-12	17-5	17-15	5-19-10

Table 5. Task allocation results of experimental scenario 2.

UAV	Algorithms					
	SOA	TLISOA	CAM-GA	A-QCDPSO	PSO-AWOA	CESMA
U_1	32-7-34-44-33-16-37-31-50	28-6-45-32-50-22-48-37-4	4-20-29-19-47-38-45-49-30-31	23-14-16-25-30-32-50-38-31	45-18-28-2-32-16-43-15-14	47-15-26-31-3-33-16-38-14-46
U_2	45-5-48-15-35-38-9-8-24	1-18-27-33-24-25-17-2-44	39-41-14-9-28-7-35-11-48	7-41-3-37-27-12-33-45-6-10-29-18	39-4-20-47-25-6-37	1-23-11-25-41-49-39-43-34-9
U_3	18-25-1-20-17-40-27-19-49	47-30-11-46-19-14-13-16	23-1-43-24-34-3-6-10-5	24-1-20-43-44-48	23-3-29-1-27-19-44-46-9	19-28-21-42-17-27-6-35-12-30
U_4	41-30-13-23-43-6-3-28-46-39	42-9-49-10-12-23-43-40	8-42-27-33-37-15-50-21	42-5-2-26-28-13	48-21-11-26-31-17-50-35-41	24-36-45-22-10-32-20-2
U_5	36-26-2-4-22-11-47	29-39-8-35-7-5-38-41	16-32-18-25-22-44-36-12	34-21-4-49-22-19-36-8-17-39	36-42-8-38-7-24-40-5-22	40-37-4-8-50-18
U_6	10-14-42-21-12-29	31-21-36-20-3-34-15-26	2-40-26-13-17-46	47-35-11-46-9-40-15	49-10-13-30-33-34-12	13-5-29-44-7-48

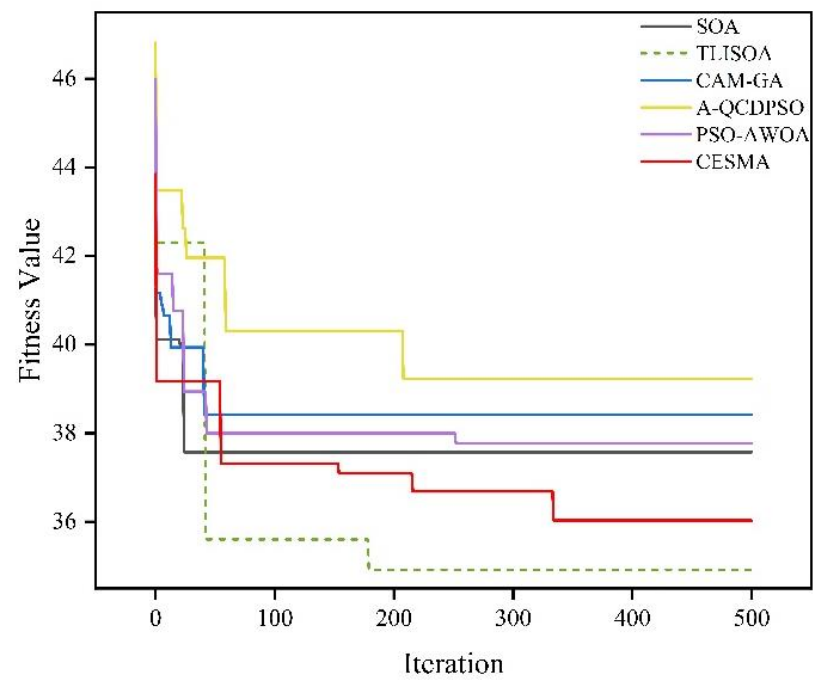


Figure 6. Algorithms' fitness curve in experimental scenario 1.

