electronics

MDPI

Article

A Multi-Strategy Crazy Sparrow Search Algorithm for the Global Optimization Problem

Xuewei Jiang, Wei Wang *, Yuanyuan Guo and Senlin Liao

College of Mechanical and Electrical Engineering, Northeast Forestry University, Harbin 150040, China; jxwvv0@nefu.edu.cn (X.J.); 961213@nefu.com (Y.G.); linefu@nefu.edu.cn (S.L.)
* Correspondence: vickywong@nefu.edu.cn; Tel.: +86-133-1361-3588

Abstract: A multi-strategy crazy sparrow search algorithm (LTMSSA) for logic-tent hybrid chaotic maps is given in the research to address the issues of poor population diversity, slow convergence, and easily falling into the local optimum of the sparrow search algorithm (SSA). Firstly, the LTMSSA employs an elite chaotic backward learning strategy and an improved discoverer-follower ratio factor to improve the population's quality and diversity. Secondly, the LTMSSA updates the positions of discoverers and followers by the crazy operator and the Lévy flight strategy to expand the selection range of target following individuals. Finally, during the algorithm's optimization search, the LTMSSA introduces the tent hybrid and Corsi variable perturbation strategies to improve the population's ability to jump out of the local optimum. Different types and dimensions of test functions are used as performance benchmark functions to test the performance of the LTMSSA with SSA variants and other algorithms. The simulation results show that the LTMSSA can jump out of the optimal local solution, converge faster, and have higher accuracy. Its overall performance is better than the other seven algorithms, and the LTMSSA can find smaller optimal values than other algorithms in the welded beam and reducer designs. The results confirm that the LTMSSA is an effective aid for computationally complex practical tasks, provides high-quality solutions, and outperforms other algorithms.

Keywords: sparrow search algorithm; metaheuristic; multi-strategy hybrid; engineering design problems

Citation: Jiang, X.; Wang, W.; Guo, Y.; Liao, S. A Multi-Strategy Crazy Sparrow Search Algorithm for the Global Optimization Problem. *Electronics* **2023**, *12*, 3967. https://doi.org/10.3390/electronics12183967

Academic Editors: Jose-Luis Sanchez-Romero, Christopher Paolini and Krzysztof Wołk

Received: 25 August 2023
Revised: 13 September 2023
Accepted: 18 September 2023
Published: 20 September 2023

1. Introduction

Optimization refers to finding the best solution to achieve an objective by satisfying all constraints. Traditional optimization methods, such as simplex and gradient descent [1], are fast to solve and have a more mature mathematical theoretical basis. The feasibility requirements greatly influence the extent to which these algorithms can be implemented, and the solution capability could be better for optimization problems with unknown mathematical characteristics. The metaheuristic algorithm is an intelligent iterative search-based optimization method combining local search strategies and stochastic algorithms [1], which has no constraints on the nature of the problem, is highly searchable and widely applicable, and can solve practical optimization problems efficiently. Metaheuristic algorithms include the gravitational search algorithm (GSA) [2], the fireworks algorithm (FA) [3], the genetic algorithm (GA) [4], the harmony search algorithm (HS) [5], the seagull optimization algorithm (SOA) [6], the Harris hawks optimization (HHO) [7], and so on. Based on simulated evolutionary theory, these evolutionary computational methods are self-organizing and efficient at finding the optimum, making them effective at solving intricate solutions and massive optimization issues. Swarm intelligence algorithms are widely used today and yield significant benefits in many fields. Ever since the pioneering work of Beni and Wang, who introduced the notion of swarm intelligence (SI) [8], there has been an increase in the amount of in-depth research being conducted. Swarm intelligence algorithms have achieved rich results regarding theoretical systems and practical applications.

Particle swarm optimization (PSO) [9,10] and ant colony optimization (ACO) [11] are two good examples of optimization techniques. The foundation of search algorithms lies in the intricate mechanism of information exchange between individuals in a population as they progress through the evolution of a biological system. PSO has the advantage of good global convergence ability and ease of implementation, but there is a strong parameter dependency. For this reason, some scholars have introduced taboo search mechanisms [12] and local search strategies [13,14] to enhance the PSO algorithm's efficiency. Inspired by ant foraging, ACO provides rapid solution times and computational simplicity. Researchers have increasingly been drawn to swarm intelligence optimization algorithms due to the competitive nature and global search capabilities demonstrated by algorithms like PSO and ACO. Yang et al. proposed the bat algorithm (BA) [15], which simulates bat echolocation. The grey wolf optimizer (GWO) algorithm is widely recognized for its ability to mimic the hierarchical organization and predatory behavior observed in grey wolves [16]. In addition, several algorithms for optimizing group intelligence have emerged in the past few years: the bacterial foraging algorithm (BFA) [17], the artificial bee colony (ABC) algorithm [18,19], the pigeon-inspired optimization (PIO) [20], the firefly algorithm (FA) [21,22], the cuckoo search (CS) algorithm [23,24], the whale optimization algorithm (WOA) [25], the fly optimization algorithm (FOA) [26], the krill herd (KH) [27], the dragonfly algorithm (DA) [28], the monkey algorithm (MA) [29], the beetle antennae search algorithm (BAS) [30], and so on.

The sparrow search algorithm (SSA) was put forward by Xue et al. as a cutting-edge bionic stochastic search technique [31], originating from an in-depth study of the social and behavioral characteristics of sparrow groups of predators and anti-predators. This algorithm has been widely applied and has obtained good results in various practical optimization problems, such as image segmentation [32], trajectory planning [33], workshop scheduling [34], power generation prediction [35], and other fields. However, this algorithm solves complex multidimensional optimization problems with the limitation that the sparrow population convergence is more severe and prone to early convergence. It is of great theoretical research value to determine how to effectively overcome the shortcomings of SSA to enhance its performance processing practical optimization problems and thus obtain a more widespread application.

1.1. Literature Review

There have been many improvements made by researchers to strengthen the effectiveness of SSA, and this paper will discuss two aspects of such algorithms.

1.1.1. Improving the Search Mechanism of the Algorithm

Tang et al. presented a progressive cosine algorithm aimed at modifying the placement of new members in a positive manner. They used a linear decreasing method to control individual alert scouting sparrows [36], balancing local exploitation of the algorithm with global exploration. Tang et al. developed a hierarchy and Brownian motion strategy to improve information communication between individuals; maintaining sparrow diversity involves upgrading improved sparrow placements [37]. Zhang et al. introduced a Corsi variation strategy for solving the optimum local problem using a tent mapping initialization of the population [38], effectively enhancing the algorithm's search capability. A chaotic mapping method was used by Chen et al. to determine the location of sparrow populations, and the Lévy flight and random wandering strategies were introduced [39], which effortfully optimized algorithm efficiency and exploration capabilities. Ou-Yang et al. utilized the k-means algorithm cluster analysis to differentiate the sparrow population positions in SSA [40], reducing the perturbation, and applied it to the optimization study of the active suspension LQR controller. Ou-Yang et al. introduced a mirror-opposite learning strategy into the discoverer's stage to increase population variety and search flexibility [41]. Mao et al. interfered with the optimal sparrow individuals using a Corsi variation and backward learning strategy to escape local optima [42]. Fu et al. introduced an innovative method called the elite chaotic backward learning mechanism in the population initial-

ization stage. By combining it with a randomized following strategy in the chicken flock algorithm, they enhanced the quality and diversity of individuals, resulting in improved search performance for updating the joiner location [43]. Duan et al. employed Sobol sequence mapping to construct the population's beginning location and used horizontal as well as vertical crossing to prevent local optimum and sustain sparrow variety [44]. Chen et al. incorporated the spiral exploration strategy into the discoverer search mechanism [45], which greatly improved the capacity for global exploration. By employing advanced diversity-enhancing techniques such as elite dissimilarity and random inverse hybrid variation, the algorithm effectively avoids premature convergence. This is achieved through the swift assimilation of individuals during the later stages of the iteration process. Yan et al. made the location distribution of the initial population more consistent by adding good point sets [46]. Combining the characteristics of the SSA algorithm, an iterative local search method with increased flexibility is introduced. A backward learning mechanism for dimension-by-dimensional lens imaging is introduced to reduce perturbations between dimensions and avoid premature convergence. He et al. asserted an SSA which varied the hybrid quantum behavior of the inferior population by introducing a quantum strategy. This modification had a profound effect on the superheated temperature control system, enabling them to accurately determine its parameters [47]. Wu et al. integrated a competition mechanism into their approach, aiding the sparrow in locating the optimal position nearest to the individual [48]. They cleverly employed a positive cosine strategy to strike a balance between exploring the wider context and conducting a targeted search within the local vicinity. A polynomial variation strategy was introduced to eliminate local optima. Liu et al. [49] enhanced the global search algorithm using t-distribution perturbation, which was discussed in the literature. For the sparrow individual crossing problem, a random regression strategy was adopted and used to deal with the design of the stressor. Referring to Sin's chaotic search mechanism, Ma et al. improved the initialization of the population by introducing the Lévy flight perturbation mechanism into the SSA to improve the spatial search diversity [50].

1.1.2. Integration of Other Algorithms

To enhance the multiplicity of sparrow populations, Tian et al. devised an algorithm that employs fusion arithmetic optimization. Their strategy involves substituting the population's individuals with an undirected weighted graph [51]. According to the improved circle algorithm to calculate the Hamiltonian ring length composed of population individuals, the ratio of the Hamiltonian ring lengths of two adjacent generations was used to indicate the population convergence trend. The pressure vessel and butterfly spring problems are effectively resolved by the greedy algorithm, which employs a strategy of randomly selecting individuals generated within a specific range. This approach enables the algorithm to break free from local optimum solutions. Yang et al. have incorporated the PSO algorithm with the SSA to enhance convergence speed [52]. Li et al. mixed a simulated annealing algorithm with the SSA to remove the algorithm from its local maximum [53], which boosts the algorithm's global exploration. Using the simplex method, Liu et al. moved individuals with weak adaptability to enhance algorithm variety and search capacity after each cycle [54].

1.2. Research Gaps and Motivations

Our previous analysis demonstrated the resounding effectiveness of the SSA, as it transcends across different realms. Many scholars have made notable strides in enhancing the original algorithm's search optimization capability by implementing iterative updates, amalgamating with other algorithms, and modifying parameters. Nevertheless, numerous research findings have primarily enhanced the algorithm's capacity in a one-sided manner. The algorithm must continue to strike a delicate balance between effectively utilizing data and allowing for exploratory analysis. The ability to uphold diversity still needs to be discovered. In tackling non-convex problems, the challenges of achieving accurate

optimization, conducting effective global searches, and avoiding the pitfalls of local optima persist and require attention. There is still ample opportunity for enhancing and advancing the SSA.

1.3. Contribution

To further boost the optimization performance, this research put forth a multi-strategy hybrid SSA methodology. The complete integration of dynamic parameters into every algorithm phase ensures a comprehensive evaluation of their relationships. The individual population experiences a shift towards being the majority. Significant improvements have been implemented in the dynamic parameters. We have included the active adaptive adjustment strategy as part of our approach. The purpose of dynamic global optimization is accomplished through the mutual influence of various interactions on one another. The primary advancements are showcased in the following manner:

i. The introduction of the LTMSSA aims to address the issues inherent to the initial algorithm.
ii. The LTMSSA increases the population diversity by utilizing logistic-tent hybrid chaotic maps and improves the discoverer-follower scaling factor to make the algorithm more accurate at convergence.
iii. Improving the discoverer and follower positions with the crazy operator and the Lévy flight strategy, respectively, and introducing the tent hybrid and Corsi variational perturbation strategies to reconcile the capability for both local and global search of the SSA.
iv. The LTMSSA is evaluated for efficacy on 23 standard test functions and compared to different algorithms.
v. The scalability of the LTMSSA in various dimensions is examined and implemented in practical engineering issues involving the design of welded beams and reducers. Results indicate that the enhanced LTMSSA strategy proposed applies to solving optimization-related issues.

This study follows this structure: in Section 2, the concept of the SSA is introduced. In Section 3, the LTMSSA mechanism and its mathematical model are comprehensively outlined and discussed. In Section 4, the test results for LTMSSA and several other algorithms on 23 benchmark functions are presented. In Section 5, the practical engineering hurdles associated with the welded beam and reducer design are effectively addressed through the utilization of the LTMSSA. In Section 6, the paper comes to a conclusion.

