biomimetics

Article

Path Planning of Unmanned Aerial Vehicles Based on an Improved Bio-Inspired Tuna Swarm Optimization Algorithm

Qinyong Wang [1], Minghai Xu [2,*] and Zhongyi Hu [3]

[1] School of Artificial Intelligence, Zhejiang College of Security Technology, Wenzhou 325016, China
[2] School of Intelligent Manufacturing and Electronic Engineering, Wenzhou University of Technology, Wenzhou 325035, China
[3] Institute of Intelligent Information System, Wenzhou University, Wenzhou 325000, China
* Correspondence: xmhemail@126.com

Abstract: The Sine–Levy tuna swarm optimization (SLTSO) algorithm is a novel method based on the sine strategy and Levy flight guidance. It is presented as a solution to the shortcomings of the tuna swarm optimization (TSO) algorithm, which include its tendency to reach local optima and limited capacity to search worldwide. This algorithm updates locations using the Levy flight technique and greedy approach and generates initial solutions using an elite reverse learning process. Additionally, it offers an individual location optimization method called golden sine, which enhances the algorithm's capacity to explore widely and steer clear of local optima. To plan UAV flight paths safely and effectively in complex obstacle environments, the SLTSO algorithm considers constraints such as geographic and airspace obstacles, along with performance metrics like flight environment, flight space, flight distance, angle, altitude, and threat levels. The effectiveness of the algorithm is verified by simulation and the creation of a path planning model. Experimental results show that the SLTSO algorithm displays faster convergence rates, better optimization precision, shorter and smoother paths, and concomitant reduction in energy usage. A drone can now map its route far more effectively thanks to these improvements. Consequently, the proposed SLTSO algorithm demonstrates both efficacy and superiority in UAV route planning applications.

Keywords: UAV; 3D path planning; tuna swarm optimization; horizontal crossover strategy; Levy flight

Citation: Wang, Q.; Xu, M.; Hu, Z. Path Planning of Unmanned Aerial Vehicles Based on an Improved Bio-Inspired Tuna Swarm Optimization Algorithm. *Biomimetics* **2024**, *9*, 388. https://doi.org/10.3390/biomimetics9070388

Academic Editor: Yongquan Zhou

Received: 3 May 2024
Revised: 20 June 2024
Accepted: 21 June 2024
Published: 26 June 2024

1. Introduction

As unmanned aerial technology develops, unmanned aerial vehicles (UAVs) have garnered significant attention because of their ability to function in hazardous and complex environments [1]. Path planning and design, which entails choosing a safe, realistic, and effective flight path under certain constraints from a starting point to a destination, is one of the most crucial aspects of UAV mission systems [2,3]. Because of this, it is possible to see the UAV route planning problem as a difficult optimization problem that requires effective solutions [4,5]. Several solutions have been proposed by researchers to overcome UAV route planning challenges. In addition to more modern techniques like artificial potential field algorithms, they also include more traditional techniques like neural network algorithms [6], Q-learning algorithms [7], rapidly exploring random trees (RRTs) [8], and artificial potential field algorithms. However, these methods usually demand a large time and computational resource commitment. The stochastic search algorithms that underpin swarm optimization techniques are derived from natural occurrences and biological intelligence. Their fundamental idea is to abstract collective behaviors in nature, including foraging and reproduction among particular species, and then quantify these features to construct mathematical models that may be applied to a wide range of problems. Numerous clever optimization algorithms have been created, greatly enhancing optimization

techniques and enabling the resolution of combinatorial optimization problems that are challenging to resolve using conventional techniques. They also provide a novel way to explore the concepts and mechanisms of the biological world from many perspectives.

Swarm intelligence optimization methods compensate for the shortcomings of conventional optimization approaches, enabling them to address complex real-world situations successfully [9–11]. As a result, more scholars are drawing inspiration for their own swarm intelligence optimization techniques from a variety of biological phenomena [12–14]. These algorithms are well known for their ease of use and efficacy in handling a broad variety of real-world applications. They are employed to handle problems related to function optimization [15]. They draw inspiration from natural phenomena or group actions. They offer fresh viewpoints on how to approach difficult path planning issues in unmanned aerial systems. A collaborative coevolutionary spider monkey optimization technique was proposed by Zhu et al. [16] and used for obstacle avoidance in unmanned combat aerial vehicle route planning. These metaheuristic algorithms perform better when dealing with complex environment path planning challenges. Shi et al. introduced an improved adaptive grey wolf optimization (AGWO) method in [17] specifically designed to solve three-dimensional path planning issues for unmanned aerial vehicles (UAVs) operating in complicated settings. All things considered, these metaheuristic algorithms seem to be useful approaches for dealing with difficult path planning problems in UAV applications.

In recent years, these algorithms have been widely applied to handle difficult optimization problems such as path planning, picture processing, and production scheduling. Chinese researchers Xie et al. introduced the novel swarm intelligence optimization approach known as tuna swarm optimization (TSO) in 2021. It was influenced by the helical and parabolic foraging patterns seen in tuna schools [18]. The advantages of TSO's deployment, simplicity, and ease of understanding have sped up its acceptance. Although the TSO approach is relevant to UAV route planning issues, it has limitations, such as a slow convergence time, susceptibility to local optima entrapment, and instability concerns [19]. Specifically developed for UAV route planning in hazardous and complicated situations, this work proposes a sinusoidal and Levy flight-guided tuna swarm optimization (SLTSO) method that blends the golden ratio sinusoidal approach and Levy flight strategy to overcome these shortcomings. The fundamental component of the SLTSO algorithm is the use of sinusoidal strategies to update individual positions within the population. By utilizing nonlinearly decreasing search factors and weight factors, the sine–cosine algorithm is improved while maintaining a balance between local exploitation and global exploration. The algorithm then uses the Levy flight mechanism to skew the follower's location updates in relation to the best solution, broadening the search space and raising the possibility of breaking out of local optima. Through comparisons against numerous basic intelligent algorithms and other swarm intelligence optimization algorithms, as well as comparative assessments utilizing several performance metrics on benchmark test functions, the efficacy and superiority of SLTSO are established. Furthermore, the application of SLTSO to engineering optimization design challenges validates its benefits and resilience.

This study proposes an improved tuna optimization algorithm (SLTSO) based on the sine and Levy flight strategy methods. The main contributions of the research can be summed up as follows:

(1) A mathematical model was established by reviewing existing drone path optimization strategies.
(2) A new unmanned aerial vehicle path optimization strategy was created using an improved tuna optimization algorithm, and it was improved through sine and Levy's flight.
(3) The simulation results were used to confirm the efficiency and effectiveness of the proposed technology.
(4) The performance indicators of the proposed algorithm were compared with seven other algorithms.

2. Relate Works

Unmanned aerial vehicles (UAVs) are becoming more and more common in science and technology because of their many benefits, which include their affordability, simplicity of use in dangerous environments, tiny size, and excellent mobility. Some initial applications in fields including crisis management, target tracking, and military surveillance have already resulted from these benefits [20]. In order to enable UAVs to avoid obstacles safely while completing flight missions, a major focus of UAV path planning research is to investigate practical path planning models and solution algorithms that take into account variables like flight distance, altitude, turning angles, and obstacle threats [21]. Route planning models and solution algorithms are the two main areas of research for UAV route planning. Building optimization models based on parameters such as path length, flight height, angles of flight, flight duration, and obstacle threats is the primary focus of model research endeavors. Additionally, the computing efficiency of these models has been investigated [22].

Intelligent optimization and numerical optimization are the two primary kinds of algorithms utilized today to solve UAV route planning models [23]. Accurate iterative approaches like the previous one include conventional methods such as visibility graph (VG), artificial potential field (APF), and rapidly exploring random trees (RRTs). In simple scenarios, these techniques may be able to address UAV route planning problems because of their great optimization efficiency [24]. However, when dealing with three-dimensional path planning problems under varying obstacle threat conditions, these models' high number of path nodes and significant computational loads make it difficult to produce flight route solutions that satisfy all constraint requirements [25]. This is a result of restrictions in both the UAV and the outside environment. For example, Kelner et al. [26] used the rapidly exploring random tree algorithm to solve UAV path planning problems. They treated the flight path's beginning as the search tree's root node and chose nearby nodes that met both constraint limits and had the lowest possible cost to include in the search tree. They repeated this process until they were able to determine a UAV flight route. However, the algorithm has difficulty locating exits among obstacle clusters when there are a lot of closely spaced barriers and very few escape pathways. As a consequence, the global best path cannot be found. For static UAV path planning, Cao et al. [27] used the artificial potential field approach, which is simple to use and has a quick convergence time but produces flight route options that are of lower quality. Furthermore, the produced routes have a tendency to approach concave-shaped barriers in scenarios with several complicated obstacles, which might result in the failure of the planned flight paths. Under the assumption that the locations of three-dimensional obstacles were known, Blasi et al. [28] used the visibility graph approach with flight path length as a performance parameter to create path planning routes for rotorcraft UAVs. While this approach offers a solution, it is not appropriate for locations with a high number of obstacles, and the resulting flight path plans frequently result in low flying efficiency. All things considered, even if numerical optimization methods have worked well in some situations, they have drawbacks in complex or severely limited environments. Therefore, there is a constant need for improvements in intelligent optimization algorithms designed to overcome these obstacles and improve UAV path planning's flexibility and efficiency.

In the study of UAV route planning, swarm intelligent optimization—a metaheuristic random search algorithm—has been very important. Its high chance of obtaining a globally optimal solution stems from its directed and random character in the solution process, quick speed of optimization, and steady quality of solution search [29]. A spherical vector-based particle swarm optimization (PSO) technique, for instance, was presented by Phung et al. [30], which takes into account variables including UAV route length, altitude, angles, and obstacle threats. This approach efficiently shrinks the search space, improving the quality of the solutions sought and flight safety by utilizing the magnitude, elevation, azimuth components, and turn and climb angles of linked vectors. By dividing the UAV flight time evenly using a timestamp segmentation technique and adding social hierarchy concepts to

the fundamental pigeon-inspired optimization (PIO) algorithm, Wang et al. [31] created an enhanced PIO algorithm that can successfully solve multi-UAV cooperative path planning problems. These algorithms have a straightforward structure, speedy optimization, and the ability to produce excellent path planning schemes that adhere to UAV flying parameters. Though these benefits make UAV route planning issues easier to solve, they struggle to provide sufficient practicable flight path plans because they are sluggish in developing workable solutions that avoid obstacles and frequently become stuck in local searches.

To sum up, the majority of the research performed on UAV path planning models at the moment is focused on building flight route planning models that are constrained by ground impediments and performance metrics like flight duration and distance. Planning models that incorporate multiple performance indicators and handle intricate obstacle settings are comparatively less often designed. This is especially true for models that incorporate ground and airspace limits as well as multi-performance indicator optimization. As a result, new or enhanced intelligent optimization techniques are set to replace conventional approaches as the go-to solutions for UAV route planning issues. In order to do this, this work introduces a unique approach to UAV route planning that makes use of the Levy flight-guided tuna swarm optimization (SLTSO) algorithm in conjunction with a sinusoidal strategy.

3. Mathematical Model for Path Planning of UAVs

Unmanned aerial vehicle (UAV) route planning model design takes into account factors including the flight path, the airspace that may be flown through, and flight characteristics like altitude, angle, threat level, and flight distance, which are all used as performance indicators [32].

3.1. The Problem with Trajectory Planning

An essential component of a UAV's capacity to finish a mission is trajectory planning, which has a direct impact on the effectiveness and efficiency of task execution [33]. Trajectory planning is the process of selecting, among a large number of viable paths, an ideal or nearly ideal path that avoids obstructions and satisfies predetermined requirements, as shown in Figure 1.

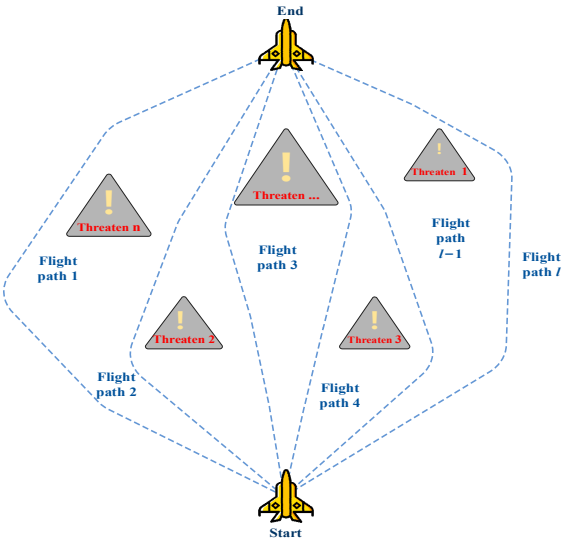


Figure 1. Obstacle avoidance path for UAVs.