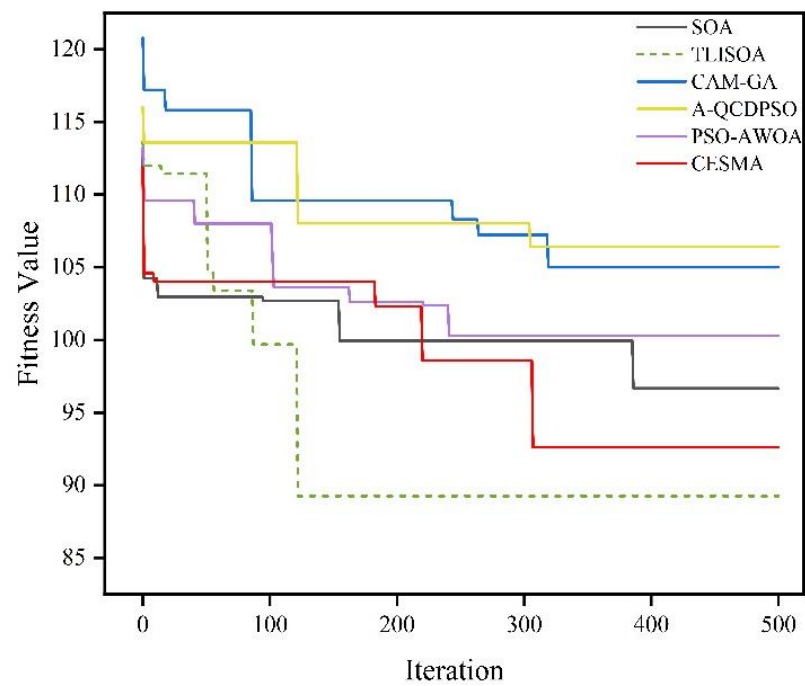


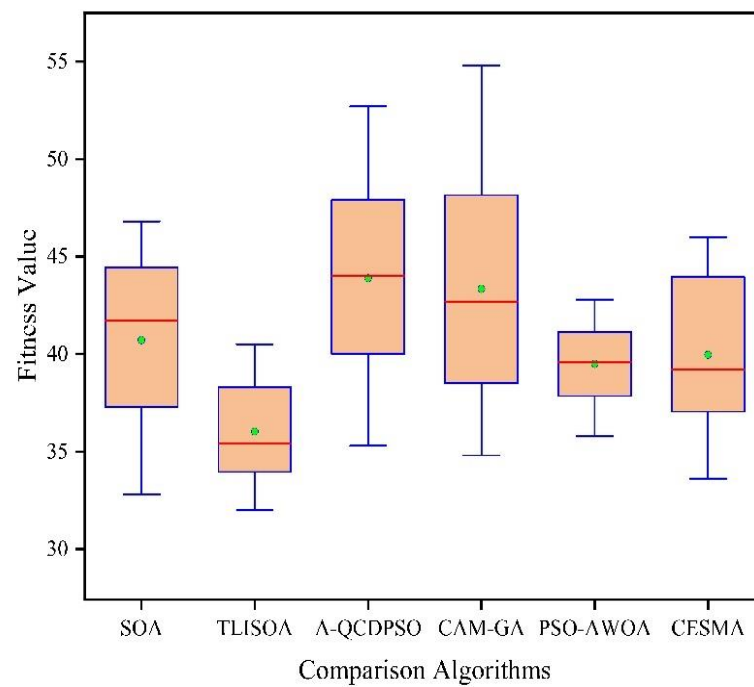
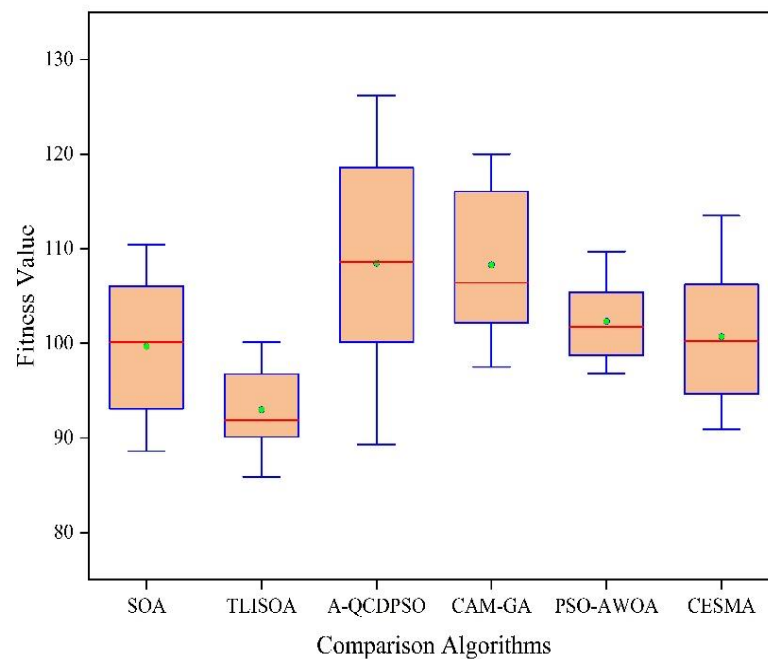
Figure 7. Algorithms' fitness curve in experimental scenario 2.