2. Sparrow Search Algorithm

Sparrow search algorithms have three categories of individuals, namely discoverers, followers, and vigilantes, and their positions are representative of potential solutions. Throughout the foraging process, the three places are updated to find the ideal food source, which is the best solution.

2.1. Sparrow Search Algorithm Model

In the following matrix, the sparrow population can be represented.

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,d} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,d} \end{bmatrix} \quad (1)$$

where d denotes the problem dimension and n is the total amount of sparrows in the population. Depending on the sparrow population's fitness, the following is calculated.

$$F_X = \begin{bmatrix} f\left(\begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d} \end{bmatrix}\right) \\ f\left(\begin{bmatrix} x_{2,1} & x_{2,2} & \cdots & x_{2,d} \end{bmatrix}\right) \\ \vdots \\ f\left(\begin{bmatrix} x_{n,1} & x_{n,2} & \cdots & x_{n,d} \end{bmatrix}\right) \end{bmatrix} \tag{2}$$

where F_X denotes the fitness value of the sparrow and f denotes the fitness value of the problem dimension corresponding to each sparrow.

Since the discoverer in the sparrow population is searching for food for all sparrows, the discoverer experiences a greater level of fitness. Throughout the algorithm's iteration, the discoverer's position is continuously adjusted in accordance with Equation (3).

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^{t} \cdot \exp\left(\frac{-i}{\alpha \cdot T}\right), & R_2 < ST \\ X_{i,j}^{t} + Q \cdot L, & R_2 \geq ST \end{cases} \tag{3}$$

where t denotes the number of iterations, $j = 1, 2 \ldots d$; T denotes the maximum number of iterations, $X_{i,j}$ denotes the coordinates of the position of the i th sparrow in the j th dimension, $\alpha \in (0, 1]$ is a randomly generated number, R_2 ($R_2 \in [0, 1]$) is the warning value, ST ($\mathrm{ST} \in [0.5, 1]$) denotes the security value, Q is a random number that follows a normal distribution, and L denotes a $1 \times$ d matrix. When $R_2 < ST$, the discoverer can perform extensive foraging operations and there are no predators near the foraging area. Conversely when $R_2 \geq ST$, some sparrows deliver a danger alert to the remaining sparrows, thus sparrows will be able to move quickly to foraging areas that are safe.

The followers will compete with the finder for food when they are looking for food. If they succeed, the followers will obtain the food searched by the finder; if not, they will continue to follow the finder. Equation (4) shows the formula for updating follower positions.

$$X_{i,j}^{t+1} = \begin{cases} Q \cdot \exp\left(\frac{X_{\text{worst}}^{t} - X_{i,j}^{t}}{i^2}\right), & i > n/2 \\ X_{p}^{t+1} + \left|X_{i,j}^{t} - X_{i,j}^{t+1}\right| \cdot A^{+} \cdot L, & i \leq n/2 \end{cases} \tag{4}$$

where X_p^{t+1} denotes the global best position of the sparrow at $t + 1$th iteration, X_{worst}^{t} denotes the global worst position of the sparrow at t th iteration, n is the population size, A is a matrix of $1 \times$ d, randomly generated 1 or -1 is assigned to the matrix A, and $A^{+} = A^{T}\left(AA^{T}\right)^{-1}$.

A random position is chosen for the vigilantes, and the formula for updating their status is presented in Equation (5).

$$X_{i,j}^{t+1} = \begin{cases} X_{\text{best}}^{t} + \beta \cdot \left(X_{i,j}^{t} - X_{\text{best}}^{t}\right), & f_i \neq f_g \\ X_{i,j}^{t} + K \cdot \left(\frac{X_{i,j}^{t} - X_{\text{worst}}^{t}}{|f_i - f_{worst}| + \varepsilon}\right), & f_i = f_g \end{cases} \tag{5}$$

where X_{best} is the current global best position, β is a randomly generated parameter that controls the row length of sparrows and follows a normal distribution with variance 1 and mean 0, K is also a random number that controls the search direction of sparrows and ranges from $[-1, 1]$, f_i is the fitness value of the ith sparrow at this time, f_g is the fitness value of the sparrow with the worst position in the entire sparrow population, f_{worst} is the fitness value of the sparrow with the worst position in the entire sparrow population, and ε is the smallest constant to make the denominator in Equation (5) not equal to 0. When $f_i \neq f_g$, the sparrow at the edge of the population has a greater chance of getting caught by predators, and X_{best} indicates that at this moment, the sparrow is in the safest position.

When $f_i = f_g$, sparrows in the middle of a population perceive a crisis and move closer to the sparrow in a safe position to prevent the risk of being caught.

2.2. Sparrow Search Algorithm Pseudo Code

According to the above algorithm design steps, the following Algorithm 1 is a pseudocode representation of the specific algorithm flow.

Algorithm 1 The framework of the SSA.

Input:
T: the maximum iterations
PD: the number of producers
SD: the number of sparrows who perceive the danger
R_2: the alarm value
ST: safety value
n: the number of sparrows
Initialize a population of n sparrows and define its relevant parameters.
Output: X_{best}, f_{best}.
1: **While** (t < T)
2: Rank the fitness values and find the current best individual
and the current worst individual.
3: R_2 = rand (1)
4: **for** i = 1: PD
5: Using Equation (3) update the sparrow's location;
6: **end for**
7: **for** i = (PD + 1): n
8: Using Equation (4) update the sparrow's location;
9: **end for**
10: **for** i = 1: SD
11: Using Equation (5) update the sparrow's location;
12: **end for**
13: Obtain the current new location;
14: If the new location is better than before, update it;
15: t = t + 1
16: **End While**
17: **return** X_{best}, f_{best}.

3. Multi-Strategy Hybrid Crazy Sparrow Search Algorithm

3.1. Population Initialization

3.1.1. Logistic-Tent Hybrid Chaos Map

Chaotic systems describe a complex chaotic phenomenon arising from deterministic nonlinear systems sensitive to initial value conditions, which are nonperiodic, and internally stochastic [55,56]. In general, chaotic systems are classified into low- and high-dimensional chaos [57]. To develop chaotic systems with better chaotic performance, researchers have combined multiple low-dimensional chaos to form a new composite chaotic system. This type of chaotic system can effectively overcome the shortcomings of low-dimensional chaos and is less complex and more accessible to implement than high-dimensional chaos. Tang et al. proposed a logistic-sine composite chaotic mapping and used it for image encryption to address the limitations of traditional one-dimensional chaotic systems and the security of parameter values [58] and achieved good encryption results. In order to safeguard the algorithm's integrity, Zhang et al. employed the chaotic sequence derived from the logistic-sine-cosine composite chaos system. This involved the application of rank dislocation and cyclic pixel diffusion methodologies [59]. Combining logistic-tent chaotic maps with tent chaotic maps improves the randomness of the algorithm.

The mathematical formula for it is:

$$X_{n+1} = \begin{cases} \left(rX_n(1-X_n) + \frac{(4-r)X_n}{2}\right) mod1, X_n < 0.5 \\ \left(rX_n(1-X_n) + \frac{(4-r)(1-X_n)}{2}\right) mod1, X_n \geq 0.5 \end{cases} \quad (6)$$

where X is the system variable, r is the control parameter, $X \in [0,1]$, and r $\in (0,4)$.

The amalgamation of the logistic and tent chaotic systems in this disorderly setup results in complex dynamics, offering faster iteration, heightened autocorrelation, and broad applicability to many sequences.

3.1.2. Initial Population Elitism

To enhance the original population by using lens imaging inverse learning [60], let x_j and x_j^* denote the current individual sparrow and its individual after the reversal of lens imaging, respectively.

$$x_j^* = \frac{a_j + b_j}{2} + \frac{a_j + b_j}{2k} - \frac{x_j}{k} \quad (7)$$

where a_j and b_j denote the minimum and maximum values in the j th dimension of the current population, respectively, and k is the scaling factor of the lens, $k = 10,000$.

In order to initiate the sparrow population using the elite chaos reversal learning approach, the following set of procedures must be followed: the initial sparrow population $X = [x_{i1}, \ldots, x_{id}]$, $i = 1, \ldots, n$, x_{id} denotes the position of the i th sparrow in the d th dimension, the population x is substituted into Equation (7) to generate the chaotic population Y, and the population X is substituted into Equation (8) to generate the lenticular imaging reversal population Z. The sparrow individuals in population Y and population Z are ranked according to their fitness values, and the top n individuals are selected to form the elite chaotic reverse population P. The top n individuals of population P and the original sparrow population X are then selected to form a new initial sparrow population based on the ranking of individual fitness values.

$$X' = \left[x'_{i1}, \ldots, x'_{id}\right] \quad (8)$$

3.2. Location Formula Update

3.2.1. Proportionality Improvement

The SSA algorithm maintains a constant ratio between discoverers and followers, which may lead to an inefficient search globally. The main objective of this study is to put forth a strategy that effectively improves the discoverer-follower ratio coefficient, resulting in a reduction in discoverers and a simultaneous increase in followers through an adaptive approach.

To determine the optimal number of discoverers and followers, one can calculate the adjustments using the following prescribed method:

$$r = b(\tan\left(-\frac{\pi t}{4 \cdot iter_{max}} + \frac{\pi}{4}\right) - \mathrm{k} \cdot \mathrm{rand}(0,1)) \quad (9)$$

$$pNum = r \cdot N \quad (10)$$

$$sNum = (1-r)\, N \quad (11)$$

where k is the perturbation variation factor to disturb the nonlinearly falling r value, $pNum$ is the total amount of discoverers, $sNum$ is the number of followers, and the value of b is instrumental in maintaining an optimal equilibrium between pioneers and adopters, effectively regulating the ratio of individuals exploring new ideas versus those embracing them.

3.2.2. Join the Madness Calculator to Improve the Discoverer

The element of "madness" plays a crucial role in amplifying the unexpected behavior exhibited by the group [61]. A madness operator is introduced into the discoverer's location update equation to maintain a diverse set of results. This operator perturbs the discoverer's position with a predetermined probability of madness, effectively adding an unpredictable aspect to the equation. Here is the latest formulation for the discoverer update:

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^{t} \cdot \exp\left(\frac{-i}{\alpha \cdot T}\right) + P(c_1) \cdot sign(c_1) \cdot x_{8\text{raziness}}, & R_2 < ST \\ X_{i,j}^{t} + Q \cdot L + P(c_1) \cdot sign(c_1) \cdot x_{8\text{raziness}}, & R_2 \geq ST \end{cases} \tag{12}$$

where $P(c_1)$ and $sign(c_1)$ are defined, respectively, as:

$$P(c_1) = \begin{cases} 1, & c_1 \leq P_{cr} \\ 0, & otherwise \end{cases} \tag{13}$$

$$sign(c_1) = \begin{cases} -1, & c_1 \geq 0.5 \\ 1, & otherwise \end{cases} \tag{14}$$

where c_1 denotes a number chosen at random, with a uniform distribution between $[0, 1]$; x_{crazines} is usually taken as a small constant (=0.0001). P_{cr} is the set probability of madness. In this case, if P_{cr} takes a small value (=0.3), c_1 will likely exceed P_{cr} and the madness factor $P(c_1)$ will be 0.