3.2. Path Encoding

The path of a UAV can be regarded as a series of line segments connected by coordinate points. Therefore, the path is encoded using a real number encoding method, which is represented as follows:

$$(x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n, z_1, z_2, \ldots, z_n) \tag{1}$$

where $x_i (i = 1, 2, \cdots, n)$ represents the x-axis coordinate of the i-th trajectory point. The parameters $y_i (i = 1, 2, \cdots, n)$ and $z_i (i = 1, 2, \cdots, n)$ represent the y-axis and z-axis coordinates of the i-th trajectory point. The parameter n is the number of path nodes.

3.3. Constraint Condition

The path planning model of unmanned aerial vehicles includes terrain constraints and corner constraints. The terrain constraints are represented as follows:

$$h(x_i, y_i) + z_{\min} < z_i < z_{\max}, i = 1, 2, \ldots, n \tag{2}$$

where $h(x_i, y_i)$ represents the terrain height corresponding to the two-dimensional terrain coordinates. Z_{min} and Z_{max} represent the minimum and maximum safe flight altitude of the UAV, respectively. Other constraints, such as minimum track segment length, maximum track length, and maximum turning angle, can be expressed as follows [34]:

$$\theta_i \leqslant \theta_{\max}, i = 1, 2, \ldots, n \tag{3}$$

$$L_i \geqslant L_{\min}, i = 1, 2, \ldots, n \tag{4}$$

$$\sum_{i=1}^{n} L_i \leqslant L_{\max}, i = 1, 2, \ldots, n \tag{5}$$

where the parameter θ_i represents the turning angle of the i-th trajectory point. $\theta_{\max}$ is the maximum turning angle. L_i is the path length for the i-th segment of the trajectory. L_{min} is the minimum track segment length, and L_{max} is the maximum track length.

3.4. Optimization Objectives

In this article, fuel consumption and risk are considered as two optimization objectives. Among them, these two goals may conflict with each other in some cases. For example, the shorter the path, the greater the threat.

3.4.1. Fuel Consumption

Generally speaking, a shorter path length means shorter flight time and also requires less fuel consumption. In this article, fuel consumption refers to the amount of fuel consumed by a UAV during its flight from the starting point to the target point during mission execution. Let f_{fuxl} be the fuel consumption cost. If the speed of the UAV remains constant during flight, fuel consumption is directly proportional to the length of the flight path. Therefore, fuel consumption f_{fuxl} can be calculated as follows:

$$f_{\text{fuxl}} = \alpha^* \sum_{i=1}^{n} L_i = \alpha^* \sum_{i=1}^{n} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2} \tag{6}$$

In Equation (6), the parameter α is the fuel coefficient and L_i is the path length of the i-th segment of the trajectory.

3.4.2. Threats

This study mainly considers detecting threats and high threats. Assuming there are m detection threats, the projection center coordinates of the k-th (k = 1, 2, …, m)) detection

threat are C_k. The threat radius is R_k. When the UAV flies from path node W_i to node W_{i+1}, the straight-line distance from C_k to W_i W_{i+1} is d_k, as shown in Figure 2.

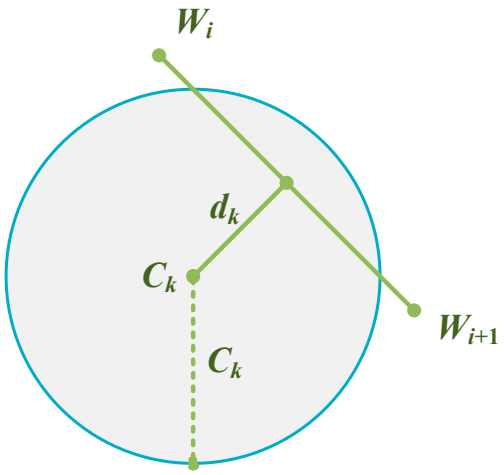


Figure 2. Threat model of the UAV flight.

At this point, the detection threat value of the UAV is inversely proportional to its distance d_k. Therefore, the detection threat f_{detect} can be calculated as follows:

$$f_{\text{detect}} = \sum_{i=1}^{n} \sum_{k=1}^{m} T_{i,k} \tag{7}$$

$$T_{i,k} = \begin{cases} 0, d_k \geqslant R_k \\ R_k/d_k, 0 < d_k < R_k \\ \infty, \ d_k = 0 \end{cases} \tag{8}$$

In Equation (8), $T_{i,k}$ represents the detection threat value from the k-th interference center received by the UAV on the i-th trajectory.

The higher the flying altitude of a UAV, the greater the probability of being detected by an enemy. Therefore, flight altitude is also one of the threats that this article needs to consider. In a 3D map, UAVs should strive to maintain an average safe altitude while preventing collisions with the terrain environment. Therefore, the high threat f_{altitude} can be calculated as follows:

$$f_{\text{altitude}} = \sum_{i=1}^{n} H_i \tag{9}$$

$$H_i = \begin{cases} \left| z_i - \frac{z_{\max} + z_{\min}}{2} \right| & , h(x_i, y_i) + z_{\min} < z_i < z_{\max} \\ \infty & , \text{others} \end{cases} \tag{10}$$

In Equation (10), the parameter H_i represents the altitude threat of the UAV in the i-th trajectory. Therefore, the threat of UAVs f_{threat} can be expressed as

$$f_{\text{threat}} = f_{\text{dctect}} + f_{\text{altitude}} \tag{11}$$

3.5. Path Planning Model for UAVs

According to Sections 3.2 and 3.3, the three-dimensional path planning model for unmanned aerial vehicles can be represented as:

$$\mathbf{J} = [\min(f_{\text{fuel}}), \min(f_{\text{threat}})])^{\text{T}} \tag{12}$$

$$\text{s.t}\begin{cases} h(x_i, y_i) + z_{\min} < z_i < z_{\max}, i = 1, 2, \ldots, n \\ \theta_i \leqslant \theta_{\max}, i = 1, 2, \ldots n \\ L_i \geqslant L_{\min}, i = 1, 2, \ldots n \\ \sum_{i=1}^{n} L_i \geqslant L_{\max}, i = 1, 2, \ldots n \end{cases} \tag{13}$$

In Equation (12), **J** is the objective vector, and the problem is a constrained minimization bi-objective optimization problem.

4. Tuna Swarm Optimization Algorithm (TSO)

In 2021, researcher Lei Xie presented the population-based global optimization metaheuristic method known as "tuna swarm optimization" (TSO). The TSO algorithm imitates two foraging strategies found in schools of tuna. The first, known as spiral foraging, explains how the fish swim in a spiral pattern to guide prey toward shallow waters where they can feed; the second, known as parabolic foraging, involves a fish swimming in a parabolic configuration, trailing the tail of the fish in front of it to encircle the school of tuna. TSO achieves global optimization by using these two foraging strategies [35]. The Tuna swarm optimization method is a unique population-based metaheuristic algorithm that mimics the helical and parabolic feeding patterns of tuna schools. It was first presented by researchers such as Xie et al. in 2021 [36]. This approach is distinguished by its robust exploratory capabilities and fewer configurable parameters. Currently, the tuna algorithm finds use in many different fields. Fan et al. [37] used an enhanced tuna swarm method to tackle the economic operation problem in hydropower facilities. A new and enhanced tuna optimization particle filter technique was suggested by Fu et al. [38], which effectively resolves the power system harmonic estimation problem. An improved tuna swarm approach was used by Gou et al. [39] to address Jensen model parameter estimation issues. Nanda et al. [40] used an ELM model based on an enhanced tuna swarm method to maximize the prediction of solar power production. An adaptive black hole–tuna swarm technique was introduced by Sheeja et al. [41] for multi-objective energy conservation optimization in wireless sensor networks.

Standard Tuna Swarm Optimization Algorithm (TSO)

The underlying principle of the tuna swarm optimization algorithm (TSO) is that every member of the population is figuratively represented as a tuna and that each tuna uses its own foraging strategy to search for the best solutions while being impacted by the other tunas' foraging activities [42]. Every tuna modifies its location according to its own fitness and the fitness of other tunas in each algorithm iteration, improving its environmental adaptation and pursuing the global optimum [43].

(1) Population Initialization

In the standard tuna swarm algorithm, the initialization of the population follows a similar approach to most population-based metaheuristics, where an initial population is randomly generated within the search space. The mathematical expression for this process can be denoted as Equation (14):

$$p_i^n = \text{rand} \cdot (ub - lb) + lb, i = 1, 2, 3, \ldots, NP \tag{14}$$

where *rand* is a random number in the interval [0, 1], *ub* is the upper limit of the search space, *lb* is the lower limit of the search space, p_i^n represents the initialization value of the tuna population, and *NP* is the number of populations.

(2) Tuna school spiral feeding

The first foraging strategy for tuna schools is spiral foraging. When tunas feed, they form a spiral shape that drives them to shallow water areas where they are more vulnerable to attack. The mathematical model of the spiral foraging strategy is as follows:

$$X_i^{t+1} = \begin{cases} \begin{cases} \alpha_1 \cdot \left(X_{\text{best}}^t + \beta \cdot \left|X_{\text{best}}^t - X_i^t\right|\right) + \alpha_2 \cdot X_i^t, i = 1 \\ \alpha_1 \cdot \left(X_{\text{best}}^t + \beta \cdot \left|X_{\text{best}}^t - X_i^t\right|\right) + \alpha_2 \cdot X_{i-1}^t, i = 2, 3, \ldots, NP \end{cases} & \text{rand} < \frac{t}{t_{\max}} \text{ (15-}a\text{)} \\ \begin{cases} \alpha_1 \cdot \left(X_{\text{rand}}^t + \beta \cdot \left|X_{\text{rand}}^t - X_i^t\right|\right) + \alpha_2 \cdot X_i^t, i = 1 \\ \alpha_1 \cdot \left(X_{\text{rand}}^t + \beta \cdot \left|X_{\text{rand}}^t - X_i^t\right|\right) + \alpha_2 \cdot X_{i-1}^t, i = 2, 3, \ldots, NP \end{cases} & \text{rand} < \frac{t}{t_{\max}} \text{ (15-}b\text{)} \end{cases} \tag{15}$$

$$\alpha_1 = a + (1 - a) \cdot \frac{t}{t_{\max}} \tag{16}$$

$$\alpha_2 = (1 - a) - (1 - a) \cdot \frac{t}{t_{\max}} \tag{17}$$

$$\beta = e^{bl} \cdot \cos(2\pi b) \tag{18}$$

$$l = e^{3\cos(((t_{\max}+1/t)-1)\pi)} \tag{19}$$

In the above equation, X_i^{t+1} represents the individual position in the $t + 1$ iteration. X_{best}^t represents the optimal position of the current individual. X_{rand}^t represents the randomly generated individual positions in the search space. α_1 and α_2 are weight coefficients that control the tendency of an individual to move towards the optimal individual and the previous individual, respectively. The parameter a is a constant and represents the degree coefficient of the tuna following the optimal individual and the previous individual in the initial stage. The parameter t represents the current number of iterations. The parameter $t_{\max}$ represents the maximum number of iterations. The parameters β and l are intermediate variables. The parameters b and *rand* are represented as random numbers uniformly distributed between [0, 1]. The parameter NP represents the tuna population size. Tunas enhance their search capability in the surrounding space through spiral foraging behavior, as per Equation (15-a), where, when all tunas spiral forage around food sources, they exhibit a strong capacity to explore the search space effectively to locate optimal solutions. However, when the best-performing individual fails to find food, blindly following the optimal individual may lead to decreased search efficiency for the entire group. Consequently, to augment the global search ability of the tuna swarm algorithm, a random individual position is introduced as a reference point for spiral searching, according to Equation (15-b). This helps each tuna to probe a broader area in its quest. As the number of algorithm iterations increases, the tuna swarm algorithm gradually shifts the reference point for spiral foraging from a randomly chosen individual's location to that of the optimal individual, thereby enhancing local search capability and search precision [44].