Table 6. Performance comparison of algorithms for scenario 1.

TLISOA vs.	Optimum Value	Average Value	Median Value	Worst Value	Standard Deviation	Average Convergence Time/s
SOA	2.44%	11.55%	15.11%	13.46%	42.22%	12.24%
CAM-GA	8.05%	16.86%	17.10%	26.09%	55.17%	13.06%
A-QCDPSO	9.35%	18.00%	19.55%	23.15%	49.02%	0.00%
PSO-AWOA	10.61%	10.89%	12.16%	10.20%	−30.00%	11.07%
CESMA	4.76%	9.72%	10.61%	5.37%	33.33%	15.73%

Table 7. Performance comparison of algorithms for scenario 2.

TLISOA vs.	Optimum Value	Average Value	Median Value	Worst Value	Standard Deviation	Average Convergence Time/s
SOA	3.05%	6.72%	8.19%	9.33%	41.67%	13.76%
CAM-GA	11.90%	14.13%	13.63%	16.58%	46.15%	13.30%
A-QCDPSO	3.81%	14.29%	15.38%	20.68%	61.47%	−1.24%
PSO-AWOA	11.26%	9.09%	9.64%	8.75%	−5.00%	11.25%
CESMA	5.50%	7.65%	8.28%	11.81%	37.31%	14.96%

**Figure 8.** Box plot corresponding to statistical data in experiment scenario 1.**Figure 9.** Box plot corresponding to statistical data in experiment scenario 2.

6. Conclusions

This paper delves into the multi-channel multi-UAV cooperative task allocation problem (MCMUAVCTAP) within agricultural contexts. To align with real-world agricultural operations, we formulate a corresponding problem model, considering varying execution durations for different agricultural monitoring tasks. Subsequently, we propose an enhanced Seagull Optimization Algorithm (TLISOA) that integrates Tent chaotic mapping and Lévy flight strategies, along with an improved spiral coefficient. Through multiple sets of simulation experiments, we demonstrate the algorithm's efficacy in addressing MCMUAVCTAP, particularly in scenarios with extensive agricultural tasks. The comparative analysis with alternative optimization algorithms reveals TLISOA's superior ability to identify optimal solutions, faster convergence rates, and enhanced stability. Future research directions involve extending the algorithm's application to more intricate multi-UAV task allocation problems, incorporating factors such as farmland environment and terrain characteristics.