3.2.3. Lévy Flight Strategy to Improve Followers

Lévy flight strategies have been successfully applied to improve many swarm intelligence algorithms, and researchers have been inspired by it to introduce Lévy mechanisms in update strategies to improve algorithm performance. The step length of the walk satisfies a heavy-tailed Lévy distribution as shown in Equation (15):

$$L(s) \sim \left|s\right|^{-1-\beta}, 0 < \beta \leqslant 2 \tag{15}$$

Equation (16) visually represents the erratic trajectory of the Lévy flight strategy, which is characterized by its random and sporadic steps.

$$s = \frac{\mu}{\left|\nu\right|^{\frac{1}{\beta}}} \tag{16}$$

where $\mu \sim \left(0, \sigma_{\mu}^{2}\right), v \sim \left(0, \sigma_{v}^{2}\right), 0 < \beta_1 < 2, \beta_1 = 1.5$.

$$\sigma_{\mu} = [\frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma(1+\beta)/2]\beta 2^{(\beta-1)/2}}]^{1/\beta_1} \tag{17}$$

The Lévy flight strategy has improved the following follower formula for the SSA algorithm:

$$X_{i,j}^{t+1} = \begin{cases} Q \cdot \exp\left(\frac{X_{\text{worst}}^{t} - X_{i,j}^{t}}{i^2}\right), & i > n/2 \\ X_{p}^{t+1} + \left|X_{i,j}^{t} - X_{i,j}^{t+1}\right| \cdot s, & i \leq n/2 \end{cases} \tag{18}$$

3.3. Tent Chaotic Perturbation and Corsi Variation Strategy

3.3.1. Tent Chaos Perturbation

The tent chaotic mapping equation was modified by Zhang et al., who introduced random variables $\text{rand}(0,1) \times \frac{1}{N}$ [62]. As a result, a new version of the tent chaotic mapping equation is presented below.

$$z_{i+1} = \begin{cases} 2z_i + \text{rand}(0,1) \times \frac{1}{N}, & 0 \le z \le \frac{1}{2} \\ 2(1-z_i) + \text{rand}(0,1) \times \frac{1}{N}, & \frac{1}{2} < z \le 1 \end{cases} \tag{19}$$

The following is the expression using the Bernoulli shift improvement:

$$z_{i+1} = (2z_i) mod 1 + \text{rand}(0,1) \times \frac{1}{N} \tag{20}$$

where N is the number of particles in the sequence.

3.3.2. Corsi Mutation

The Cauchy variation comes from constant frequency distributions and Cauchy distributions, and is characterized by a smaller peak at zero and a slow decline from the peak to the zero value [63], making the variation range more uniform. Variation can be characterized as follows.

$$mutation(x) = x(1 + \tan(\pi(u - 0.5)) \tag{21}$$

where x is the original individual position, $mutation(x)$ is the individual position after the Corsi variation, and u is a random number in the $(0,1)$ interval. Where x is the initial location of the individual, $mutation(x)$ is the position of the individual after the Corsi variation, and u is a random value within the range of $(0,1)$.

3.4. LTMSSA Flow Chart

The LTMSSA process diagram is shown in Figure 1. First, the initial parameters of the SSA are set, and the populations are initialized according to logistic-tent chaotic maps and elite reverse learning. Next, calculate the fitness value of each sparrow and its position and count the number of discoverers and followers, update the position of the three populations, and calculate the fitness value and average fitness value again. Finally, the tent mixing perturbation and the Cauchy mutation are performed, and when the perturbed and mutated individuals are better than the original individuals, the population fitness value, optimal position, and worst position are updated.

3.5. Computational Complexity

Assume that the algorithm incorporates a population of N individuals, each characterized by D dimensions. With a specified maximum iteration limit ($\text{iter}_{\max}$), s_1 is the optimal moment for initiating the population parameters in a random manner. Additionally, $j(D)$ encompasses the assessment of the suitability of every individual, working in conjunction with pNum discoverers, and the time taken for each dimension to be updated is represented by s_2. Moreover, it consists of sNum followers who necessitate s_3 time to update their dimensions, as well as alerters who rely on s_4 time for their updates. Therefore, the time complexity at the outset can be expressed as $T_1 = O(s_1 + N(j(D) + Ds_1))$. The update of the discoverer is characterized by a time complexity of $T_2 = O(pNums_2D))$. The time required to update followers is represented by the complexity $T_3 = O(sNums_3D)$. The time complexity for updating the alerters has been improved to $T_4 = O((N - pNum - sNum)s_4D$. To sum up, the complete time complexity of the SSA can be represented by the equation: $T = T_1 + (T_2 + T_3 + T_4)\text{iter}_{\max} = O(D + j(D))$.

The initial stage in the LTMSSA involves elite chaotic backward learning, which is estimated to take u_1 time, and sorting selection, which takes u_2 time. As a result, the

time complexity of this stage can be denoted as $T_{11} = O(s_1 + N(u_1 + j(D) + Ds_1) + u_2)$. The updated formula for the count of explorers and supporters is referred to as u_3, and consequently, the time complexity for updating the explorers is denoted as $T_{22} = O(pNums_2D + iter_{max}u_3)$. The time complexity for updating the followers is denoted as $T_{33} = O(sNums_3D + iter_{max}u_3)$, while the time complexity for updating the alerters is denoted as $T_{44} = O((N - pNum - sNum)s_4D)$. In the context of the Corsi variation and the tent chaos perturbation process, assign the variable u_4 to represent the time taken to solve f_{avg}, while u_5 and u_6, respectively, denote the time needed for computing the perturbation formula and the Corsi variation formula. During this stage, u_7 denotes the specific point in time when we compare the fitness value of the sparrow with the average fitness value. Similarly, u_8 represents the moment when the target position is updated based on merit. As a result, the time complexity for this stage is succinctly expressed as $T_{55} = O(u_4 + u_5 + u_6 + N(j(D) + u_7) + u_8)$. In conclusion, the overall time complexity of the LTMSSA can be represented as follows: $TT = T_{11} + (T_{22} + T_{33} + T_{44} + T_{55})iter_{max} = O(D + j(D))$. Given that $TT = T$, it demonstrates that the time complexity of the LTMSSA remains unchanged.

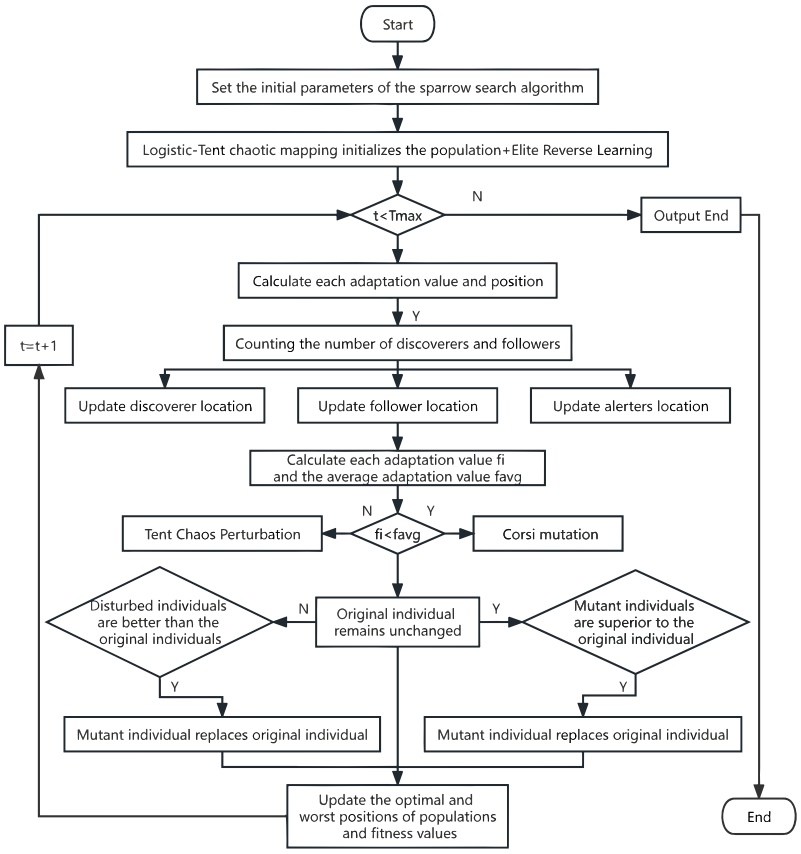

Figure 1. The LTMSSA process diagram.

4. Experimental Results and Discussion

4.1. Test Function and Algorithm Parameters

To gauge the performance of the LTMSSA, a simulation has been conducted using 23 benchmark test functions. Each algorithm maintains a population size of 30 (represented as N) and caps the number of iterations at 500 (designated as M). Meanwhile, SSA [31], SSSA [64], FSSA [65], CSFSSA [45], GWO [16], PSO [10], and BOA [66] are the algorithms used in this study for comparison, and the experimental parameters of each algorithm are shown in Table 1. Table 2 showcases a graphical depiction of single-peak functions F1–F7 in high-dimensional settings, multi-peak functions F8–F13 in high-dimensional settings, and multi-peak functions F14–F23 in low-dimensional settings. High-dimensional single-peak

functions exhibit a distinct global optimum point and do not feature any local extreme points when assessing the speed of convergence. From a multidimensional perspective, local extremum points can give rise to multi-peaked functions, showcasing their diverse peaks in different dimensions.

Table 1. Experimental parameters.

Algorithms	Parameters
SSA	ST = 0.8, PD = 0.2, SD = 0.2
SSSA	ST = 0.8, PD = 0.2, SD = 0.2
FSSA	ST = 0.8, PD = 0.2, SD = 0.2
CSFSSA	ST = 0.8, PD = 0.2, SD = 0.2
LTMSSA	ST = 0.8, PD = 0.2, SD = 0.2
GWO	a = (2→0), r_1, $r_2 \in [0, 1]$
PSO	W = 0.9, C_1 = 1.49445, C_2 = 1.49445
BOA	a = (0.1→0.3)

Table 2. Test Functions.

Type	Function	Dimension	Scope	Optimal Value
Unimodal functions	$F_1(x) = \sum_{k=1}^{D} x_k^2$	30	$[-100, 100]$	0
	$F_2(x) = \sum_{k=1}^{D} \lvert x_k \rvert + \prod_{k=1}^{D} \lvert x_k \rvert$	30	$[-10, 10]$	0
	$F_3(x) = \sum_{k=1}^{D} (\sum_{l=1}^{k} x_l)^2$	30	$[-100, 100]$	0
	$F_4(x) = \max_k [\lvert x_k \rvert, 1 \le k \le D]$	30	$[-100, 100]$	0
	$F_5(x) = \sum_{k=1}^{D-1} [100((x_{k+1} - x_k^2))^2 + (x_k - 1)^2]$	30	$[-30, 30]$	0
	$F_6(x) = \sum_{k=1}^{D} (\lvert x_k + 0.5 \rvert)^2$	30	$[-100, 100]$	0
	$F_7(x) = \sum_{k=1}^{D} k x_k^4 + \text{random}(0, 1)$	30	$[-1.28, 1.28]$	0
Multimodal functions	$F_8(x) = \sum_{k=1}^{D} -x_k \sin(\sqrt{\lvert x_k \rvert})$	30	$[-500, 500]$	$-418.9826 \times$ D
	$F_9(x) = \sum_{k=1}^{D} [x_k^2 - 10\cos(2\pi x) + 10]$	30	$[-5.12, 5.12]$	0
	$F_{10}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{k=1}^{D} x_k^2} - \exp\left(\frac{1}{D}\sum_{k=1}^{D} \cos 2\pi x_k\right)\right) + 20 + e$	30	$[-32, 32]$	0
	$F_{11}(x) = \sum_{k=1}^{D} \lvert x_k \sin(x_k) + 0.1 x_k \rvert$	30	$[-10, 10]$	0
	$F_{12}(x) = \frac{\pi}{D} \cdot (10 sin(\pi y_1) + (y_k - 1)^2) + \frac{\pi}{D} \cdot \sum_{k=1}^{D-1} (y_k - 1)^2 [1 + 10\sin^2(\pi y_{k+1})] + \sum_{k=1}^{D} \mu(x_k, 10, 100, 4)$, $y_k = 1 + \frac{x_k + 1}{4}$, $\mu(x_k, p, a, m) = \begin{cases} p(x_k - a)^m, x_k > a \\ p(-x_k - a)^m, x_k < -a \end{cases}$	30	$[-50, 50]$	0
	$F_{13}(x) = 0.1\{\sin^2(3\pi x_1) + \sum_{k=1}^{D} (x_k - 1)^2 [1 + \sin^2(3\pi x_k + 1)] + (x_k - 1)^2 [1 + \sin^2(2\pi x_k)] + \sum_{k=1}^{D} \mu(x_k, 5, 100, 4)\}$, $\mu(x_k, p, a, m) = \begin{cases} p(x_k - a)^m, x_k > a \\ 0, -a < x_k < a \\ p(-x_k - a)^m, x_k < -a \end{cases}$	30	$[-50, 50]$	0