(3) Tuna parabolic foraging

The second tuna feeding strategy is parabolic feeding, where each tuna swims along with the previous one, forming a parabolic shape to surround the tuna. In addition, tunas also search for food around themselves. The mathematical model of the parabolic foraging strategy is as follows:

$$X_i^{t+1} = \begin{cases} X_{\text{best}}^t + rand \cdot \left(X_{\text{rand}}^t - X_i^t\right) + TF \cdot p^2 \cdot \left(X_{\text{rand}}^t - X_i^t\right) & rand < 0.5 \\ TF \cdot p^2 \cdot X_i^t & rand \geq 0.5 \end{cases} \tag{20}$$

$$p = \left(1 - \frac{t}{t_{\max}}\right)^{\left(\frac{t}{t_{\max}}\right)} \tag{21}$$

where TF in Equation (20) is a random number of -1 or 1, used to control the direction of population development, and the parameter p is an adjustment factor used to control the magnitude of population development [45].

Tunas cooperate through the above two foraging strategies, continuously updating their individual positions until the termination conditions are met. In each iteration, each tuna updates its individual historical optimal position and global historical optimal position based on its own position and fitness value and finally returns the optimal individual position and its corresponding fitness value.

5. Improved TSO Algorithm Based on the Sine Strategy and Levy Flight (SLTSO)

5.1. Elite Opposition-Based Learning Mechanism

It has been found via scientific study that applying a reverse learning process can lead to a 50% improvement in the likelihood of finding the ideal answer. The fundamental idea behind this concept is to create a new population by choosing the best individuals from both possible solutions and their inverses [46]. Building on this basis, the elite opposition-based learning (EOBL) approach creates a reverse population that is exclusive to the elite members of the original population. Because elite people are more likely to possess relevant knowledge, EOBL increases population quality overall and search efficiency more than standard opposition-based learning [47]. The EOBL method enhances the likelihood of locating the global optimum by concentrating on elite people, which successfully keeps the population from being stuck in local optima during the optimization process. By focusing computing resources on high-performing regions, this selective strategy optimizes the trade-off between exploration and exploitation while preserving the possibility to explore beyond the current best solutions.

The definition of elite reverse solving assumes that $X_{i,j}^E = \left(X_{i,1}^E, X_{i,2}^E, \cdots, X_{i,d}^E\right)$, $(i = 1, 2, \cdots, N), (j = 1, 2, \cdots, d)$, is an elite individual in d-dimensional space, and its inverse solution is defined as $\overline{X_{i,j}^E} = \left(X_{i,1}^E, X_{i,2}^E, \cdots, X_{i,d}^E\right)$, such that

$$\overline{X_{i,j}^E} = c \cdot \left(\mathrm{lb}_j + \mathrm{ub}_j\right) - X_{i,j}^E \tag{22}$$

where $X_{i,j}^E \in \left[lb_j, ub_j\right], c \in [0, 1]$, the parameter c is a random number between 0 and 1, and $ub_j = \max(X_{i,j})$ and $lb_j = \min(X_{i,j})$ are the upper and lower bounds of the dynamic search boundary. Dynamic changes in search boundaries can save previous optimization experience and improve search efficiency. Individuals beyond the search boundary are reset using Formula (23):

$$\overline{X_{i,j}^E} = \mathrm{rand}\left(lb_j + ub_j\right) \tag{23}$$

When using the elite reverse learning strategy to calculate the reverse solution for the initial population and the elite individuals in the later iteration stage, the reverse solution for the top $N/2$ individuals in the population fitness ranking is calculated according to Equation (22). The reverse population and the original population are sorted based on individual fitness, and the optimal N elite individuals are selected to form the next generation of optimization population. This strategy can effectively avoid blind searching by algorithms and reduced solution efficiency, laying the foundation for improving solution accuracy and accelerating convergence speed.

5.2. Introducing the Levy Flight Strategy and the Greedy Strategy to Update Positions

Levy flight is used to update the position of a tuna in the TSO algorithm. Levy flight is a random flight with alternating search ranges, which can enhance the algorithm's global search ability, making it possible for the algorithm to jump out of local optima [48,49]. The mathematical expression for position update is shown in Equation (24):

$$xl_i(t) = x_i(t) + l \times C1 \otimes levy \tag{24}$$

In Formula (24), $xl_i(t)$ represents the position of the Levy flight update, and the parameter *1* is the step weight. Levy represents the step size that follows the Levy distribution, and the step size calculation Formula is shown in Equation (25):

$$\text{levy} \;=\; u/|v|^{\frac{1}{\beta_l}} \tag{25}$$

where the parameters u and v follow a normal distribution, $u \sim N\left(0, \sigma_u^2\right), v \sim N\left(0, \sigma_v^2\right)$. The definitions of σ_u^2 and σ_v^2 are shown in Equations (26) and (27).

$$\sigma_u \;=\; \left[\frac{\Gamma(1+\delta)\sin(\delta\pi/2)}{\Gamma((1+\delta)/2)2^{(\delta-1)/2}\delta}\right]^{1/\delta} \tag{26}$$

$$\sigma_v \;=\; 1 \tag{27}$$

where the parameter δ takes a value of 1.5.

After updating the positions of tunas using Levy flight, because of the random flight characteristics of Levy flight, the updated position of a may not necessarily be a better position. Therefore, a greedy strategy is adopted to choose whether to retain the original position or find a new position. The selection Formula is shown in Equation (28).

$$x_i(t) \;=\; \begin{cases} x_i(t), \text{ fitness } (x_i(t)) \leq \text{ fitness } (xl_i(t)) \\ xl_i(t), \text{ fitness } (x_i(t)) > \text{ fitness } (xl_i(t)) \end{cases} \tag{28}$$

where, in Formula (28), *fitness*($x(t)$) and *fitness*($xl(t)$) respectively represent the original tuna position calculation solution of the algorithm and the tuna position calculation solution after the Levy flight update. To distinguish the updated position for a tuna, $x_i(t)$ is assigned to $xl_i(t)$ in Equation (28), thus obtaining Equation (29):

$$Xh_i(t) \;=\; x_i(t) \tag{29}$$

5.3. Introducing the Golden Sine Strategy to Update Individual Positions

This work also offers uses the golden sine approach to update a tuna's position in response to the sluggish convergence speed of TSO and its tendency to easily fall into local optima. The golden sine algorithm is the source of the golden sine strategy [50]. First, because of the strong global search ability of this method, which is based on the angle relationship between the unit circle and the sine function, the sine function may traverse all points within the search range of the unit circle [51]. Second, the algorithm's convergence speed is accelerated by employing the golden ratio coefficient to condense the solution space and provide a possibly ideal search region [52].

The position update Formula used in the golden sine strategy to update the position of a tuna in the previous iteration is shown in Equation (30):

$$xg_i(t) \;=\; x_i(t)\left|\sin\left(r_{g1}\right)\right| + r_{g2}\sin\left(r_{g2}\right)\left|c_1 T_{\text{pos}} \;- c_2 x_i(t)\right| \tag{30}$$

where $xg_i(t)$ is the tuna position updated by the golden sine strategy, r_{g1} is a random number within [0, 2π], r_{g2} is a random number within [0, π], and c_1 and c_2 are the golden ratio coefficients. The expressions for c_1 and c_2 are shown in Equations (31) and (32):

$$c_1 \;=\; a\tau + b(1-\tau) \tag{31}$$

$$c_2 \;=\; a(1-\tau) + b\tau \tag{32}$$

where $\tau = (\sqrt{5} - 1)/2 \sim 0.6183$. The value of the parameter a is $-\pi$, and the value of the parameter b is π.

Then, $x_i(t)$ in the tuna position update Formula is replaced with $xg_i(t)$, thus reducing the optimization space, accelerating the tuna search speed, and improving the algorithm convergence speed. The new tuna position update Formula is shown in Equation (33):

$$x_i(t+1) \;=\; T_{pos} + C1Z\cos(2\pi R_4) \times \left(T_{pos} - xg_i(t)\right) \tag{33}$$

where R_4 is a random number within the range of $[-1, 1]$. To distinguish the updated position for a tuna, $x_i(t)$ is assigned to $Xp_i(t)$ in Equation (33), and the Formula becomes Equation (34).

$$Xp_i(t) = x_i(t) \tag{34}$$

Combining the optimized tuna and individual search agent mechanisms, the final algorithm position update Formula is obtained by combining Equations (29) and (34) to obtain Equation (35). R_5 is a random number within the range of [0, 1], and β is the adjustment parameter.

$$\begin{array}{rcl} x_i(t) & = & Xh_i(t), R_5 < \beta \text{ (35-a)}, \\ x_i(t) & = & Xp_i(t), R_5 \geq \beta \text{ (35-b)} \end{array} \tag{35}$$

5.4. The Workflow for the Proposed SLTSO Algorithm

Based on the above methods, the standard tuna swarm algorithm was improved. The steps for applying the improved SLTSO algorithm to UAV trajectory planning are as follows:

Step 1: Initialize the basic parameters of the improved tuna swarm algorithm. The population size is NP, the maximum number of iterations is t_{max}, the position vector dimension is D, the upper and lower limits of the search space are ub and lb, and the probability parameter is z.

Step 2: Use the elite reverse learning mechanism to initialize the position of the tuna school.

Step 3: Calculate the fitness value J of a tuna according to Equation (12), and determine whether the constraint conditions are met using Equation (13). Compare the fitness values that meet the constraint conditions, update the optimal value of the current individual tuna (i.e., the fitness value of the tuna), and save it.

Step 4: When the algorithm enters the iterative process, if the random number rand generated in [0, 1] is less than z, update the position of the tuna according to Equation (14) and proceed to step 8.

Step 5: If the random number *rand* generated within [0, 1] satisfies the condition rand $\geq$ z and also meets the condition $p < p'$, and if another random number rand generated within [0, 1] is less than t/t_{max}, update the tuna's position according to Equation (15-a) and proceed to Step 8.

Step 6: If the condition $p < p''$ is met and the random number *rand* is generated in [0, 1], and rand $\geq t/t_{max}$, update the position of the tuna according to Equation (28) and proceed to step 8.

Step 7: If the condition $p \geq p'$ is met, update the position of the tuna according to Equation (20) and proceed to step 8.

Step 8: Calculate the fitness value of the updated tuna position according to Formula (30), compare the fitness value that meets the constraints of the UAV with the optimal value of the previously saved individual tuna, and update the compared optimal value to the current individual tuna's optimal value and save it.

Step 9: Determine if the maximum number of iterations has been reached. If it has, stop the iteration and output the optimal fitness value of the tuna individual. Otherwise, return to step 4.

The workflow of the SLTSO algorithm is shown in Figure 3.

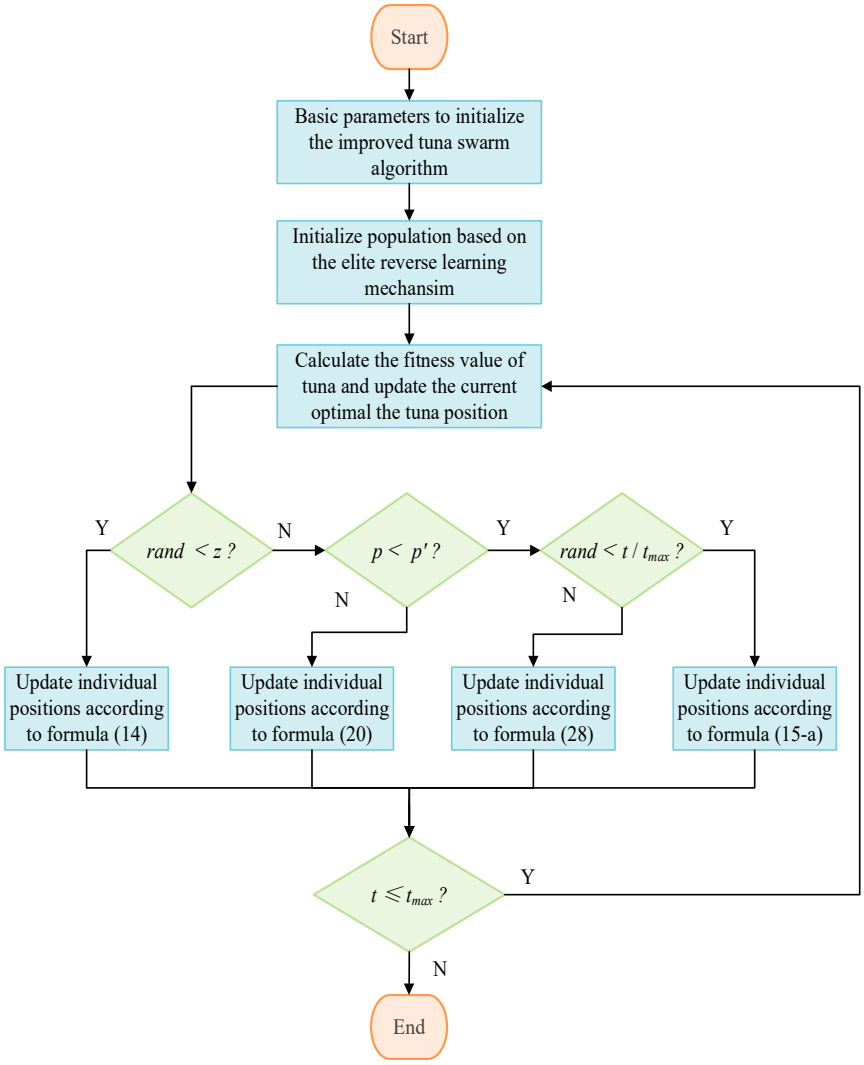


Figure 3. The workflow diagram of the SLTSO algorithm.