Author Contributions: Investigation, L.Q.; Writing—original draft, Z.Z.; Writing—review & editing, H.L. and Y.D.; Z.Z. and H.L. contributed to the work equally and should be regarded as co-first authors. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Gansu Natural Science Foundation grant number [21JR7RA204, 1506RJZA007] and Gansu Province Higher Education Innovation Foundation [2022B-107, 2019A-056].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

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

References

1. Reddy Maddikunta, P.K.; Hakak, S.; Alazab, M.; Bhattacharya, S.; Gadekallu, T.R.; Khan, W.Z.; Pham, Q.-V. Unmanned Aerial Vehicles in Smart Agriculture: Applications, Requirements, and Challenges. *IEEE Sens. J.* **2021**, *21*, 17608–17619. [\[CrossRef\]](#)
2. Przesmycki, R.; Michalak, J. Military suitability of COTS UAV due to the level of radiated emissions. *Bull. Pol. Acad. Sci. Tech. Sci.* **2024**, *72*, 148838. [\[CrossRef\]](#)
3. Yudhi, R.M.; Ambar, Y.; Abdurrahman, P.; Gunawan, S.A.; Arafat, F.; Cahyo, T.; Ridwan, S. UAV-assisted heavy metal tracking in oil palm plantations: Present applications and future prospects. *Remote Sens. Lett.* **2024**, *15*, 55–65.
4. Ashish, S.; Jay, P. Techniques, Answers, and Real-World UAV Implementations for Precision Farming. *Wirel. Pers. Commun.* **2023**, *131*, 2715–2746.
5. Quan, X.H.; Guo, Q.L.; Ma, J.; Doluschitz, R. The economic effects of unmanned aerial vehicles in pesticide application: Evidence from Chinese grain farmers. *Precis. Agric.* **2023**, *24*, 1965–1981. [\[CrossRef\]](#)
6. Kurdi, H.; AlDaood, M.F.; Al-Megren, S.; Aloboud, E.; Aldawood, A.S.; Youcef-Toumi, K. Adaptive task allocation for multi-UAV systems based on bacteria foraging behaviour. *Appl. Soft Comput. J.* **2019**, *83*, 105643. [\[CrossRef\]](#)
7. Tang, J.; Duan, H.B.; Lao, S.Y. Swarm intelligence algorithms for multiple unmanned aerial vehicles collaboration: A comprehensive review. *Artif. Intell. Rev.* **2022**, *56*, 4295–4327. [\[CrossRef\]](#)
8. Zhang, X.P.; Chen, X.J. UAV Task Allocation Based on Clone Selection Algorithm. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 5518927. [\[CrossRef\]](#)
9. Schwarzrock, J.; Zacarias, I.; Bazzan, A.L.C.; Fernandes, R.Q.d.A.; Moreira, L.H.; Freitas, E.P.d. Solving task allocation problem in multi Unmanned Aerial Vehicles systems using Swarm intelligence. *Eng. Appl. Artif. Intell.* **2018**, *72*, 10–20. [\[CrossRef\]](#)
10. Yu, X.Y.; Gao, X.H.; Wang, L.; Wang, X.W.; Ding, Y.; Lu, C.; Zhang, S. Cooperative Multi-UAV Task Assignment in Cross-Regional Joint Operations Considering Ammunition Inventory. *Drones* **2022**, *6*, 77. [\[CrossRef\]](#)
11. Deng, M.; Yao, Z.; Li, X.; Wang, H.; Nallanathan, A.; Zhang, Z. Dynamic Multi-Objective AWPSP in DT-Assisted UAV Cooperative Task Assignment. *IEEE J. Sel. Areas Commun.* **2023**, *41*, 3444–3460. [\[CrossRef\]](#)
12. Zhou, T.L.; Chen, M.; Han, Z.L.; Wang, Q. Multi-UAV Cooperative Multiple Targets Assignment Based on Improved Wolf Pack Algorithm. *Navig. Position Timing* **2022**, *9*, 46–55.
13. Shi, J.Q.; Tan, L.; Lian, X.F.; Xu, T.Y.; Zhang, H.T.; Zhang, Y. A multi- unmanned aerial vehicle dynamic task assignment method based on bionic algorithms. *Comput. Electr. Eng.* **2022**, *99*, 107820. [\[CrossRef\]](#)
14. Jiang, Y.; Wang, D.B.; Lin, F.; Bai, T.T.; Jiang, D.D. Unbalanced Target Assignment of Multi-UAV Based on Hungarian-Genetic Algorithm. *Electron. Opt. Control.* **2023**, *30*, 6–10+22.