Table 2. *Cont.*

Type	Function	Dimension	Scope	Optimal Value
Fixed dimensional functions	$F_{14}(x) = \left(\frac{1}{500} + \sum_{l=1}^{25}\left(\frac{1}{l+\sum_{k=1}^{D}(x_k - a_{kl})^6}\right)\right)^{-1}$	2	$[-65.536, 65.536]$	0.998
	$F_{15}(x) = \sum_{k=1}^{11}\left[a_i - \frac{x_1(b_k^2+b_k x_2)}{b_k^2+b_k x_3+x_4}\right]^2$	4	$[-5, 5]$	0.0003075
	$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	$[-5, 5]$	-1.0316
	$F_{17}(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$	2	$[-5, 5]$	0.398
	$F_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]$ $\times[30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	$[-2, 2]$	3
	$F_{19}(x) = -\sum_{k=1}^{4} c_k \exp\left(-\sum_{l=1}^{3} a_{kl}(x_l - p_{kl})^2\right)$	3	$[0, 1]$	-3.86
	$F_{20}(x) = -\sum_{k=1}^{4} c_k \exp\left(-\sum_{l=1}^{6} a_{kl}(x_l - p_{kl})^2\right)$	6	$[0, 1]$	-3.32
	$F_{21}(x) = -\sum_{k=1}^{5}\left[(X - a_k)(X - a_k)^T c_k\right]^{-1}$	4	$[0, 10]$	-10.1532
	$F_{22}(x) = -\sum_{k=1}^{7}\left[(X - a_k)(X - a_k)^T c_k\right]^{-1}$	4	$[0, 10]$	-10.4028
	$F_{23}(x) = -\sum_{k=1}^{10}\left[(X - a_k)(X - a_k)^T c_k\right]^{-1}$	4	$[0, 10]$	-10.5363

4.2. Scalability Testing

To test the algorithmic scalability of the LTMSSA, the LTMSSA was compared with SSA in different dimensions. According to Table 3, the test outcomes of the LTMSSA in dimensions 20, 50, and 80 are nearly identical. The comparative data show that the quality of the SSA scheme before the improvement needs to be higher. The LTMSSA is more effective in terms of experimental performance.

Table 3. Algorithm scalability testing.

F		Dim = 20		Dim = 50		Dim = 80	
		Avg	Std	Avg	Std	Avg	Std
F1	LTMSSA	0	0	0	0	0	0
	SSA	4.58×10^{-29}	2.51×10^{-28}	1.47×10^{-36}	6.76×10^{-36}	9.75×10^{-33}	5.34×10^{-32}
F2	LTMSSA	0	0	0	0	0	0
	SSA	1.00×10^{-30}	4.03×10^{-30}	3.11×10^{-32}	1.62×10^{-31}	2.49×10^{-33}	1.36×10^{-32}
F3	LTMSSA	0	0	0	0	0	0
	SSA	3.21×10^{-14}	1.19×10^{-13}	1.85×10^{-13}	9.13×10^{-13}	2.19×10^{-15}	6.46×10^{-15}
F4	LTMSSA	0	0	0	0	0	0
	SSA	2.62×10^{-9}	1.27×10^{-8}	3.48×10^{-9}	1.31×10^{-8}	1.37×10^{-9}	3.70×10^{-9}
F5	LTMSSA	9.15×10^{-3}	1.19×10^{-2}	1.66×10^{-1}	2.19×10^{-1}	3.42×10^{-1}	4.04×10^{-1}
	SSA	8.51×10^{-4}	2.50×10^{-3}	3.11×10^{-3}	7.30×10^{-3}	5.83×10^{-3}	1.04×10^{-2}
F6	LTMSSA	1.30×10^{-3}	1.28×10^{-3}	1.20×10^{-2}	9.59×10^{-3}	1.96×10^{-2}	2.39×10^{-2}
	SSA	6.29×10^{-6}	1.23×10^{-5}	2.17×10^{-5}	4.13×10^{-5}	5.46×10^{-5}	1.08×10^{-4}
F7	LTMSSA	2.17×10^{-4}	1.30×10^{-4}	2.12×10^{-4}	1.64×10^{-4}	2.24×10^{-4}	1.68×10^{-4}
	SSA	2.61×10^{-4}	1.85×10^{-4}	4.06×10^{-4}	3.36×10^{-4}	2.66×10^{-4}	2.58×10^{-4}
F8	LTMSSA	-7.28×10^{3}	8.00×10^{2}	-1.86×10^{4}	1.80×10^{3}	-2.38×10^{4}	4.46×10^{3}
	SSA	-6.60×10^{3}	1.56×10^{3}	-1.65×10^{4}	4.60×10^{3}	-2.98×10^{4}	4.84×10^{3}
F9	LTMSSA	0	0	0	0	0	0
	SSA	0	0	0	0	0	0
F10	LTMSSA	8.88×10^{-16}	0	8.88×10^{-16}	0	8.88×10^{-16}	0
	SSA	3.73×10^{-15}	6.42×10^{-15}	1.72×10^{-15}	2.02×10^{-15}	1.95×10^{-15}	1.90×10^{-15}

Table 3. *Cont.*

F		Dim = 20		Dim = 50		Dim = 80	
		Avg	**Std**	**Avg**	**Std**	**Avg**	**Std**
F11	LTMSSA	0	0	0	0	0	0
	SSA	0	0	0	0	0	0
F12	LTMSSA	4.13×10^{-4}	5.23×10^{-4}	4.24×10^{-4}	3.64×10^{-4}	3.40×10^{-4}	5.04×10^{-4}
	SSA	5.82×10^{-7}	1.69×10^{-6}	6.73×10^{-7}	1.04×10^{-6}	3.54×10^{-7}	5.52×10^{-7}
F13	LTMSSA	9.01×10^{-3}	1.44×10^{-2}	1.89×10^{-2}	2.17×10^{-2}	2.37×10^{-2}	2.40×10^{-2}
	SSA	1.38×10^{-5}	3.64×10^{-5}	1.15×10^{-5}	2.04×10^{-5}	1.76×10^{-5}	3.29×10^{-5}

4.3. Population Diversity Analysis of the LTMSSA

The population initialization was randomized, and to evaluate the impact of the improvement strategy, the Sphere function was selected for the merit-seeking experiment.

$$F(x) = \sum_{k=1}^{D} x_k^2 \tag{22}$$

The initial population distribution and the distribution of individual sparrow positions after 10 and 50 iterations of the LTMSSA are shown in Figure 2a–c, and the distribution of individual sparrow positions after 10 and 50 iterations of the SSA are shown in Figure 2d.

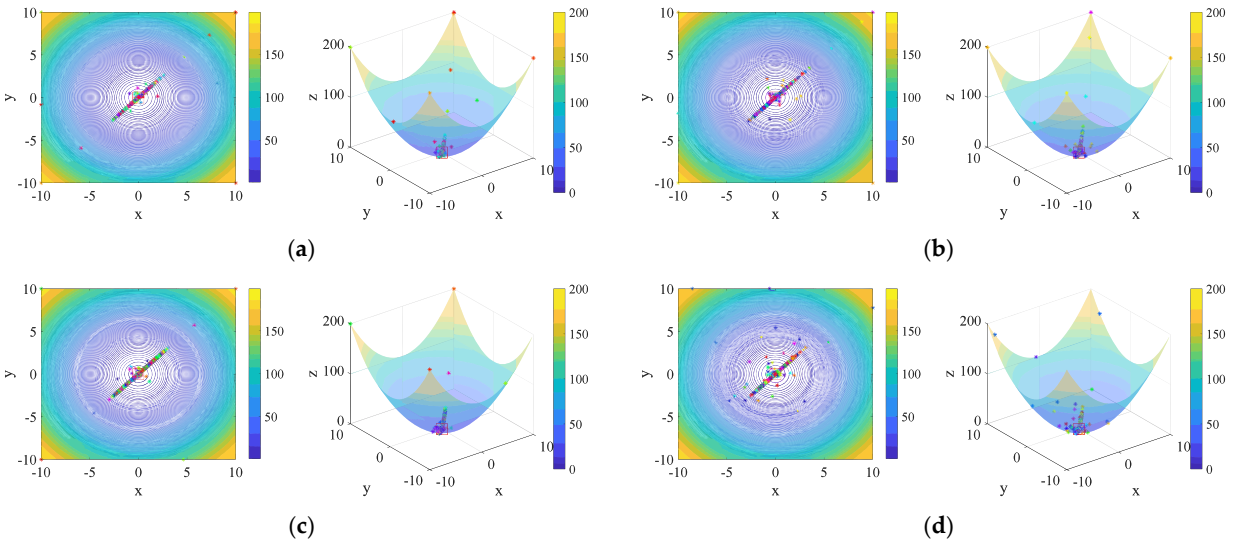

Figure 2. (**a**): Initial population distribution of the LTMSSA; (**b**): Sparrow population distribution for 10 iterations of the LTMSSA; (**c**): Sparrow population distribution for 50 iterations of the LTMSSA; (**d**): Sparrow population distribution for 50 iterations of the SSA.

From Figure 2, we can see that the population initialized by the elite chaos inverse learning strategy has good diversity, and individual sparrows have a uniform distribution around the optimal value, which gives the algorithm a good starting point for iteratively finding an optimal solution. Elite sparrows led the population to the ideal solution quicker as iterations rose. After 50 iterations, the sparrow population in the LTMSSA was more uniform and concentrated near the optimal solution compared with the distribution of sparrow individuals in the SSA, which verified the effective improvement of population diversity and population quality by the improved strategy.

4.4. Algorithm Comparison

4.4.1. Comparison of Single-Peak Test Functions

A widely adopted method in this field involves assessing the algorithm's performance by subjecting it to a test function that is equipped with predetermined global optima. Based

on the SSA mentioned, we further evaluated the algorithm on a single-peaked test function. The single-peak function is a better test for local exploitation capabilities. The optimization results for the LTMSSA, SSA, SSSA, FSSA, CSFSSA, GWO, PSO, and BOA are given in Table 4 for running 30 independent experiments.

Table 4. Unimodal function optimization results.

F		LTMSSA	SSA	SSSA	FSSA	CSFSSA	GWO	PSO	BOA
F1	Avg	0	2.53×10^{-32}	1.08×10^{-30}	1.37×10^{-32}	2.43×10^{-30}	1.32×10^{-27}	1.18×10^{-5}	7.77×10^{-11}
	Std	0	1.38×10^{-31}	5.91×10^{-30}	5.31×10^{-32}	1.33×10^{-29}	1.95×10^{-27}	2.98×10^{-5}	8.62×10^{-12}
F2	Avg	0	5.19×10^{-31}	2.67×10^{-37}	1.18×10^{-34}	3.39×10^{-37}	8.15×10^{-17}	1.68×10^{-1}	2.25×10^{-8}
	Std	0	2.84×10^{-30}	1.46×10^{-36}	4.37×10^{-34}	1.86×10^{-36}	6.37×10^{-17}	5.36×10^{-1}	8.11×10^{-9}
F3	Avg	0	4.74×10^{-13}	2.98×10^{-8}	1.18×10^{-14}	1.48×10^{-7}	6.66×10^{-6}	7.36×10	6.33×10^{-11}
	Std	0	2.56×10^{-12}	8.45×10^{-8}	5.97×10^{-14}	6.79×10^{-7}	1.77×10^{-5}	4.88×10	7.40×10^{-12}
F4	Avg	0	6.07×10^{-9}	1.78×10^{-7}	6.74×10^{-9}	3.38×10^{-6}	8.38×10^{-7}	1.52	3.58×10^{-8}
	Std	0	2.40×10^{-8}	9.37×10^{-7}	1.51×10^{-8}	1.04×10^{-5}	1.02×10^{-6}	7.02×10^{-1}	4.18×10^{-9}
F5	Avg	6.43×10^{-2}	1.15×10^{-3}	4.70×10^{-1}	1.62×10^{-3}	1.51	2.70×10	5.89×10	2.89×10
	Std	9.91×10^{-2}	2.72×10^{-3}	1.04	2.77×10^{-3}	3.63	8.32×10^{-1}	3.47e+01	2.07×10^{-1}
F6	Avg	4.11×10^{-3}	2.32×10^{-5}	1.18×10^{-1}	1.45×10^{-5}	3.50×10^{-2}	8.64×10^{-1}	2.59×10^{-2}	5.34
	Std	4.28×10^{-3}	4.22×10^{-5}	8.42×10^{-2}	3.03×10^{-5}	1.46×10^{-2}	3.79×10^{-1}	9.38×10^{-2}	6.90×10^{-1}
F7	Avg	1.75×10^{-4}	3.84×10^{-4}	7.52×10^{-4}	1.95×10^{-3}	4.28×10^{-3}	1.82×10^{-3}	6.69×10^{-2}	2.34×10^{-3}
	Std	1.60×10^{-4}	3.96×10^{-4}	1.89×10^{-3}	1.02×10^{-3}	4.21×10^{-3}	1.10×10^{-3}	3.76×10^{-2}	8.92×10^{-4}

Figure 3 provides a visual representation of the convergence curves for each algorithm, illustrating their performance on the single-peak test functions.