5.5. Time Complexity Analysis of the SLTSO Algorithm

Time complexity is a crucial metric for evaluating the performance of an algorithm. It assumes the size of the tuna swarm population is N, the dimensionality of the problem is D, and the maximum number of iterations is T, with the time required to evaluate the objective fitness function denoted as $f(D)$. According to the implementation steps of the SLTSO (Sine–Levy truncated tuna swarm optimization) algorithm, the initialization phase has a time complexity of $O(N \cdot (f(D) + D))$, meaning that the time required scales linearly with the population size and the combined effort of evaluating the fitness function and initializing the dimensions.

Since the tuna optimization strategy in the SLTSO algorithm modifies only the update method for discoverers without adding extra computational steps, the position update stage has a time complexity of $O(N \cdot D)$, indicating that the position update operations for each individual take place independently and proportionally to the number of dimensions. The phase of applying position update strategies also incurs a time complexity of $O(N \cdot D)$. The introduction of the golden sine strategy and Levy flight adds another layer of complexity, which amounts to $O(N \cdot f(D) + D)$, reflecting the additional overhead for applying these advanced strategies to each individual in the population. Taking all these stages into account, the overall time complexity of the SLTSO algorithm is $O(N \cdot D \cdot T)$, which is equivalent to the basic TSO algorithm. Hence, SLTSO does not introduce any

additional computational burden in terms of its time complexity compared to the original TSO algorithm.

6. Simulation Experiments and Result Analysis

By contrasting it with five sample dynamic multi-objective algorithms, the suggested SLTSO algorithm's performance is validated and analyzed. The test functions, comparison algorithms, parameter configurations, and analysis of the experimental data are described in detail below. The 12th Generation Intel(R) Core(TM) i9-12900K CPU, running at 3.20 GHz, 64 GB of RAM, and MATLAB software version R2022b were installed on the laptop used in this research. In order to validate the benefits of the refined approach put forward in this work, simulations were run using a population size of 30 and a maximum iteration count of 1000. The standard particle swarm optimization (PSO) [53], dung beetle optimization (DBO) [54], slime mold algorithm (SMA) [55], Harris hawk optimization (HHO) [56], subtraction–averaging-based optimization (SABO) [57], sand cat swarm optimization (SCSO) [58], basic tuna swarm optimization (TSO), and the improved TSO algorithm incorporating both the sine strategy and Levy flight were the algorithms used to benchmark the performance of the SLTSO algorithm.

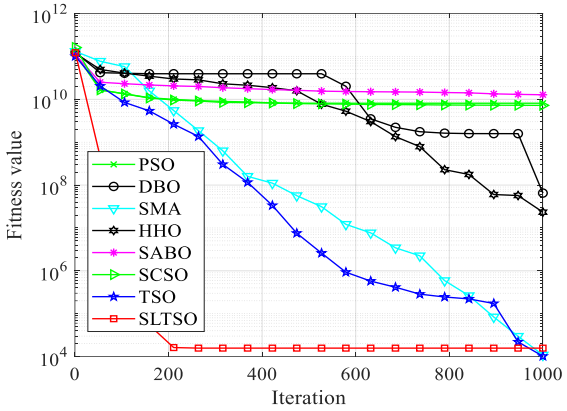
6.1. Comparison of Test Function Results

For numerical simulations, the tests used 30 benchmark test functions from the CEC2018 suite. The test functions for CEC2018 are a set of functions intended to evaluate the effectiveness of intelligent optimization algorithms. There are 30 functions altogether, which include multimodal, unimodal, hybrid, and composite function types. These functions are designed to assess the effectiveness of optimization algorithms in a variety of scenarios, taking into account various dimensions and degrees of difficulty. On all 30 base test functions, the proposed SLTSO algorithm was evaluated against the PSO, DBO, SMA, HHO, SABO, SCSO, and TSO algorithms separately. Each function was run independently for 50 trials. The objective of this comparison was to examine the optimization performance and stability benefits of the SLTSO algorithm over alternative swarm intelligence algorithms. Figure 4 displays the convergence curves of the iterative procedures for various test functions. It is noteworthy that Function F2 encountered difficulties during execution, which were caused by issues with the CEC 2018 test function itself. As a result, no results were provided for this specific function.

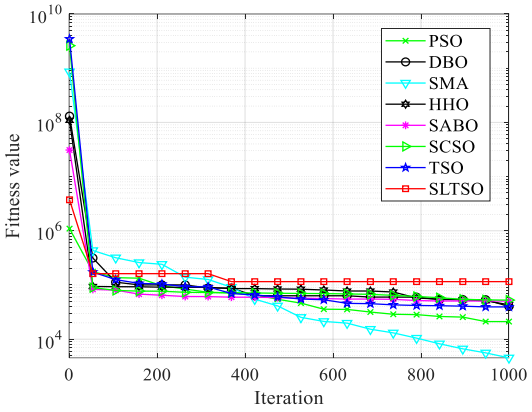
In this study, convergence curves based on iteration counts and fitness values for the test functions are presented to give a more comprehensible comparison of the convergence accuracy and speed of the methods. The optimization performance of the proposed SLTSO method, when compared to the PSO, DBO, SMA, HHO, SABO, SCSO, and TSO algorithms, is typically similar, as shown in Figure 4 for Functions F1 through F5. However, certain specific situations may not perform as well as optimally. But, in terms of Functions F19 through F30, the SLTSO algorithm performs several orders of magnitude better than the other algorithms. Its standard deviation is also typically lower than the other algorithms' standard deviations, suggesting that the SLTSO algorithm keeps a higher diversity in its initial population, which greatly improves its stability. With direct search to optimal values and 100% optimization efficiency achieved for Functions F12 and F14, the SLTSO algorithm demonstrates optimization performance for Functions F12 to F18 that is comparable to that of PSO, DBO, SMA, HHO, SABO, SCSO, and TSO. This is a significant improvement over the unmodified PSO, DBO, SMA, HHO, SABO, and SCSO algorithms. This implies that the SLTSO's implementation of the Levy flight optimization strategy and sine approach promotes population diversity, expands the search space, and strengthens the searcher's exploration technique through individual position updates, enabling the algorithm to escape local optima successfully.

The convergence curves of the test functions show that, in general, the SLTSO method has a faster rate of convergence than the PSO, DBO, SMA, HHO, SABO, and SCSO algorithms. Functions F20 through F30 are high-dimensional multi-modal functions that

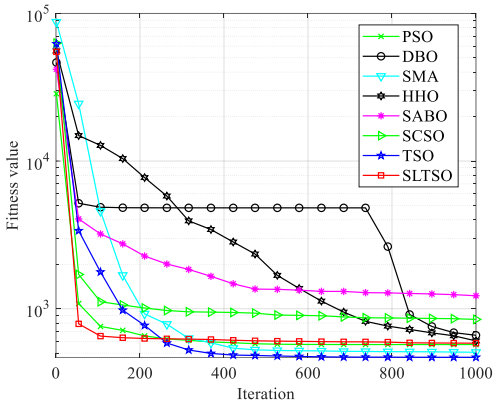
are used to evaluate the global exploration capabilities of algorithms. Because of their global optima, they are difficult to optimize. It is clear from their individual convergence curves that the SLTSO algorithm outperforms the PSO, DBO, SMA, HHO, SABO, SCSO, and TSO algorithms in terms of convergence speed and precision. This suggests that while optimizing high-dimensional functions, the other algorithms commonly become stuck in local optima, significantly slowing down their convergence rates. SLTSO, on the other hand, initially uses an elite reverse learning process to guarantee initial population diversity. The discoverer's position update technique is then improved by the sine optimization strategy, which keeps the algorithm from being trapped in local optima in the early iterations. Finally, by applying perturbations to the top performers, the Levy flight strategy allows the algorithm to exit local optima quickly and finally converge to the global optimum value.



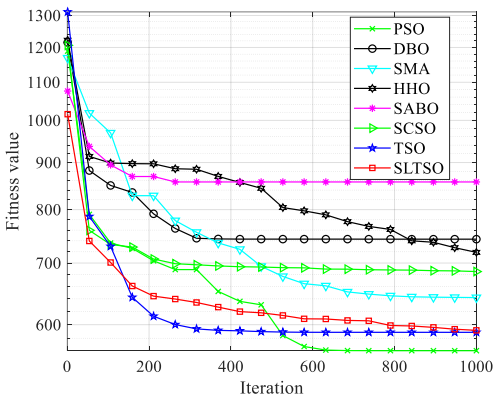
(**F1**)



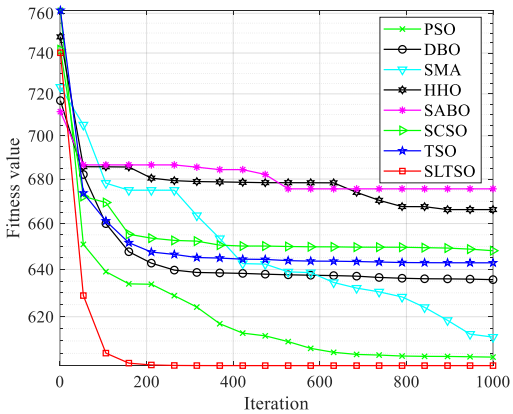
(**F3**)



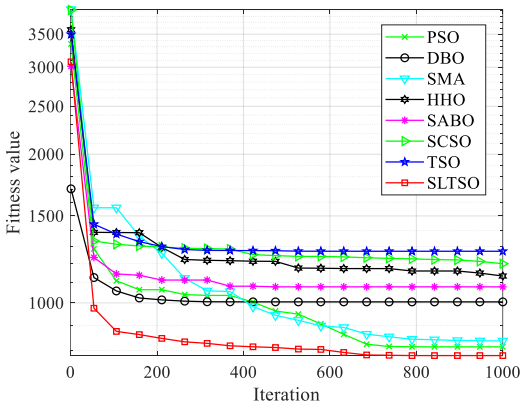
(**F4**)



(**F5**)

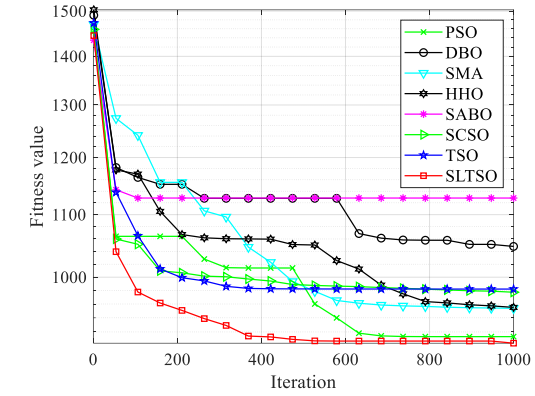


(**F6**)