15. Liang, Z.B.; Li, Q.; Fu, G.D. Multi-UAV Collaborative Search and Attack Mission Decision-Making in Unknown Environments. *Sensors* **2023**, *23*, 7398. [\[CrossRef\]](#)
16. Yu, Y.P.; Liu, J.C.; Wei, C. Hawk and pigeon's intelligence for UAV swarm dynamic combat game via competitive learning pigeon-inspired optimization. *Sci. China Technol. Sci.* **2022**, *65*, 1072–1086. [\[CrossRef\]](#)
17. Chen, J.H.; Li, W.L.; Wang, X.M. Optimal power flow calculation with reverse mutation seagull optimization algorithm. *J. Yanshan Univ.* **2023**, 1–12.
18. Xu, L.; Mo, Y.B.; Lu, Y.Y.; Li, J. Improved Seagull Optimization Algorithm Combined with an Unequal Division Method to Solve Dynamic Optimization Problems. *Processes* **2021**, *9*, 1037. [\[CrossRef\]](#)
19. Dhiman, G.; Singh, K.K.; Slowik, A.; Chang, V.; Yildiz, A.R.; Kaur, A.; Garg, M. EMoSOA: A new evolutionary multi-objective seagull optimization algorithm for global optimization. *Int. J. Mach. Learn. Cybern.* **2021**, *12*, 571–596. [\[CrossRef\]](#)
20. Guo, H.; Mo, Y.; Zhang, Y. Orthogonal Design-Based Control Vector Parameterization Combined with Improved Seagull Optimization Algorithm for Dynamic Optimization Problems. *IEEE Access* **2022**, *10*, 65238–65256. [\[CrossRef\]](#)
21. Rather, S.A.; Das, S. Levy Flight and Chaos Theory-Based Gravitational Search Algorithm for Image Segmentation. *Mathematics* **2023**, *11*, 3913. [\[CrossRef\]](#)
22. Qizhao, Z.; Hongshun, L.; Jian, G.; Yifan, W.; Luyao, L.; Hongzheng, L.; Haoxi, C. Improved GWO-MCSVM algorithm based on nonlinear convergence factor and tent chaotic mapping and its application in transformer condition assessment. *Electr. Power Syst. Res.* **2023**, *224*, 109754.
23. Xu, B.W.; Ning, Q. Multi-UAV task assignment based on improved genetic algorithm. *Mod. Comput.* **2023**, *29*, 15–21.
24. Zhang, J.D.; Chen, Y.Y.; Yang, Q.M.; Lu, Y.; Shi, G.Q.; Wang, S.; Hu, J.W. Dynamic Task Allocation of Multiple UAVs Based on Improved A-QCDPSO. *Electronics* **2022**, *11*, 1028. [\[CrossRef\]](#)
25. Dhiman, G.; Kumar, V. Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems. *Knowl.-Based Syst.* **2019**, *165*, 169–196. [\[CrossRef\]](#)
26. Zhao, J.T.; Luo, X.C.; Liu, J.M. Application of Improved Whale Optimization Algorithm in Robot Path Planning. *J. Northeast. Univ.* **2023**, *44*, 1065–1071.
27. Zhang, T.R.; Cong, S.R. Research on hybrid Seru production scheduling based on improved slime mold optimization algorithm. *Intell. Manuf.* **2023**, *8*, 47–55.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.