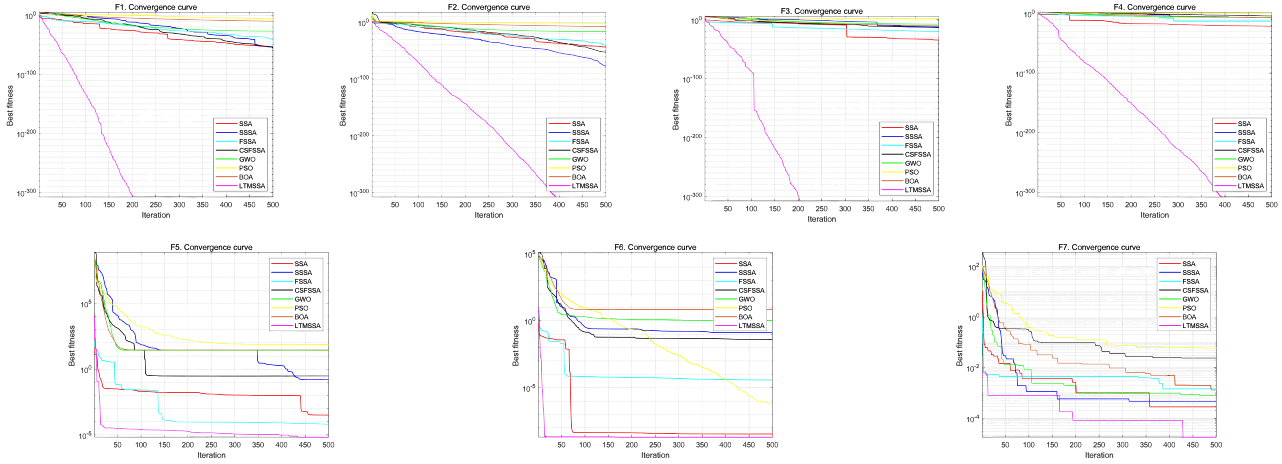

Figure 3. Convergence curves of each algorithm on the single-peak test functions.

The F1–F7 optimum algorithm is summarized in Table 4. The LTMSSA performs better on the single-peak test functions contrary to additional algorithms. F1, F2, F3, F4, and F7 have a minimal LTMSSA mean and standard deviation. The SSA and FSSA had lower mean values as well as standard deviations than the LTMSSA for F5 and F6. Figure 3 shows that the LTMSSA achieves optimal solutions in all seven single-peak test functions, and the convergence speeds and accuracy exceed those of the SSA, FSSA, and other competitors.

4.4.2. Comparison of Multi-Peak Test Functions

The multi-peak test functions include several local optima, making the global optimum difficult to identify. Consequently, an algorithm's capacity to investigate and step outside local solutions can be assessed more comprehensively. Table 5 gives the optimization results of the SSA, SSSA, FSSA, CSFSSA, GWO, PSO, and BOA on the multi-peak test function by running 30 independent experiments.

Table 5. Multimodal function optimization results.

F		LTMSSA	SSA	SSSA	FSSA	CSFSSA	GWO	PSO	BOA
F8	Avg	-1.05×10^{4}	-9.03×10^{3}	-8.60×10^{3}	-3.20×10^{3}	-3.05×10^{3}	-6.17×10^{3}	-5.39×10^{3}	-4.09×10^{3}
	Std	1.25×10^{3}	2.43×10^{3}	2.12×10^{3}	2.23×10^{3}	3.56×10^{3}	8.61×10^{3}	1.49×10^{3}	4.13×10^{3}
F9	Avg	0	0	7.51	0	3.60	2.32	6.42×10	3.91×10
	Std	0	0	3.84×10	0	1.77×10	2.97	1.46×10	7.99×10
F10	Avg	8.88×10^{-16}	1.72×10^{-15}	3.26×10^{-15}	1.84×10^{-15}	4.57×10^{-14}	1.01×10^{-13}	1.85	2.81×10^{-8}
	Std	0	1.53×10^{-15}	1.23×10^{-14}	1.60×10^{-15}	2.45×10^{-13}	1.51×10^{-14}	8.85×10^{-1}	5.16×10^{-9}
F11	Avg	0	0	3.56×10^{-3}	0	0	3.13×10^{-3}	5.64×10^{-2}	1.21×10^{-11}
	Std	0	0	1.95×10^{-2}	0	0	7.76×10^{-3}	8.70×10^{-2}	1.33×10^{-11}
F12	Avg	7.07×10^{-4}	9.13×10^{-7}	7.79×10^{-3}	4.10×10^{-7}	7.29×10^{-4}	4.32×10^{-2}	4.17×10^{-1}	5.25×10^{-1}
	Std	6.17×10^{-4}	1.62×10^{-6}	7.81×10^{-3}	4.59×10^{-7}	7.18×10^{-4}	3.90×10^{-2}	7.49×10^{-1}	1.54×10^{-1}
F13	Avg	1.33×10^{-2}	2.07×10^{-5}	1.06×10^{-1}	1.20×10^{-5}	3.58×10^{-2}	5.69×10^{-1}	2.05×10^{-1}	2.81
	Std	1.14×10^{-2}	6.86×10^{-5}	1.28×10^{-1}	3.27×10^{-5}	4.32×10^{-2}	2.25×10^{-1}	7.13×10^{-1}	3.08×10^{-1}

Figure 4 illustrates the convergence patterns of the algorithms mentioned earlier when applied to the multi-peak test functions.

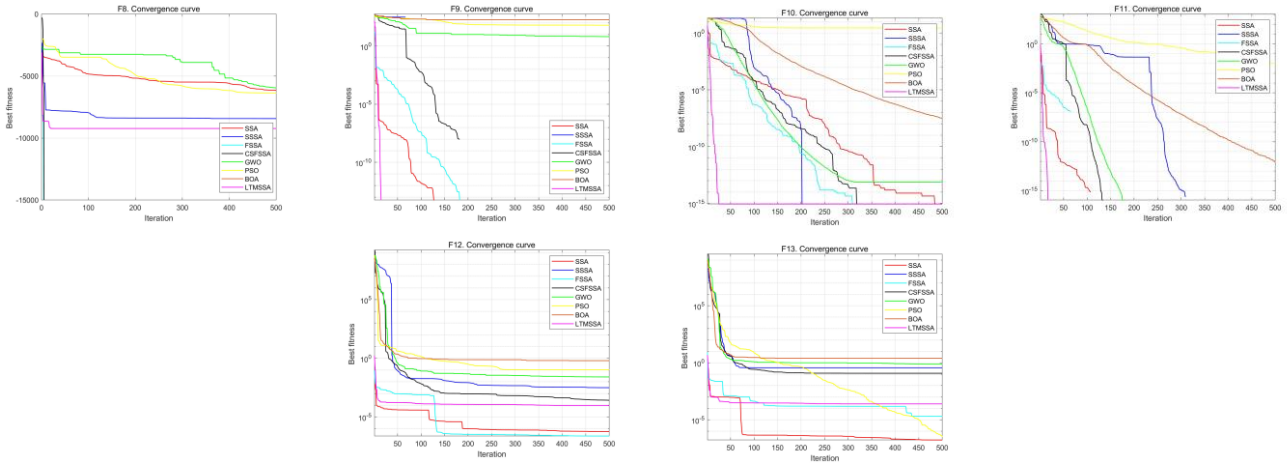

Figure 4. Convergence curves of each algorithm on the multi-peak test functions.

Table 5 shows that both the Avg and Std of the LTMSSA have the lowest values on F8-F11 and rank first overall in the multi-peak test functions. The LTMSSA converges significantly faster than other methods, as seen in Figure 4. Based on the findings, the LTMSSA solution has high accuracy and does not eventually fall into the local optimum. In particular, in F8, F9, F10 and F11, the LTMSSA shows a better exploration mechanism than other methods. The convergence accuracy of the LTMSSA is not optimal from F12 and F13, but the rate at which the convergence occurs gradually increases, leading to the eventual discovery of the optimal outcome. Therefore, the overall optimization effect of the LTMSSA is more substantial.

4.4.3. Comparison of Fixed-Dimensional Test Functions

Finding the global optimum of these test functions needs a well-balanced algorithm that is constantly adapting to new information. In order to validate the sparrow search algorithm's ability to explore globally and exploit locally, we specifically chose fixed-dimensional test functions for testing purposes. Based on 30 independent experiments, the optimization outcomes of each algorithm regarding fixed-dimensional test functions are listed in Table 6.

Table 6. Fixed dimensional functions optimization results.

F		LTMSSA	SSA	SSSA	FSSA	CSFSSA	GWO	PSO	BOA
F14	Avg	2.64	4.34	4.84	2.80	5.30	5.46	2.77	1.10
	Std	3.07	4.46	3.61	3.23	4.51	4.61	2.23	2.59×10^{-1}
F15	Avg	3.40×10^{-4}	3.98×10^{-4}	8.24×10^{-3}	3.54×10^{-4}	4.20×10^{-4}	4.46×10^{-3}	5.80×10^{-4}	3.90×10^{-4}
	Std	4.75×10^{-5}	2.44×10^{-4}	2.00×10^{-2}	1.05×10^{-4}	1.41×10^{-4}	8.09e $\times 10^{-3}$	2.52×10^{-4}	6.34e $\times 10^{-5}$
F16	Avg	-1.03	-1.03	-8.68×10^{-1}	-1.03	-4.89×10^{-1}	-1.03	-1.03	-1.31×10^{4}
	Std	1.053×10^{-9}	7.65×10^{-16}	3.32×10^{-1}	5.44×10^{-16}	3.91×10^{-1}	2.56×10^{-8}	6.39×10^{-16}	1.22×10^{4}
F17	Avg	3.98×10^{-1}	3.98×10^{-1}	4.04×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.99×10^{-1}
	Std	1.48×10^{-6}	5.78×10^{-7}	3.01×10^{-2}	2.84×10^{-6}	1.40×10^{-6}	1.74×10^{-6}	0	6.83×10^{-4}
F18	Avg	3.00	1.29×10	1.34×10	3.00	2.91×10	3.90	3.00	3.08
	Std	9.50×10^{-13}	1.32×10	1.85×10	1.71×10^{-3}	1.32×10	4.93	1.08×10^{-12}	2.47×10^{-1}
F19	Avg	-3.86	-3.81	-3.82	-3.86	-3.86	-3.86	-3.84	-1.87×10^{18}
	Std	1.93×10^{-3}	1.96×10^{-1}	6.66×10^{-2}	2.07×10^{-3}	2.54×10^{-3}	2.71×10^{-3}	1.41×10^{-1}	1.38×10^{19}
F20	Avg	-3.28	-3.27	-3.11	-3.28	-3.28	-3.23	-3.27	-3.01
	Std	2.41×10^{-2}	7.17×10^{-2}	1.93×10^{-1}	5.96×10^{-2}	5.39×10^{-2}	9.67×10^{-2}	5.99×10^{-2}	1.14×10^{-1}
F21	Avg	-1.01×10	-7.43	-6.34	-1.01×10	-9.81	-9.28	-6.36	-4.83
	Std	7.13×10^{-2}	2.59	3.13	7.04×10^{-1}	1.44	1.99	3.60	4.80×10^{-1}
F22	Avg	-1.04×10	-7.75	-6.06	-1.04×10	-1.00×10	-1.04×10	-6.70	-4.47
	Std	2.80×10^{-2}	2.70	3.21	1.67×10^{-1}	1.55	9.70×10^{-1}	3.60	3.81×10^{-1}
F23	Avg	-1.05×10	-7.11	-5.52	-1.05	-9.76	-9.99	-5.98	-4.57
	Std	2.62×10^{-2}	2.65	3.12	8.21×10^{-2}	2.05	2.06	3.85	8.70×10^{-1}

The convergence curves for each method on the fixed-dimensional test functions are displayed in Figure 5.