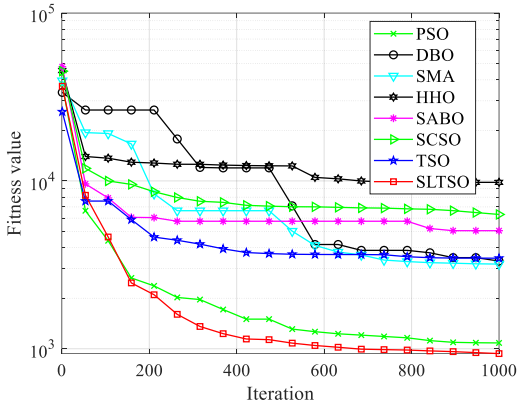


(**F7**)

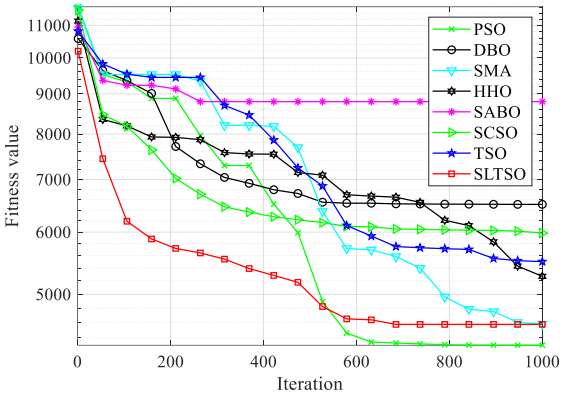
Figure 4. *Cont.*



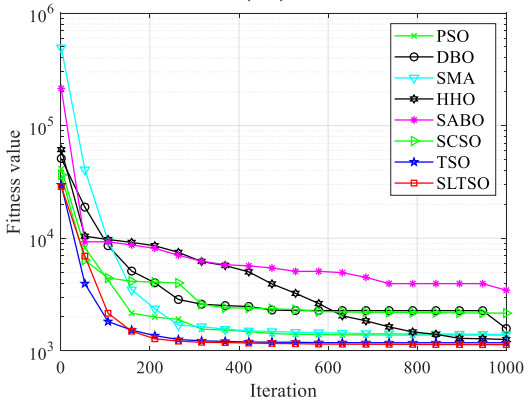
(F8)



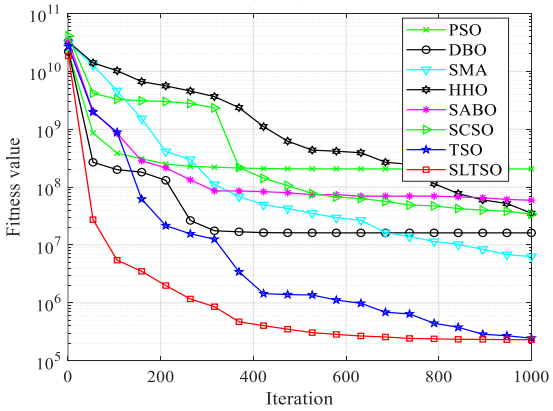
(F9)



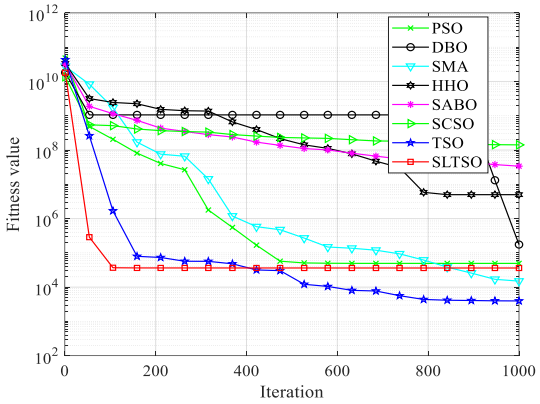
(F10)



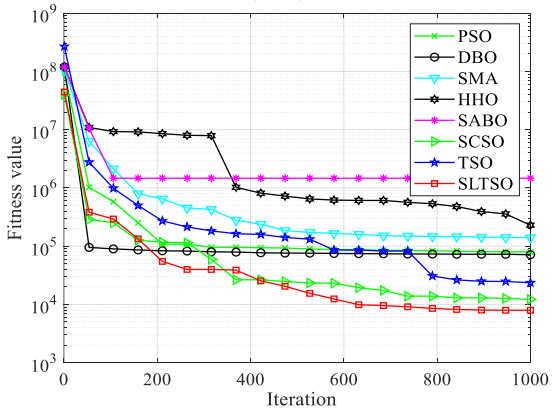
(F11)



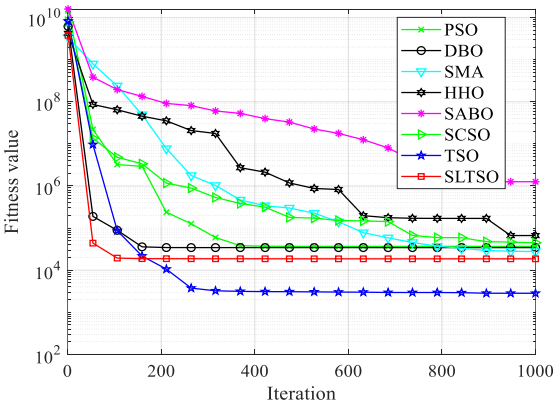
(F12)



(F13)

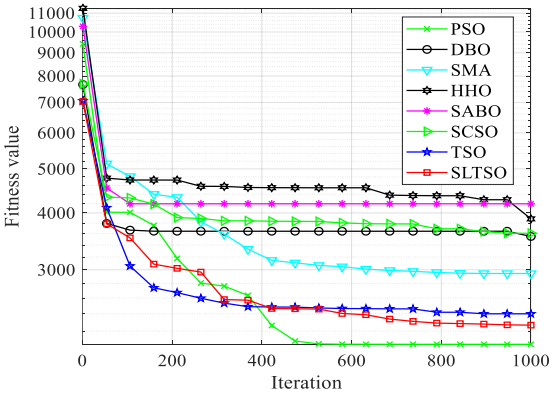


(F14)

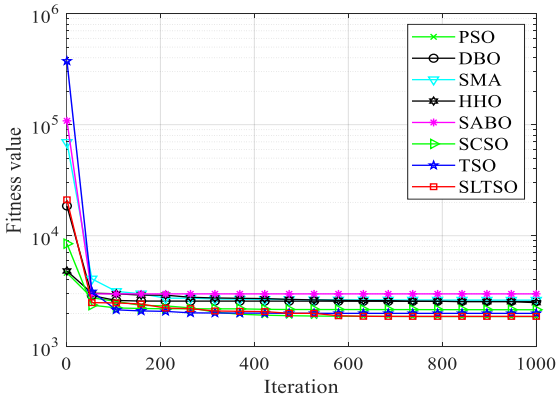


(F15)

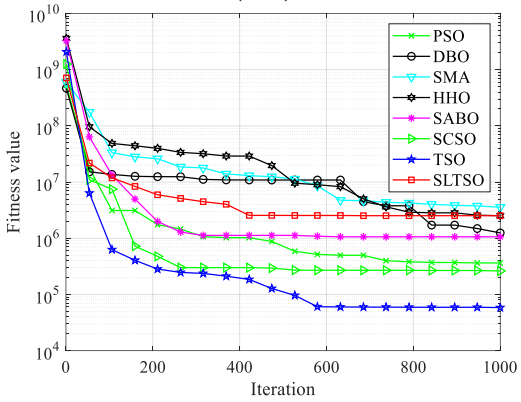
Figure 4. *Cont.*



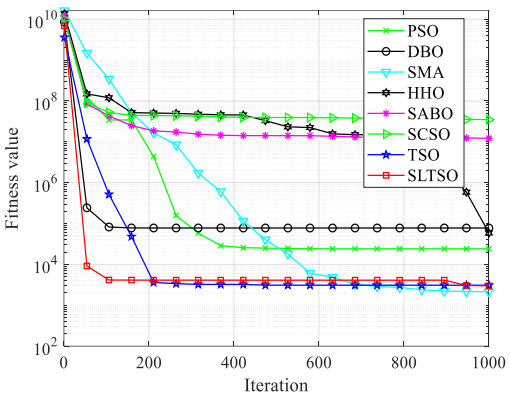
(F16)



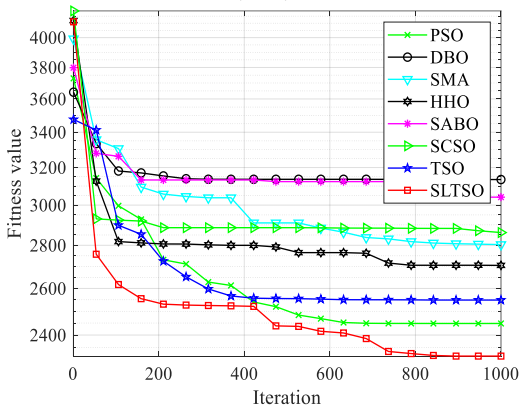
(F17)



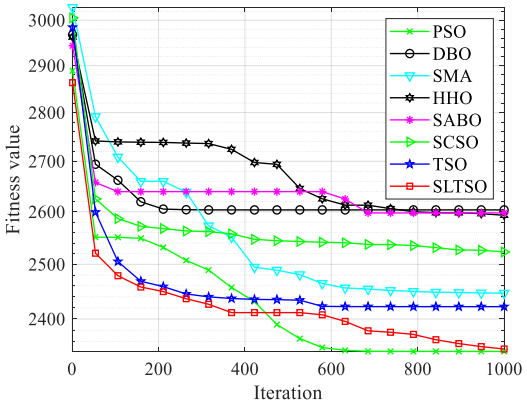
(F18)



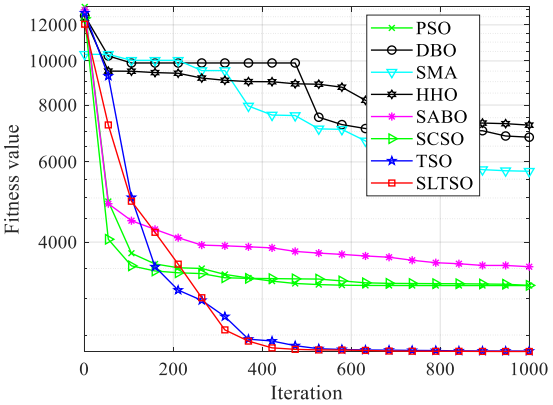
(F19)



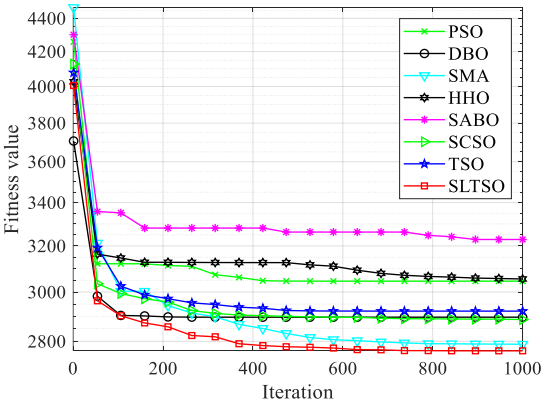
(F20)



(F21)



(F22)



(F23)