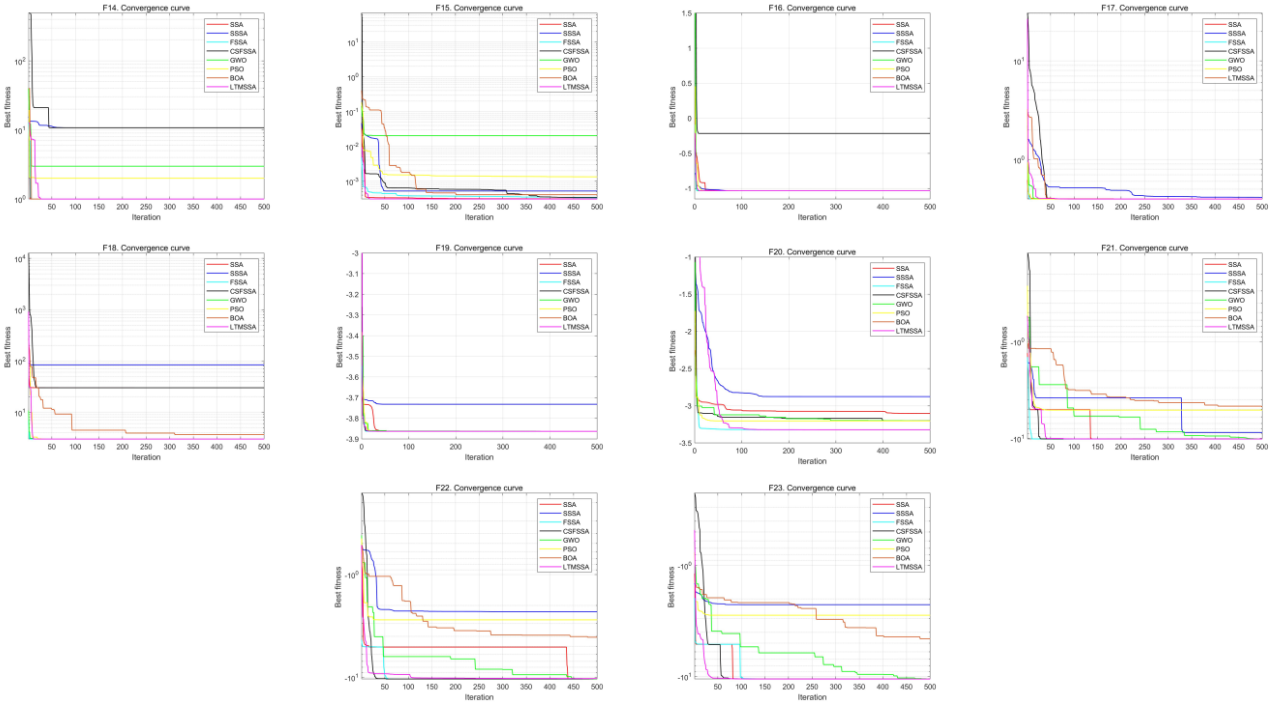

Figure 5. Convergence curves of each algorithm on the fixed-dimensional test functions.

Table 6 shows excellent performance on the fixed-dimension LTMSSA test function as well. On F14, the Avg and Std of the BOA outperform the LTMSSA, but the convergence plot of F14 shows that the LTMSSA converges faster. In addition, on F16 and F17, although the Std of the LTMSSA is not optimal, the Avg is the lowest among all algorithms. On F15, F18, F19, F20, F21, F22, and F23, both the Avg and Std of the LTMSSA are optimal. From Figure 5, all fixed dimensional test functions find the optimal values quickly and have a high convergence rate.

4.4.4. Optimal Value of Each Algorithm

For each algorithm, 30 experiments were carried out in every benchmark function to ascertain the best possible outcome. A box plot analysis was also carried out to confirm the long-term viability and converging of the LTMSSA. The box plot shows the maximum, minimum, upper, lower, median, and outliers. In Figure 6, each box plot has "◆" for outliers, "–" for medians, upper and lower quartiles at the ends of the rectangular boxes, and "-" for maximum or minimum values.

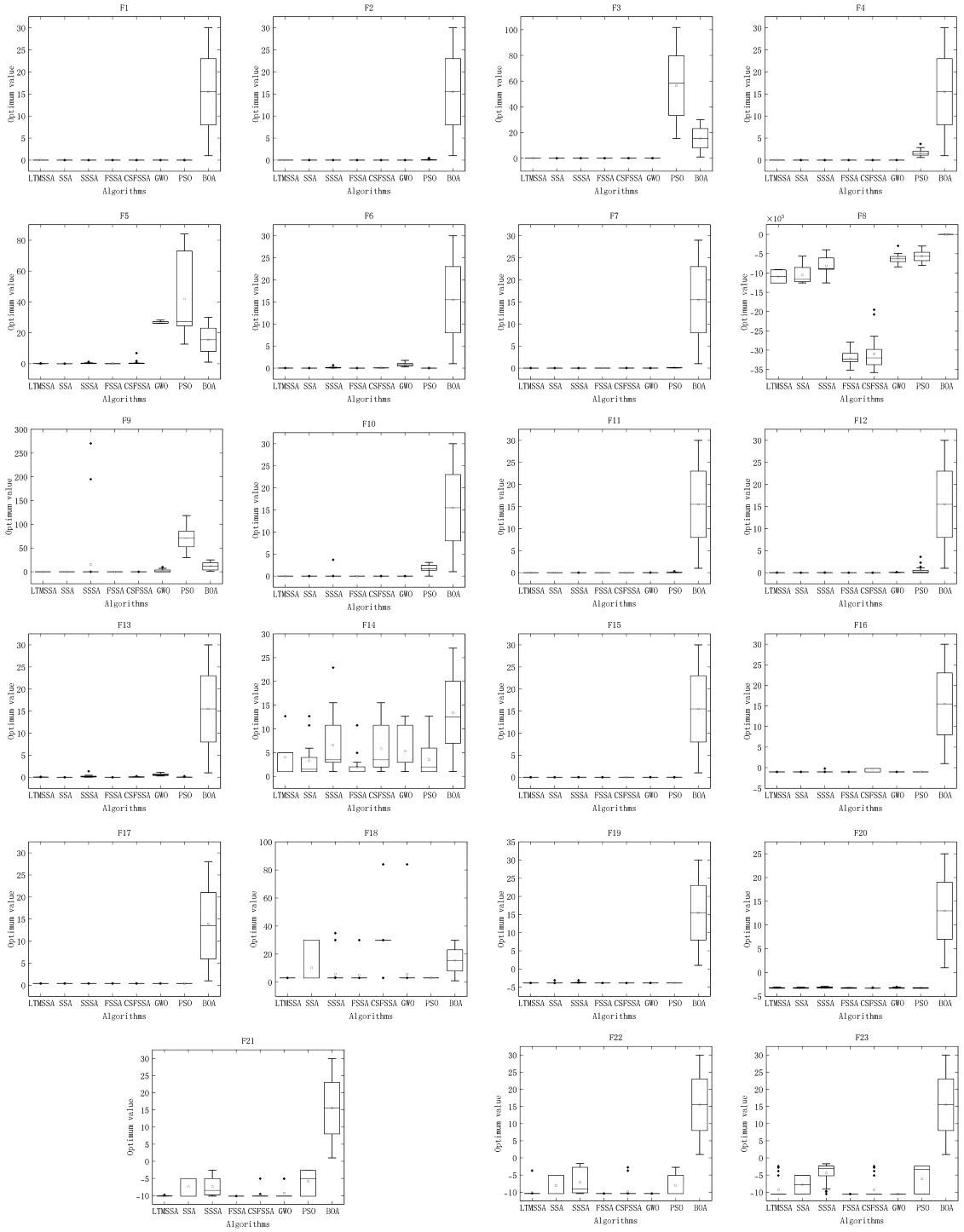

Figure 6. Box plot of 23 benchmark functions.

From the box plot, the LTMSSA has the most stable optimal value in the iterations and the median is closest to the optimal value. From the convergence value, we can indicate that the LTMSSA algorithm has stronger robustness. In addition, in each benchmark function, the LTMSSA has fewer outliers compared to other algorithms.

The radar plots are plotted below after ranking according to the optimal values to evaluate the comprehensive optimization capability of the LTMSSA.

Optimal performance is achieved when the algorithm's total score is minimized, and its graph representation occupies a smaller area. Among the eight algorithms, the LTMSSA emerges as the top scorer with exceptional efficiency in Figure 7. This can be attributed to its ability to enclose the smallest area among the single-peak, multi-peak, and fixed-dimension test functions.

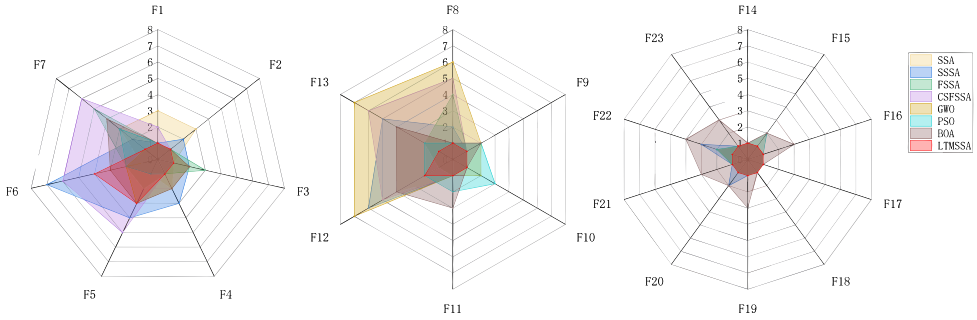

Figure 7. Radar map of 23 benchmark functions.

4.5. Discussion

The LTMSSA introduced in this research demonstrates superior performance compared to the SSA, SSSA, FSSA, CSFSSA, GWO, PSO, and BOA in optimizing the functions across the conducted experiments. Through a comprehensive integration of convergence performance and optimization performance, the findings were thoroughly analyzed and subsequently discussed. Despite the fact that the LTMSSA had lower average and standard deviation values compared to SSA for functions F5, F6, F12, and F13, it exhibited significantly higher convergence rates. On F14, the Avg and Std of the BOA outperformed the LTMSSA, but the convergence plot of F14 shows that the LTMSSA converges faster. In addition, on F16 and F17, although the Std of the LTMSSA is not optimal, the Avg is the lowest among all algorithms. The LTMSSA has the highest ranking in the number of optimal values obtained among the 23 tested functions and has the most stable optimal values in the iterative process. Based on the aforementioned analysis, the LTMSSA has the ability to swiftly achieve the most accurate global solution while ensuring rapid convergence. The findings indicate that the algorithm possesses a robust ability to explore both wide-ranging and localized areas, consistently delivering effective optimization outcomes. The achieved outcomes cannot be separated from the initial population that has been fine-tuned through comprehensive exploration, as well as the ongoing adjustments made throughout the iterative process. Undoubtedly, there are still some aspects of the algorithm that require refinement and improvement. To provide an instance, the algorithm's effectiveness in optimization is influenced by random numbers, thus slightly compromising its overall accuracy. There is potential for enhancing the existing approach to adjusting parameters and weights, as the current dynamic scheme may not offer the most effective solution. Additionally, it is crucial to verify the algorithm's effectiveness across various iterations and diverse population sizes.

5. Engineering Design Issues

5.1. Welded Beam Design

Welded beam design (WBD) is an issue of cost minimization to reduce production expenses [67]. It is also a typical nonlinear programming problem with four design variables: height (t), thickness (b), weld width (h), and length (l), as shown in Figure 8.

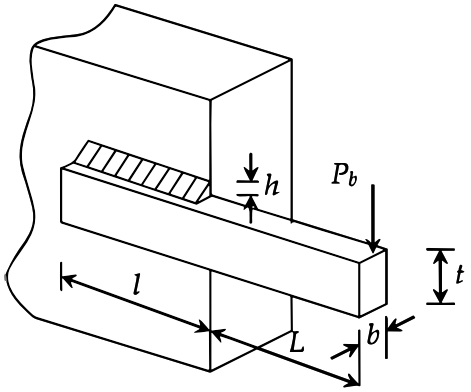

Figure 8. Welded beam design problem.

To express the mathematical model, we can utilize the following representation.
Variables:

$$\vec{z} = [z_1 z_2 z_3\ z_4] = [\text{hltb}] \tag{23}$$

Objective function:

$$f\left(\vec{z}\right) = 1.10471 z_1^2 z_2 + 0.04811 z_3 z_4 (14.0 + z_2) \tag{24}$$

where $f\left(\vec{z}\right)$ denotes the total cost, and cost-effectiveness is needed.
The decision variables take a range of values:

$$0.1 \leq z_1 \leq 2 \tag{25}$$

$$0.1 \leq z_2 \leq 10 \tag{26}$$

$$0.1 \leq z_3 \leq 10 \tag{27}$$

$$0.1 \leq z_4 \leq 2 \tag{28}$$

Constraints:

$$s_1\left(\vec{z}\right) = \tau\left(\vec{z}\right) - \tau_{max} \leq 0 \tag{29}$$

$$s_2\left(\vec{z}\right) = \sigma\left(\vec{z}\right) - \sigma_{max} \leq 0 \tag{30}$$

$$s_3\left(\vec{z}\right) = \delta\left(\vec{z}\right) - \delta_{max} \leq 0 \tag{31}$$

$$s_4\left(\vec{z}\right) = z_1 - z_4 \leq 0 \tag{32}$$

$$s_5\left(\vec{z}\right) = P - P_c\left(\vec{z}\right) \leq 0 \tag{33}$$

$$s_6\left(\vec{z}\right) = 0.125 - z_1 \leq 0 \tag{34}$$

$$s_7\left(\vec{z}\right) = 1.10471 z_1^2 + 0.04811 z_3 z_4 (14.0 + z_2) - 5.0 \leq 0 \tag{35}$$

The expressions of each function in the constraints can be referred to as Equations (36)–(42).

$$\tau\left(\vec{z}\right) = \sqrt{\tau'^2 + 2\tau'\tau''\left(z_2/R\right) + \left(\tau''\right)^2} \tag{36}$$

$$\tau' = \frac{P}{\sqrt{2}z_1 z_2} \tag{37}$$

$$\tau'' = MR/J \tag{38}$$

$$M = p(L + z_2/2) \tag{39}$$

$$R = \sqrt{\frac{\left[z_2^2 + (z_1 + z_3)^2\right]}{4}} \tag{40}$$

$$J = 2\{\sqrt{2}z_1 z_2[\frac{z_2^2}{12} + \frac{(z_1 + z_3)^2}{14}]\} \tag{41}$$

$$P_c(\vec{z}) = \frac{4.013Ez_3 z_4^2}{6L^2}(1 - \frac{z_3\sqrt{E}}{8LG}) \tag{42}$$

where, $\sigma_{max} = 30000$ psi, $P = 6000$ lb, $L = 14$ in., $\delta_{max} = 0.25$ in., $E = 3 \times 10^6$ psi, $\tau_{max} = 136000$ psi, and $G = 1.2 \times 10^7$ psi.

Table 7 shows that the LTMSSA achieves the lowest cost. It shows that the LTMSSA performs well in the challenging task of designing welded beams.

Table 7. Parameter optimization comparison of welded beam design problem.

Algorithms	Optimal Values for Variables				Optimum Value
	h	l	t	b	
LTMSSA	0.3345	2.0109	8.2792	0.2451	1.8117
SSA	0.2890	2.5890	7.4037	0.3065	2.0499
SSSA	0.2798	4.5590	9.3352	0.2043	2.0970
FSSA	0.3888	1.7238	8.1146	0.2551	1.8541
CSFSSA	0.4953	2.1360	5.5490	0.5495	2.9458
GWO	0.3007	2.1222	9.0381	0.2058	1.9549
PSO	0.1000	7.0875	9.0366	0.2057	1.9769
BOA	0.3322	3.2012	6.6256	0.3865	2.5096

5.2. Reducer Design

Reducers are often used in mechanical systems and are an important component of gearboxes [68]. The task of designing a reducer involves finding the optimal solution to minimize its size, with 11 constraints in which the reducer's weight must be lowered. Seven variables are involved in the problem, including the tooth width b, the gear module m, the amount of teeth z in the pinion, the length l_1 of the first shaft between bearings, the length l_2 of the second shaft between bearings, and the diameters d_1 and d_2, as shown in Figure 9.

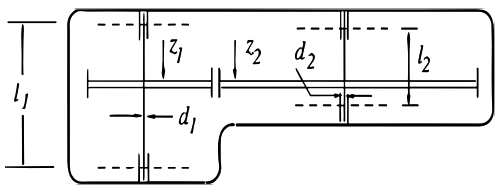

Figure 9. Reducer design problem.

Variables:

$$\vec{x} = [x_1\ x_2 x_3 x_4\ x_5\ x_6 x_7] = [\mathrm{bmz}l_1\ l_2 d_1\ d_2] \tag{43}$$

Objective function:

$$f(X) = 0.7854x_1x_2^2\left(3.3333x_3^2 + 14.9334x_3 - 43.0934\right) - 1.508x_1\left(x_6^2 + x_7^2\right) + 7.4777\left(x_6^3 + x_7^3\right) + 0.7854\left(x_4x_6^2 + x_5x_7^2\right) \tag{44}$$

Constraints:

$$d_1(X) = \frac{27}{x_1x_2^2x_3} - 1 \leq 0 \tag{45}$$

$$d_2(X) = \frac{397.5}{x_1x_2^2x_3^2} - 1 \leq 0 \tag{46}$$

$$d_3(X) = \frac{1.93x_4^3}{x_2x_6^4x_3} - 1 \leq 0 \tag{47}$$

$$d_4(X) = \frac{1.93x_5^3}{x_2x_7^4x_3} - 1 \leq 0 \tag{48}$$

$$d_5(X) = \frac{\sqrt{(745x_4/(x_2x_3))^2 + 16.9\times10^6}}{110x_6^3} - 1 \leq 0 \tag{49}$$

$$d_6(X) = \frac{\sqrt{(745x_5/(x_2x_3))^2 + 157.5\times10^6}}{85x_7^3} - 1 \leq 0 \tag{50}$$

$$d_7(X) = \frac{x_2x_3}{40} - 1 \leq 0 \tag{51}$$

$$d_8(X) = \frac{5x_2}{x_1} - 1 \leq 0 \tag{52}$$

$$d_9(X) = \frac{x_1}{12x_2} - 1 \leq 0 \tag{53}$$

$$d_{10}(X) = \frac{1.5x_6 + 1.9}{x_4} - 1 \leq 0 \tag{54}$$

$$d_{11}(X) = \frac{1.1x_7 + 1.9}{x_5} - 1 \leq 0 \tag{55}$$

Boundary constraints:

$$2.6 \leq x_1 \leq 3.6 \tag{56}$$

$$0.7 \leq x_2 \leq 0.8 \tag{57}$$

$$17 \leq x_3 \leq 28 \tag{58}$$

$$7.3 \leq x_4 \leq 8.3 \tag{59}$$

$$7.8 \leq x_5 \leq 8.3 \tag{60}$$

$$2.9 \leq x_6 \leq 3.9 \tag{61}$$

$$5.0 \leq x_7 \leq 5.5 \tag{62}$$

Table 8 lists optimization outcomes. The LTMSSA optimizes reducer design and improves optimization results.

Table 8. Parameter optimization comparison of the reducer design problem.

Algorithms	Optimal Values for Variables							Optimum
	b	m	z	l_1	l_2	d_1	d_2	Value
LTMSSA	3.6538	0.7000	15.0919	6.8289	8.1556	3.3513	5.2872	2733.9
SSA	3.2104	0.7375	15.4116	7.2710	7.4555	3.3662	5.2867	2981.9
SSSA	3.5327	0.7000	15.1698	8.2012	7.9430	3.0855	5.4437	3106.5
FSSA	3.3172	0.7079	15.7402	7.5154	7.5370	3.3576	5.2868	2897.4
CSFSSA	3.6443	0.7000	16.7168	8.5967	7.9260	3.3535	5.2868	3017.5
GWO	3.4837	0.7000	17.000	7.5806	7.6425	3.3577	5.2877	3002.1

6. Conclusions

A notable limitation of the SSA lies in its inclination to get trapped in local optimum solutions, coupled with the sluggish pace at which it converges during iterations. These factors significantly curtail its effectiveness in various practical scenarios. To overcome the drawbacks of the original algorithm, this paper puts forth a new solution called the logistic-tent hybrid chaotic maps-based multi-strategy mad sparrow search algorithm (LTMSSA). By utilizing a logistic-tent hybrid chaotic algorithm, the population is effectively initialized while also ensuring a balanced and unpredictable distribution across the board. First, the LTMSSA employs an elite chaotic backward learning strategy and an improved discoverer-follower scaling factor, resulting in improved quality and diversity. Secondly, the LTMSSA updates the positions of discoverers and followers by the crazy operator and the Lévy flight strategy to expand the selection of target followers. Finally, during the optimization search of the algorithm, the LTMSSA introduces tent mixing and Corsi variable perturbation strategies to improve the ability of populations to jump out of local optimum. The proposed LTMSSA algorithm is compared with other classical metaheuristic algorithms and SSA variants. Based on the optimization experiments carried out on 23 benchmark functions, it is evident that the proposed LTMSSA provides a noteworthy solution to the drawbacks of the SSA. Notably, it surpasses other SSA variants and advanced algorithms in terms of both iterative convergence and optimization performance. The findings from the optimization of the welded beam and reducer clearly demonstrate that the LTMSSA outperforms various classical metaheuristic algorithms in terms of optimization performance. Experimental results demonstrate that the suggested algorithm effectively strikes a balance between utilizing and exploring, adapting the algorithm's global and local search, enabling swift iterations, and effortlessly attaining global optimization.

Author Contributions: Conceptualization, X.J. and Y.G.; methodology, X.J.; software, X.J.; validation, X.J., W.W., Y.G. and S.L.; formal analysis, X.J.; investigation, Y.G.; resources, X.J.; data curation, X.J.; writing—original draft preparation, X.J.; writing—review and editing, X.J.; visualization, X.J.; supervision, Y.G. and S.L.; project administration, W.W.; funding acquisition, W.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Fundamental Research Funds for the Central Universities, grant number 2572019BL04, and the Scientific Research Foundation for the Returned Overseas Chinese Scholars of Heilongjiang Province, grant number LC201407.

Data Availability Statement: Data are available upon request from the corresponding author.

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

References

1. Yapici, H.; Cetinkaya, N. A new meta-heuristic optimizer: Pathfinder algorithm. *Appl. Soft Comput.* **2019**, *78*, 545–568. [CrossRef]
2. Yazdani, S.; Nezamabadi-Pour, H.; Kamyab, S. A gravitational search algorithm for multimodal optimization. *Swarm Evol. Comput.* **2014**, *14*, 69–85. [CrossRef]
3. Tan, Y.; Zhu, Y. Fireworks algorithm for optimization. In *Advances in Swarm Intelligence, Proceedings of the International Conference in Swarm Intelligence, Beijing, China, 12–15 June 2010*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 355–364. [CrossRef]