Figure 4. *Cont.*

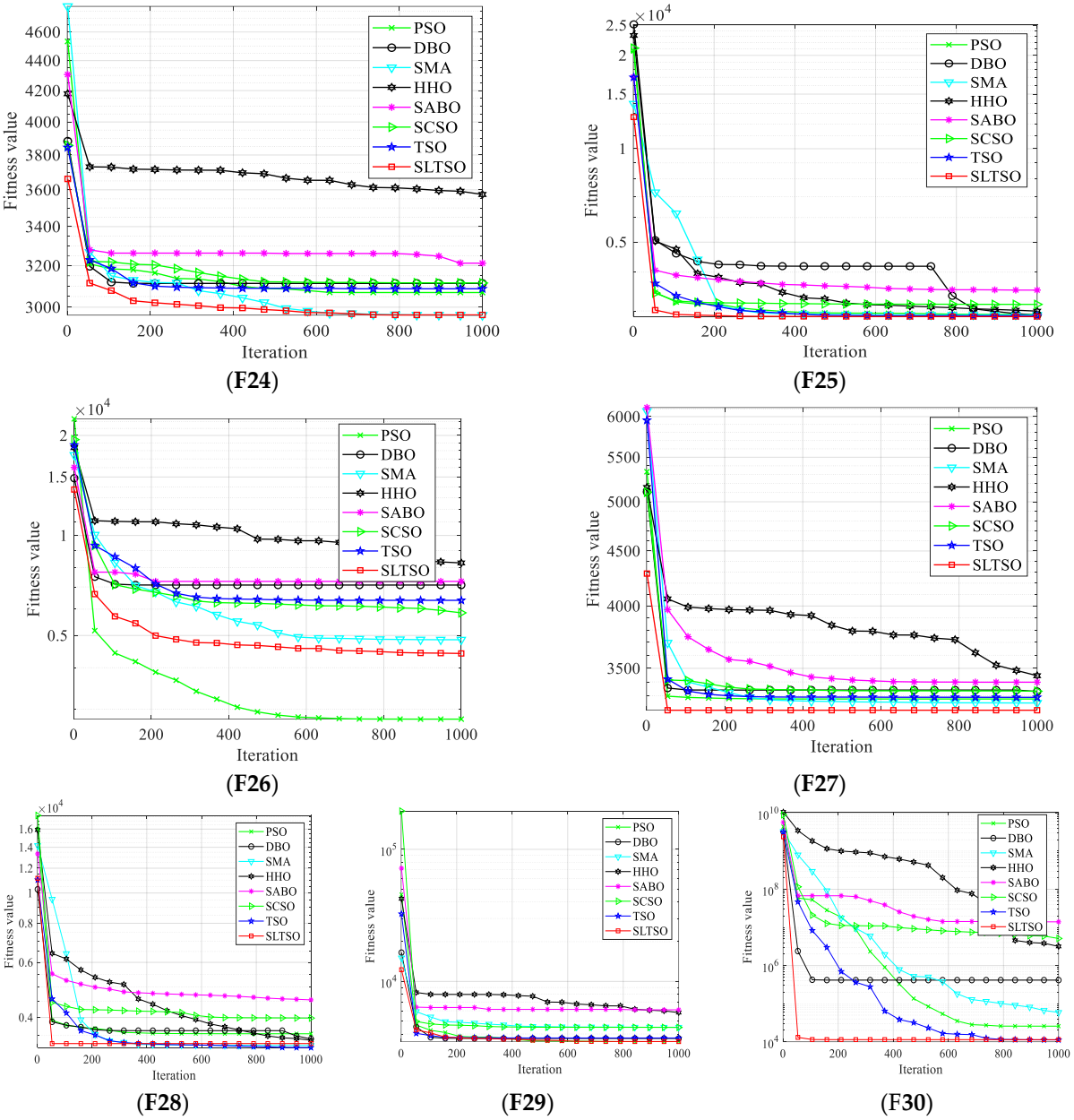


Figure 4. Iterative convergence curves of the test functions.

In conclusion, the SLTSO algorithm shows significant stability and optimization performance improvement for the set of 30 benchmark test functions. It performs especially well for Functions F20 to F30, outperforming the seven other algorithms by a significant margin. Moreover, for high-dimensional multi-modal functions, the SLTSO algorithm has a notable capacity to escape local optima quickly and converge to the global optimum.

For every strategy in this work, the population size and number of iterations are fixed at 30 and 1000, respectively, to improve test result dependability and reduce the impact of randomness inherent in heuristic algorithms. The ideal value findings from the 30 different runs were used to compute statistical variables such as the mean, standard deviation (Std), and min. The algorithm's average ability to optimize the target function is represented by the mean. For every target function, the min displays the best optimization result out of the thirty executions. The stability of the algorithm's ability to optimize a given target function is measured by the Std. Tables 1–3 present the testing outcomes of the eight optimization algorithms under consideration across 10, 30, and 50 dimensions for the entire set of 30 benchmark functions, providing a comprehensive overview of their comparative

performances. These statistical data substantiate the feasibility and effectiveness of the SLTSO algorithm.

Table 1. Comparison of 8 algorithms of 10 dimension.

Function		PSO	DBO	SMA	HHO	SABO	SCSO	TSO	SLTSO
	Min	1.16×10^{2}	1.02×10^{2}	9.92×10^{2}	3.00×10^{5}	8.98×10^{6}	2.59×10^{4}	1.01×10^{2}	1.05×10^{2}
F1	Std	1.28×10^{7}	1.75×10^{7}	3.92×10^{3}	5.84×10^{5}	6.34×10^{8}	5.55×10^{8}	2.57×10^{3}	2.03×10^{3}
	Mean	2.34×10^{6}	5.55×10^{6}	7.43×10^{3}	1.11×10^{6}	4.84×10^{8}	2.79×10^{8}	1.59×10^{3}	2.05×10^{3}
	Min	3.00×10^{2}	3.00×10^{2}	3.00×10^{2}	3.06×10^{2}	7.68×10^{2}	3.17×10^{2}	3.00×10^{2}	3.00×10^{2}
F3	Std	7.73×10^{-1}	1.55×10^{3}	6.75×10^{-1}	3.12×10^{2}	1.61×10^{3}	2.49×10^{3}	1.06×10^{0}	7.02×10^{0}
	Mean	3.00×10^{2}	1.37×10^{3}	3.00×10^{2}	6.16×10^{2}	3.09×10^{3}	2.87×10^{3}	3.00×10^{2}	3.02×10^{2}
	Min	4.01×10^{2}	4.01×10^{2}	4.00×10^{2}	4.00×10^{2}	4.07×10^{2}	4.03×10^{2}	4.00×10^{2}	4.00×10^{2}
F4	Std	4.21×10^{1}	3.28×10^{1}	3.21×10^{1}	3.54×10^{1}	2.22×10^{1}	3.89×10^{1}	1.30×10^{1}	2.01×10^{1}
	Mean	4.23×10^{2}	4.23×10^{2}	4.22×10^{2}	4.27×10^{2}	4.43×10^{2}	4.41×10^{2}	4.06×10^{2}	4.11×10^{2}
	Min	5.05×10^{2}	5.17×10^{2}	5.09×10^{2}	5.24×10^{2}	5.34×10^{2}	5.12×10^{2}	5.12×10^{2}	5.02×10^{2}
F5	Std	8.53×100	1.37×10^{1}	6.52×10^{0}	1.72×10^{1}	1.03×10^{1}	1.55×10^{1}	1.24×10^{1}	2.37×10^{0}
	Mean	5.14×10^{2}	5.40×10^{2}	5.20×10^{2}	5.59×10^{2}	5.54×10^{2}	5.41×10^{2}	5.27×10^{2}	5.07×10^{2}
	Min	6.00×10^{2}	6.01×10^{2}	6.00×10^{2}	6.14×10^{2}	6.09×10^{2}	6.03×10^{2}	6.02×10^{2}	6.00×10^{2}
F6	Std	2.89×10^{-1}	8.23×10^{0}	6.09×10^{-1}	1.25×10^{1}	1.03×10^{1}	1.07×10^{1}	6.83×10^{0}	9.05×10^{-9}
	Mean	6.00×10^{2}	6.13×10^{2}	6.00×10^{2}	6.42×10^{2}	6.20×10^{2}	6.18×10^{2}	6.11×10^{2}	6.00×10^{2}
	Min	7.13×10^{2}	7.30×10^{2}	7.19×10^{2}	7.62×10^{2}	7.39×10^{2}	7.39×10^{2}	7.22×10^{2}	7.06×10^{2}
F7	Std	6.44×100	1.45×10^{1}	5.96×10^{0}	1.83×10^{1}	1.39×10^{1}	1.68×10^{1}	1.57×10^{1}	3.46×10^{0}
	Mean	7.23×10^{2}	7.49×10^{2}	7.29×10^{2}	7.91×10^{2}	7.67×10^{2}	7.68×10^{2}	7.48×10^{2}	7.16×10^{2}
	Min	8.05×10^{2}	8.07×10^{2}	8.05×10^{2}	8.15×10^{2}	8.32×10^{2}	8.18×10^{2}	8.06×10^{2}	8.02×10^{2}
F8	Std	4.71×100	1.58×10^{1}	8.80×10^{0}	8.53×10^{0}	9.17×10^{0}	7.51×10^{0}	7.37×10^{0}	2.10×10^{0}
	Mean	8.13×10^{2}	8.37×10^{2}	8.20×10^{2}	8.30×10^{2}	8.49×10^{2}	8.31×10^{2}	8.22×10^{2}	8.07×10^{2}
	Min	9.00×10^{2}	9.02×10^{2}	9.00×10^{2}	9.82×10^{2}	9.18×10^{2}	9.15×10^{2}	9.11×10^{2}	9.00×10^{2}
F9	Std	1.16×10^{-1}	1.14×10^{2}	2.94×10^{0}	2.14×10^{2}	9.53×10^{1}	2.33×10^{2}	7.82×10^{1}	8.29×10^{-2}
	Mean	9.00×10^{2}	9.98×10^{2}	9.01×10^{2}	1.58×10^{3}	1.04×10^{3}	1.12×10^{3}	9.96×10^{2}	9.00×10^{2}
	Min	1.02×10^{3}	1.28×10^{3}	1.14×10^{3}	1.27×10^{3}	2.34×10^{3}	1.37×10^{3}	1.37×10^{3}	1.05×10^{3}
F10	Std	2.55×10^{2}	3.10×10^{2}	2.52×10^{2}	3.65×10^{2}	1.98×10^{2}	3.20×10^{2}	3.00×10^{2}	1.71×10^{2}
	Mean	1.58×10^{3}	1.98×10^{3}	1.59×10^{3}	2.03×10^{3}	2.81×10^{3}	2.08×10^{3}	1.96×10^{3}	1.49×10^{3}
	Min	1.10×10^{3}	1.11×10^{3}	1.11×10^{3}	1.11×10^{3}	1.14×10^{3}	1.11×10^{3}	1.11×10^{3}	1.10×10^{3}
F11	Std	2.97×10^{1}	1.22×10^{2}	1.04×10^{2}	5.38×10^{1}	1.37×10^{3}	4.90×10^{1}	2.38×10^{1}	3.01×10^{0}
	Mean	1.12×10^{3}	1.23×10^{3}	1.21×10^{3}	1.18×10^{3}	1.86×10^{3}	1.16×10^{3}	1.14×10^{3}	1.10×10^{3}
	Min	3.65×10^{3}	1.93×10^{3}	6.40×10^{3}	5.05×10^{4}	1.41×10^{5}	1.79×10^{4}	2.27×10^{3}	1.98×10^{3}
F12	Std	1.49×10^{6}	5.67×10^{6}	6.24×10^{5}	2.98×10^{6}	1.72×10^{6}	1.95×10^{6}	9.44×10^{3}	1.20×10^{4}
	Mean	2.96×10^{5}	3.41×10^{6}	4.68×10^{5}	2.46×10^{6}	2.01×10^{6}	1.78×10^{6}	9.76×10^{3}	1.35×10^{4}
	Min	1.38×10^{3}	1.51×10^{3}	1.55×10^{3}	2.45×10^{3}	3.56×10^{3}	2.47×10^{3}	1.58×10^{3}	1.36×10^{3}
F13	Std	9.29×10^{3}	1.19×10^{4}	1.24×10^{4}	8.47×10^{3}	6.16×10^{3}	1.05×10^{4}	6.04×10^{3}	1.32×10^{4}
	Mean	9.63×10^{3}	1.45×10^{4}	1.12×10^{4}	1.42×10^{4}	1.30×10^{4}	1.43×10^{4}	1.00×10^{4}	1.24×10^{4}
	Min	1.43×10^{3}	1.45×10^{3}	1.43×10^{3}	1.50×10^{3}	1.51×10^{3}	1.46×10^{3}	1.43×10^{3}	1.41×10^{3}
F14	Std	8.39×10^{2}	5.29×10^{2}	2.57×10^{3}	9.72×10^{2}	1.05×10^{3}	2.03×10^{3}	3.96×10^{1}	2.49×10^{2}
	Mean	1.76×10^{3}	1.97×10^{3}	2.16×10^{3}	2.09×10^{3}	2.21×10^{3}	3.48×10^{3}	1.50×10^{3}	1.56×10^{3}
	Min	1.51×10^{3}	1.63×10^{3}	1.53×10^{3}	1.78×10^{3}	2.75×10^{3}	1.58×10^{3}	1.52×10^{3}	1.51×10^{3}
F15	Std	2.35×10^{3}	2.45×10^{3}	3.62×10^{3}	3.00×10^{3}	5.27×10^{3}	1.80×10^{3}	3.86×10^{2}	7.64×10^{2}
	Mean	3.10×10^{3}	4.34×10^{3}	5.19×10^{3}	7.92×10^{3}	9.55×10^{3}	4.92×10^{3}	1.81×10^{3}	1.86×10^{3}
	Min	1.60×10^{3}	1.61×10^{3}	1.60×10^{3}	1.63×10^{3}	1.79×10^{3}	1.62×10^{3}	1.60×10^{3}	1.60×10^{3}
F16	Std	1.37×10^{2}	1.45×10^{2}	9.73×10^{1}	1.49×10^{2}	1.23×10^{2}	1.34×10^{2}	1.19×10^{2}	4.60×10^{1}
	Mean	1.76×10^{3}	1.82×10^{3}	1.70×10^{3}	1.93×10^{3}	2.06×10^{3}	1.82×10^{3}	1.79×10^{3}	1.63×10^{3}
	Min	1.71×10^{3}	1.73×10^{3}	1.72×10^{3}	1.75×10^{3}	1.76×10^{3}	1.73×10^{3}	1.72×10^{3}	1.70×10^{3}
F17	Std	3.24×10^{1}	4.04×10^{1}	3.63×10^{1}	5.87×10^{1}	7.79×10^{1}	2.98×10^{1}	2.32×10^{1}	2.49×10^{1}
	Mean	1.75×10^{3}	1.78×10^{3}	1.76×10^{3}	1.80×10^{3}	1.87×10^{3}	1.77×10^{3}	1.75×10^{3}	1.72×10^{3}
	Min	2.23×10^{3}	2.18×10^{3}	4.46×10^{3}	2.89×10^{3}	2.55×10^{3}	4.08×10^{3}	2.01×10^{3}	1.97×10^{3}
F18	Std	1.52×10^{4}	1.62×10^{4}	1.40×10^{4}	1.33×10^{4}	1.21×10^{4}	1.77×10^{4}	7.14×10^{3}	7.76×10^{3}
	Mean	2.27×10^{4}	1.91×10^{4}	2.57×10^{4}	1.89×10^{4}	1.88×10^{4}	2.41×10^{4}	9.01×10^{3}	8.51×10^{3}
	Min	1.96×10^{3}	1.97×10^{3}	1.93×10^{3}	1.94×10^{3}	3.05×10^{3}	1.95×10^{3}	1.94×10^{3}	1.91×10^{3}
F19	Std	6.92×10^{3}	2.06×10^{4}	1.05×10^{4}	1.87×10^{4}	5.33×10^{3}	6.89×10^{3}	7.06×10^{2}	9.75×10^{2}
	Mean	6.29×10^{3}	1.14×10^{4}	1.49×10^{4}	2.08×10^{4}	8.90×10^{3}	8.01×10^{3}	2.56×10^{3}	2.57×10^{3}
	Min	2.00×10^{3}	2.03×10^{3}	2.00×10^{3}	2.04×10^{3}	2.12×10^{3}	2.04×10^{3}	2.01×10^{3}	2.00×10^{3}
F20	Std	5.44×10^{1}	7.86×10^{1}	5.01×10^{1}	6.89×10^{1}	6.66×10^{1}	6.78×10^{1}	6.41×10^{1}	2.23×10^{1}
	Mean	2.05×10^{3}	2.12×10^{3}	2.05×10^{3}	2.18×10^{3}	2.23×10^{3}	2.14×10^{3}	2.10×10^{3}	2.01×10^{3}
	Min	2.20×10^{3}	2.20×10^{3}	2.10×10^{3}	2.20×10^{3}	2.33×10^{3}	2.20×10^{3}	2.20×10^{3}	2.20×10^{3}
F21	Std	4.09×10^{1}	5.52×10^{1}	5.58×10^{1}	7.04×10^{1}	1.52×10^{1}	5.48×10^{1}	4.02×10^{1}	5.17×10^{1}
	Mean	2.30×10^{3}	2.23×10^{3}	2.31×10^{3}	2.32×10^{3}	2.35×10^{3}	2.31×10^{3}	2.22×10^{3}	2.27×10^{3}

Table 1. *Cont.*

Function		PSO	DBO	SMA	HHO	SABO	SCSO	TSO	SLTSO
F22	Min	2.23×10^{3}	2.24×10^{3}	2.22×10^{3}	2.26×10^{3}	2.31×10^{3}	2.30×10^{3}	2.24×10^{3}	2.20×10^{3}
	Std	2.93×10^{1}	2.31×10^{1}	2.72×10^{2}	2.10×10^{2}	5.19×10^{1}	7.85×10^{1}	1.60×10^{1}	1.95×10^{1}
	Mean	2.31×10^{3}	2.31×10^{3}	2.41×10^{3}	2.35×10^{3}	2.34×10^{3}	2.34×10^{3}	2.30×10^{3}	2.30×10^{3}
F23	Min	2.61×10^{3}	2.62×10^{3}	2.61×10^{3}	2.62×10^{3}	2.63×10^{3}	2.61×10^{3}	2.61×10^{3}	2.61×10^{3}
	Std	1.22×10^{1}	1.78×10^{1}	7.85×10^{0}	2.99×10^{1}	2.01×10^{1}	1.52×10^{1}	1.42×10^{1}	2.89×10^{0}
	Mean	2.63×10^{3}	2.64×10^{3}	2.62×10^{3}	2.68×10^{3}	2.66×10^{3}	2.64×10^{3}	2.63×10^{3}	2.61×10^{3}
F24	Min	2.50×10^{3}	2.50×10^{3}	2.74×10^{3}	2.50×10^{3}	2.53×10^{3}	2.50×10^{3}	2.50×10^{3}	2.50×10^{3}
	Std	4.93×10^{1}	1.06×10^{2}	1.26×10^{1}	1.43×10^{2}	4.98×10^{1}	7.23×10^{1}	8.02×10^{1}	9.11×10^{1}
	Mean	2.76×10^{3}	2.72×10^{3}	2.76×10^{3}	2.78×10^{3}	2.78×10^{3}	2.75×10^{3}	2.73×10^{3}	2.70×10^{3}
F25	Min	2.90×10^{3}	2.90×10^{3}	2.90×10^{3}	2.90×10^{3}	2.94×10^{3}	2.91×10^{3}	2.90×10^{3}	2.90×10^{3}
	Std	3.18×10^{1}	2.48×10^{1}	3.78×10^{1}	2.17×10^{1}	2.25×10^{1}	3.08×10^{1}	2.42×10^{1}	2.33×10^{1}
	Mean	2.93×10^{3}	2.94×10^{3}	2.94×10^{3}	2.94×10^{3}	2.97×10^{3}	2.94×10^{3}	2.93×10^{3}	2.92×10^{3}
F26	Min	2.60×10^{3}	2.82×10^{3}	2.90×10^{3}	2.61×10^{3}	3.11×10^{3}	2.86×10^{3}	2.60×10^{3}	2.60×10^{3}
	Std	3.36×10^{2}	2.31×10^{2}	4.42×10^{2}	6.56×10^{2}	4.02×10^{2}	2.28×10^{2}	1.43×10^{2}	1.29×10^{2}
	Mean	3.08×10^{3}	3.17×10^{3}	3.19×10^{3}	3.56×10^{3}	3.47×10^{3}	3.17×10^{3}	2.96×10^{3}	2.92×10^{3}
F27	Min	3.09×10^{3}	3.09×10^{3}	3.09×10^{3}	3.11×10^{3}	3.10×10^{3}	3.09×10^{3}	3.09×10^{3}	3.07×10^{3}
	Std	2.52×10^{1}	1.02×10^{1}	1.91×10^{0}	4.97×10^{1}	1.84×10^{1}	2.84×10^{1}	2.55×10^{1}	2.26×10^{0}
	Mean	3.12×10^{3}	3.10×10^{3}	3.09×10^{3}	3.17×10^{3}	3.11×10^{3}	3.11×10^{3}	3.11×10^{3}	3.07×10^{3}
F28	Min	3.10×10^{3}	3.17×10^{3}	3.17×10^{3}	3.10×10^{3}	3.23×10^{3}	3.17×10^{3}	3.10×10^{3}	3.10×10^{3}
	Std	1.22×10^{2}	1.27×10^{2}	1.30×10^{2}	1.12×10^{2}	1.38×10^{2}	1.18×10^{2}	1.61×10^{2}	4.61×10^{1}
	Mean	3.39×10^{3}	3.35×10^{3}	3.41×10^{3}	3.38×10^{3}	3.50×10^{3}	3.32×10^{3}	3.40×10^{3}	3.26×10^{3}
F29	Min	3.14×10^{3}	3.16×10^{3}	3.13×10^{3}	3.21×10^{3}	3.20×10^{3}	3.17×10^{3}	3.16×10^{3}	3.15×10^{3}
	Std	4.47×10^{1}	8.95×10^{1}	5.93×10^{1}	8.34×10^{1}	9.03×10^{1}	8.41×10^{1}	6.59×10^{1}	1.27×10^{1}
	Mean	3.20×10^{3}	3.30×10^{3}	3.22×10^{3}	3.36×10^{3}	3.34×10^{3}	3.29×10^{3}	3.25×10^{3}	3.17×10^{3}
F30	Min	6.30×10^{3}	6.66×10^{3}	6.71×10^{3}	1.68×10^{4}	5.59×10^{4}	5.04×10^{3}	4.67×10^{3}	3.23×10^{3}
	Std	5.61×10^{5}	1.44×10^{6}	6.02×10^{5}	2.77×10^{6}	4.55×10^{6}	1.31×10^{6}	2.12×10^{6}	1.25×10^{3}
	Mean	4.25×10^{5}	1.21×10^{6}	3.66×10^{5}	2.07×10^{6}	2.52×10^{6}	1.11×10^{6}	7.61×10^{5}	3.71×10^{3}

Table 2. Comparison of 8 algorithms of 30 dimension.