4. Karimi, H.; Kani, I.M. Finding the worst imperfection pattern in shallow lattice domes using genetic algorithms. *J. Build. Eng.* **2019**, *23*, 107–113. [CrossRef]
5. Geem, Z.W.; Kim, J.H.; Loganathan, G.V. A new heuristic optimization algorithm: Harmony search. *Simulation* **2001**, *76*, 60–68. [CrossRef]
6. 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]
7. Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. Harris hawks optimization: Algorithm and applications. *Future Gener. Comput. Syst.* **2019**, *97*, 849–872. [CrossRef]
8. Beni, G.; Wang, J. Swarm intelligence in cellular robotic systems. In *Robots and Biological Systems: Towards a New Bionics? NATO ASI Series*; Springer: Berlin/Heidelberg, Germany, 1993; pp. 703–712. [CrossRef]
9. Kennedy, J.; Eberhart, R.C. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; pp. 1942–1948. [CrossRef]
10. Eberhart, R.C.; Shi, Y. Particle swarm optimization: Developments, applications and resources. In Proceedings of the IEEE Congress on Evolutionary Computation, Seoul, Republic of Korea, 27–30 May 2001; pp. 81–86. [CrossRef]
11. Dorigo, M.; Maniezzo, V.; Colorni, A. The ant system: Optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern. B* **1996**, *26*, 29–41. [CrossRef]
12. Xia, X.; Liu, J.; Hu, Z. An improved particle swarm optimizer based on tabu detecting and local learning strategy in a shrunk search space. *Appl. Soft Comput.* **2014**, *23*, 76–90. [CrossRef]
13. Li, S.; Tan, M. A hybrid PSO-BFGS strategy for global optimization of multimodal functions. *IEEE Trans. Syst. Man Cybern. B* **2011**, *41*, 1003–1014.
14. Zhao, S.; Liang, J.J.; Suganthan, P.N. Dynamic multi-swarm particle swarm optimizer with local search for large scale global optimization. In Proceedings of the Congress on Evolutionary Computation, Singapore, 1–6 June 2008; pp. 3845–3852. [CrossRef]
15. Yang, X.S.; He, X. Bat algorithm: Literature review and applications. *Int. J. Bio-Inspired Comput.* **2013**, *5*, 141–149. [CrossRef]
16. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
17. Passino, K.M. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Syst. Mag.* **2002**, *22*, 52–67. [CrossRef]
18. Karaboga, D. *An Idea Based on Honey Bee Swarm for Numerical Optimization*; Technical Report-TR06; Erciyes University, Engineering Faculty, Computer Engineering Department: Talas, Turkey, 2005; Volume 129, pp. 2865–2874. [CrossRef]
19. Karaboga, D.; Basturk, B. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *J. Glob. Optim.* **2007**, *39*, 459–471. [CrossRef]
20. Duan, H.; Qiao, P. Pigeon-inspired optimization: A new swarm intelligence optimizer for air robot path planning. *Int. J. Intell. Comput. Cybern.* **2014**, *7*, 24–37. [CrossRef]
21. Yang, X.S. *Nature-Inspired Metaheuristic Algorithms*; Luniver Press: London, UK, 2008; pp. 1–147. [CrossRef]
22. Yang, X.S. Firefly algorithm, stochastic test functions and design optimisation. *Int. J. Bio-Inspired Comput.* **2010**, *2*, 78–84. [CrossRef]
23. Yang, X.S.; Deb, S. Cuckoo search via L'evy flights. In Proceedings of the World Congress on Nature & Biologically Inspired Computing (NaBIC '09), Coimbatore, India, 9–11 December 2009; pp. 210–214. [CrossRef]
24. Yang, X.S.; Deb, S. Engineering optimisation by cuckoo search. *Int. J. Math. Model. Numer. Optim.* **2010**, *1*, 330–343. [CrossRef]
25. Mirjalili, S.; Lewi, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [CrossRef]
26. Pan, W.T. A new fruit fly optimization algorithm: Taking the financial distress model as an example. *Knowl.-Based Syst.* **2012**, *26*, 69–74. [CrossRef]
27. Gandomi, A.H.; Alavi, A.H. Krill herd: A new bio-inspired optimization algorithm. *Commun. Nonlinear Sci. Numer. Simul.* **2012**, *17*, 4831–4845. [CrossRef]
28. Mirjalili, S. Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Comput. Appl.* **2016**, *27*, 1053–1073. [CrossRef]
29. Zhao, R.Q.; Tang, W.S. Monkey Algorithm for global numerical optimization. *J. Uncertain Syst.* **2008**, *2*, 164–175. [CrossRef]
30. Jiang, X.; Li, S. BAS: Beetle antennae search algorithm for optimization problems. *Int. J. Robot. Control* **2018**, *1*, 1–5. [CrossRef]
31. Xue, J.K.; Shen, B. A novel swarm intelligence optimization approach: Sparrow search algorithm. *Syst. Sci. Control Eng.* **2020**, *8*, 22–34. [CrossRef]
32. Liu, T.T.; Yuan, Z.; Wu, L.; Badami, B. An optimal brain tumor detection by convolutional neural network and enhanced sparrow search algorithm. *Proc. Inst. Mech. Eng. Part H J. Eng. Med.* **2021**, *235*, 459–469. [CrossRef]
33. Liu, Q.L.; Zhang, Y.; Li, M.Q.; Zhang, Z.Y.; Chao, N.; Shang, J. Multi-UAV Path Planning Based on Fusion of Sparrow Search Algorithm and Improved Bioinspired Neural Network. *IEEE Access* **2021**, *9*, 124670–124681. [CrossRef]
34. Liu, L.N.; Nan, X.Y.; Shi, Y.F. Improved sparrow search algorithm for solving job shop scheduling problem. *Comput. Appl. Res.* **2021**, *38*, 3634–3639. [CrossRef]
35. Wei, P.F.; Fan, S.Z.; Shi, R.J.; Wang, W.Q.; Cheng, C.J. Short-term photovoltaic power prediction based on improved sparrow search algorithm with optimized support vector machine. *Therm. Power Gener.* **2021**, *50*, 74–79. [CrossRef]
36. Tang, A.D.; Han, T.; Xu, D.W.; Xie, L. A chaotic sparrow search algorithm-based approach for UAV trajectory planning. *Comput. Appl.* **2021**, *41*, 2128–2136. [CrossRef]

37. Tang, A.D.; Han, T.; Xu, D.W.; Xie, L. Chaotic sparrow search algorithm based on hierarchy and Brownian motion. *J. Air Force Eng. Univ. (Nat. Sci. Ed.)* **2021**, *22*, 96–103. [CrossRef]
38. Zhang, S.D.; Zhang, J.Y.; Wang, Z.H.; Li, Q.H. Regression prediction of material grinding particle size based on improved sparrow search algorithm to optimize BP neural network. In Proceedings of the 2021 2nd International Symposium on Computer Engineering and Intelligent Communications, Nanjing, China, 6–8 August 2021; pp. 216–219. [CrossRef]
39. Chen, X.X.; Huang, X.Y.; Zhu, D.L.; Qiu, Y.X. Research on chaotic flying sparrow search algorithm. *J. Phys. Conf. Ser.* **2021**, *1848*, 012044. [CrossRef]
40. Ou-yang, C.T.; Zhu, D.L. Research on multi-strategy improved sparrow search algorithm incorporating K-means. *Electro-Opt. Control* **2021**, *28*, 11–16. [CrossRef]
41. Ou-Yang, C.T.; Liu, Y.J.; Zhu, D.L. An adaptive chaotic sparrow search optimization algorithm. In Proceedings of the 2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering, Nanchang, China, 26–28 March 2021; pp. 76–82. [CrossRef]
42. Mao, Q.H.; Zhang, Q. An improved sparrow algorithm incorporating Corsi variation and backward learning. *Comput. Sci. Explor.* **2021**, *15*, 1155–1164.
43. Fu, H.; Liu, H. Improved sparrow search algorithm with multi-strategy fusion and its application. *Control Decis. Mak.* **2022**, *37*, 87–96. [CrossRef]
44. Duan, Y.X.; Liu, C.Y. A sparrow search algorithm based on Sobol sequences and vertical and horizontal crossover strategies. *Comput. Appl.* **2022**, *42*, 36–43. [CrossRef]
45. Chen, G.; Zeng, G.; Huang, B.; Liu, J. Sparrow search algorithm based on spiral exploration and adaptive hybrid mutation. *J. Chin. Comput. Syst.* **2023**, *44*, 779–786.
46. Yan, S.Q.; Yang, P.; Zhu, D.L.; Wu, F.X.; Yan, Z. Improved sparrow search algorithm based on good point set. *J. Beijing Univ. Aeronaut. Astronaut.* **2022**, *2021*, 1–13. [CrossRef]
47. He, G.S.; Dong, Z.; Sun, M. Parameter identification of superheated steam temperature model based on hybrid quantum sparrow algorithm. *J. N. China Univ. Electr. Power (Nat. Sci. Ed.)* **2023**, *1*, 92–100. [CrossRef]
48. Wu, W.S.; Tian, L.Q.; Wang, Z.G.; Zhang, Y.; Wu, J.I.; Gui, F. A multi-objective sparrow optimization algorithm based on a novel non-dominated ranking. *Comput. Appl. Res.* **2022**, *39*, 2012–2019. [CrossRef]
49. Liu, R.; Mo, W.B. Enhanced sparrow search algorithm and its engineering optimization application. *Small Microcomput. Syst.* **2022**, 1–10. [CrossRef]
50. Ma, W.; Zhu, X. A sparrow search algorithm based on Lévy flight perturbation strategy. *J. Appl. Sci.* **2022**, *40*, 116–130.
51. Tian, L.; Liu, S. A hybrid sparrow and arithmetic optimization algorithm incorporating Hamiltonian graphs. *Comput. Sci. So* **2022**, *2022*, 1–13. [CrossRef]
52. Yang, L.; Li, Z.; Wang, D.S.; Hong, M.; Wang, Z.B. Software defects prediction based on hybrid particle swarm optimization and sparrow search algorithm. *IEEE Access* **2021**, *9*, 60865–60879. [CrossRef]
53. Li, F.; Lin, Y.X.; Zou, L.H.; Zhong, L.Y. Improved sparrow search algorithm applied to path planning of mobile robot. In Proceedings of the 2021 International Conference on Computer Information Science and Artificial Intelligence, Kunming, China, 17–19 September 2021; pp. 294–300.
54. Liu, C.H.; He, Q. Improved search mechanism of simplex method to guide sparrow search algorithm. *Comput. Eng. Sci.* **2022**, *2022*, 9950161. [CrossRef]
55. Zhou, Y.Q.; Zhang, H. New 3D affine transform applied to image encryption. *Comput. Age* **2022**, *2022*, 31–35. [CrossRef]
56. Li, H.M.; Li, T.; Li, C.L. A new discrete memory-resistive chaotic system and its image encryption application. *J. Hunan Inst. Technol. (Nat. Sci. Ed.)* **2022**, *35*, 20–30. [CrossRef]
57. Yang, K.X.; Wu, Z.H.; Hao, R.B. Four-dimensional chaotic systems and their applications in image encryption. *Comput. Appl. Res.* **2020**, *37*, 3433–3436. [CrossRef]
58. Tang, C.H.; Wu, C.X. Logistic-Sine mapping and bit recombination for image encryption algorithms. *Intell. Comput. Appl.* **2022**, *12*, 173–179.
59. Zhang, S.N.; Li, C.M. A color image encryption algorithm based on Logistic–Sine–Cosine mapping. *Comput. Sci.* **2022**, *49*, 353–358.
60. Long, W.; Wu, T.B.; Tang, M.Z. Grey wolf optimizer algorithm based on lens imaging learning strategy. *Acta Autom. Sin.* **2020**, *46*, 2148–2164. [CrossRef]
61. Wang, X.W.; Wang, W.; Wang, Y. An adaptive bat algorithm. In *Intelligent Computing Theories and Technology—ICIC 2013*; Springer: Berlin, Heidelberg, Germany, 2013; pp. 216–223. [CrossRef]
62. Zhang, N.; Zhao, Z.D.; Bao, X.A. Gravitational search algorithm based on improved Tent chaos. *Control Decis.* **2020**, *35*, 893–900. [CrossRef]
63. Guo, Z.Z.; Wang, P.; Ma, Y.F. Whale optimization algorithm based on adaptive weight and Cauchy mutation. *Microelectron. Comput.* **2017**, *34*, 20–25. [CrossRef]
64. Li, A.L.; Quan, L.X.; Cui, G.M. A sparrow search algorithm incorporating positive cosine and Corsi variance. *Comput. Eng. Appl.* **2022**, *58*, 91–99. [CrossRef]
65. Jiang, Y.; Ma, Y.; Liang, Y.Z. Optimized OTSU lung tissue segmentation algorithm based on fractional-order sparrow search. *Comput. Sci.* **2021**, *48*, 28–32. [CrossRef]

66. Arora, S.; Singh, S. Butterfly optimization algorithm: A novel approach for global optimization. *Soft Comput.* **2019**, *23*, 715–734. [CrossRef]
67. Carlos, A.; Coello, C. Use of a self-adaptive penalty approach for engineering optimization problems. *Comput. Ind.* **2000**, *41*, 113–127. [CrossRef]
68. Sadollah, A.; Bahreininejad, A.; Eskandar, H.; Hamdi, M. Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems. *Appl. Soft Comput.* **2013**, *13*, 2592–2612. [CrossRef]