Function		PSO	DBO	SMA	HHO	SABO	SCSO	TSO	SLTSO
F1	Min	2.18×10^{3}	5.32×10^{7}	4.58×10^{4}	8.83×10^{7}	3.71×10^{9}	8.02×10^{8}	1.11×10^{6}	1.06×10^{2}
	Std	3.09×10^{9}	1.77×10^{8}	4.53×10^{4}	2.43×10^{8}	5.66×10^{9}	3.97×10^{9}	1.49×10^{7}	9.81×10^{3}
	Mean	2.00×10^{9}	2.59×10^{8}	1.05×10^{5}	4.30×10^{8}	1.34×10^{10}	9.08×10^{9}	1.07×10^{7}	7.75×10^{3}
F3	Min	3.99×10^{4}	6.29×10^{4}	1.39×10^{4}	3.80×10^{4}	4.59×10^{4}	3.58×10^{4}	2.48×10^{4}	7.14×10^{4}
	Std	2.50×10^{4}	9.19×10^{3}	1.55×10^{4}	8.27×10^{3}	1.02×10^{4}	1.10×10^{4}	1.51×10^{4}	3.48×10^{4}
	Mean	7.43×10^{4}	8.57×10^{4}	3.52×10^{4}	5.76×10^{4}	6.57×10^{4}	6.07×10^{4}	6.08×10^{4}	1.43×10^{5}
F4	Min	4.92×10^{2}	5.34×10^{2}	4.76×10^{2}	5.96×10^{2}	9.13×10^{2}	5.31×10^{2}	4.86×10^{2}	4.18×10^{2}
	Std	3.86×10^{2}	1.63×10^{2}	1.50×10^{1}	9.82×10^{1}	1.10×10^{3}	4.74×10^{2}	2.51×10^{1}	3.41×10^{1}
	Mean	7.68×10^{2}	6.68×10^{2}	5.02×10^{2}	7.29×10^{2}	2.21×10^{3}	9.90×10^{2}	5.31×10^{2}	4.73×10^{2}
F5	Min	5.40×10^{2}	6.68×10^{2}	5.61×10^{2}	7.24×10^{2}	7.64×10^{2}	6.84×10^{2}	6.00×10^{2}	5.60×10^{2}
	Std	2.33×10^{1}	5.76×10^{1}	4.59×10^{1}	2.64×10^{1}	3.60×10^{1}	4.98×10^{1}	3.02×10^{1}	1.75×10^{1}
	Mean	5.83×10^{2}	7.59×10^{2}	6.37×10^{2}	7.70×10^{2}	8.27×10^{2}	7.67×10^{2}	6.65×10^{2}	5.94×10^{2}
F6	Min	6.02×10^{2}	6.18×10^{2}	6.07×10^{2}	6.58×10^{2}	6.37×10^{2}	6.40×10^{2}	6.25×10^{2}	6.00×10^{2}
	Std	6.16×10^{0}	1.09×10^{1}	1.02×10^{1}	5.11×10^{0}	1.47×10^{1}	1.08×10^{1}	7.91×10^{0}	1.19×10^{-2}
	Mean	6.08×10^{2}	6.46×10^{2}	6.18×10^{2}	6.69×10^{2}	6.67×10^{2}	6.62×10^{2}	6.44×10^{2}	6.00×10^{2}
F7	Min	7.93×10^{2}	8.54×10^{2}	8.32×10^{2}	1.18×10^{3}	1.04×10^{3}	9.62×10^{2}	9.61×10^{2}	8.03×10^{2}
	Std	4.86×10^{1}	9.67×10^{1}	3.99×10^{1}	6.13×10^{1}	5.97×10^{1}	1.07×10^{2}	7.92×10^{1}	1.98×10^{1}
	Mean	8.44×10^{2}	1.02×10^{3}	8.93×10^{2}	1.30×10^{3}	1.13×10^{3}	1.16×10^{3}	1.08×10^{3}	8.36×10^{2}
F8	Min	8.44×10^{2}	9.21×10^{2}	8.73×10^{2}	9.26×10^{2}	1.04×10^{3}	9.56×10^{2}	8.67×10^{2}	8.56×10^{2}
	Std	2.69×10^{1}	5.11×10^{1}	2.86×10^{1}	3.00×10^{1}	2.60×10^{1}	2.88×10^{1}	3.19×10^{1}	1.69×10^{1}
	Mean	8.90×10^{2}	1.02×10^{3}	9.19×10^{2}	9.92×10^{2}	1.08×10^{3}	1.01×10^{3}	9.30×10^{2}	8.83×10^{2}
F9	Min	9.24×10^{2}	3.52×10^{3}	2.16×10^{3}	6.48×10^{3}	3.84×10^{3}	3.65×10^{3}	2.46×10^{3}	9.01×10^{2}
	Std	7.47×10^{2}	1.86×10^{3}	1.65×10^{3}	1.38×10^{3}	1.74×10^{3}	1.58×10^{3}	1.36×10^{3}	3.57×10^{2}
	Mean	1.55×10^{3}	6.49×10^{3}	4.75×10^{3}	8.78×10^{3}	6.71×10^{3}	6.47×10^{3}	4.58×10^{3}	1.08×10^{3}
F10	Min	2.71×10^{3}	4.41×10^{3}	3.98×10^{3}	5.31×10^{3}	7.90×10^{3}	4.80×10^{3}	4.12×10^{3}	3.89×10^{3}
	Std	7.57×10^{2}	1.14×10^{3}	5.90×10^{2}	7.52×10^{2}	3.81×10^{2}	6.72×10^{2}	1.14×10^{3}	4.80×10^{2}
	Mean	4.58×10^{3}	6.84×10^{3}	4.94×10^{3}	6.39×10^{3}	8.77×10^{3}	6.10×10^{3}	6.09×10^{3}	5.12×10^{3}

Table 2. *Cont.*

	Min	1.16×10^{3}	1.29×10^{3}	1.18×10^{3}	1.28×10^{3}	2.31×10^{3}	1.52×10^{3}	1.18×10^{3}	1.15×10^{3}
F11	Std	1.17×10^{2}	1.10×10^{3}	7.79×10^{1}	2.41×10^{2}	2.16×10^{3}	1.46×10^{3}	4.32×10^{1}	2.73×10^{1}
	Mean	1.31×10^{3}	2.10×10^{3}	1.29×10^{3}	1.59×10^{3}	5.60×10^{3}	3.25×10^{3}	1.26×10^{3}	1.18×10^{3}
	Min	1.85×10^{5}	1.28×10^{6}	3.26×10^{5}	5.54×10^{6}	1.09×10^{8}	2.43×10^{7}	4.06×10^{5}	4.83×10^{4}
F12	Std	1.46×10^{8}	1.40×10^{8}	3.58×10^{6}	7.73×10^{7}	7.16×10^{8}	7.65×10^{8}	2.85×10^{6}	1.81×10^{6}
	Mean	7.63×10^{7}	9.69×10^{7}	4.34×10^{6}	6.83×10^{7}	9.74×10^{8}	4.87×10^{8}	3.16×10^{6}	1.79×10^{6}
	Min	7.28×10^{3}	1.10×10^{4}	1.46×10^{4}	3.30×10^{5}	2.55×10^{6}	2.46×10^{4}	4.30×10^{3}	1.36×10^{3}
F13	Std	2.68×10^{8}	1.73×10^{7}	5.33×10^{4}	7.42×10^{5}	2.92×10^{8}	8.98×10^{7}	1.29×10^{4}	1.73×10^{4}
	Mean	7.66×10^{7}	6.67×10^{6}	6.64×10^{4}	1.04×10^{6}	1.50×10^{8}	3.83×10^{7}	1.93×10^{4}	1.86×10^{4}
	Min	5.00×10^{3}	1.32×10^{4}	1.43×10^{4}	2.16×10^{4}	4.68×10^{4}	1.57×10^{4}	3.38×10^{3}	3.82×10^{3}
F14	Std	9.64×10^{4}	4.69×10^{5}	2.88×10^{5}	1.07×10^{6}	8.95×10^{5}	8.53×10^{5}	4.70×10^{4}	6.49×10^{4}
	Mean	1.04×10^{5}	3.84×10^{5}	2.84×10^{5}	1.15×10^{6}	1.17×10^{6}	6.35×10^{5}	3.83×10^{4}	6.54×10^{4}
	Min	2.20×10^{3}	9.46×10^{3}	3.45×10^{3}	3.16×10^{4}	1.45×10^{5}	2.43×10^{4}	1.80×10^{3}	1.66×10^{3}
F15	Std	2.49×10^{4}	2.35×10^{6}	1.62×10^{4}	1.08×10^{5}	5.09×10^{6}	2.58×10^{6}	1.16×10^{4}	1.12×10^{4}
	Mean	2.11×10^{4}	5.86×10^{5}	2.15×10^{4}	1.40×10^{5}	3.13×10^{6}	2.06×10^{6}	7.75×10^{3}	1.22×10^{4}
	Min	1.95×10^{3}	2.59×10^{3}	2.00×10^{3}	2.28×10^{3}	3.75×10^{3}	2.43×10^{3}	2.37×10^{3}	2.14×10^{3}
F16	Std	2.99×10^{2}	4.02×10^{2}	3.34×10^{2}	5.15×10^{2}	2.86×10^{2}	3.67×10^{2}	2.95×10^{2}	2.36×10^{2}
	Mean	2.60×10^{3}	3.25×10^{3}	2.67×10^{3}	3.71×10^{3}	4.17×10^{3}	3.30×10^{3}	2.85×10^{3}	2.60×10^{3}
	Min	1.79×10^{3}	1.92×10^{3}	1.91×10^{3}	1.81×10^{3}	2.44×10^{3}	1.89×10^{3}	1.97×10^{3}	1.79×10^{3}
F17	Std	1.67×10^{2}	3.43×10^{2}	2.64×10^{2}	3.16×10^{2}	2.42×10^{2}	2.48×10^{2}	2.11×10^{2}	1.78×10^{2}
	Mean	2.11×10^{3}	2.68×10^{3}	2.32×10^{3}	2.71×10^{3}	2.92×10^{3}	2.38×10^{3}	2.38×10^{3}	2.13×10^{3}
	Min	1.70×10^{5}	7.87×10^{4}	3.53×10^{5}	1.83×10^{5}	2.69×10^{5}	5.12×10^{4}	4.72×10^{4}	1.83×10^{5}
F18	Std	1.14×10^{6}	4.03×10^{6}	2.78×10^{6}	4.47×10^{6}	6.12×10^{6}	7.48×10^{6}	2.06×10^{5}	6.45×10^{5}
	Mean	1.31×10^{6}	3.11×10^{6}	2.99×10^{6}	4.09×10^{6}	4.56×10^{6}	4.71×10^{6}	2.93×10^{5}	1.06×10^{6}
	Min	1.98×10^{3}	2.93×10^{3}	2.14×10^{3}	1.18×10^{5}	2.75×10^{5}	1.55×10^{5}	2.05×10^{3}	1.99×10^{3}
F19	Std	1.10×10^{5}	6.97×10^{7}	1.99×10^{4}	1.37×10^{6}	4.73×10^{6}	1.59×10^{6}	9.15×10^{3}	1.44×10^{4}
	Mean	5.21×10^{4}	1.69×10^{7}	1.98×10^{4}	1.75×10^{6}	5.32×10^{6}	1.89×10^{6}	1.05×10^{4}	1.32×10^{4}
	Min	2.07×10^{3}	2.30×10^{3}	2.09×10^{3}	2.54×10^{3}	2.80×10^{3}	2.30×10^{3}	2.23×10^{3}	2.16×10^{3}
F20	Std	1.75×10^{2}	2.38×10^{2}	2.20×10^{2}	1.96×10^{2}	1.50×10^{2}	2.02×10^{2}	2.01×10^{2}	1.25×10^{2}
	Mean	2.42×10^{3}	2.71×10^{3}	2.54×10^{3}	2.84×10^{3}	3.08×10^{3}	2.69×10^{3}	2.58×10^{3}	2.37×10^{3}
	Min	2.35×10^{3}	2.42×10^{3}	2.38×10^{3}	2.47×10^{3}	2.54×10^{3}	2.46×10^{3}	2.39×10^{3}	2.37×10^{3}
F21	Std	3.43×10^{1}	5.40×10^{1}	2.56×10^{1}	5.86×10^{1}	3.77×10^{1}	4.58×10^{1}	3.06×10^{1}	1.20×10^{1}
	Mean	2.41×10^{3}	2.55×10^{3}	2.43×10^{3}	2.60×10^{3}	2.60×10^{3}	2.53×10^{3}	2.44×10^{3}	2.39×10^{3}
	Min	2.30×10^{3}	2.37×10^{3}	2.30×10^{3}	3.12×10^{3}	3.39×10^{3}	2.52×10^{3}	2.31×10^{3}	2.30×10^{3}
F22	Std	1.69×10^{3}	2.27×10^{3}	9.37×10^{2}	1.45×10^{3}	8.31×10^{2}	9.93×10^{2}	1.78×10^{3}	2.33×10^{3}
	Mean	4.76×10^{3}	4.97×10^{3}	6.16×10^{3}	7.37×10^{3}	4.29×10^{3}	3.71×10^{3}	3.01×10^{3}	4.77×10^{3}
	Min	2.73×10^{3}	2.84×10^{3}	2.71×10^{3}	3.01×10^{3}	2.99×10^{3}	2.85×10^{3}	2.79×10^{3}	2.72×10^{3}
F23	Std	1.01×10^{2}	7.20×10^{1}	2.66×10^{1}	1.30×10^{2}	9.77×10^{1}	5.56×10^{1}	6.34×10^{1}	2.39×10^{1}
	Mean	2.91×10^{3}	2.99×10^{3}	2.77×10^{3}	3.30×10^{3}	3.20×10^{3}	2.94×10^{3}	2.92×10^{3}	2.77×10^{3}
	Min	2.91×10^{3}	3.02×10^{3}	2.87×10^{3}	3.25×10^{3}	3.13×10^{3}	3.00×10^{3}	2.95×10^{3}	2.91×10^{3}
F24	Std	1.06×10^{2}	6.97×10^{1}	3.91×10^{1}	1.53×10^{2}	1.11×10^{2}	5.13×10^{1}	1.19×10^{2}	2.15×10^{1}
	Mean	3.09×10^{3}	3.21×10^{3}	2.96×10^{3}	3.50×10^{3}	3.30×10^{3}	3.08×10^{3}	3.15×10^{3}	2.95×10^{3}
	Min	2.88×10^{3}	2.90×10^{3}	2.88×10^{3}	2.95×10^{3}	3.20×10^{3}	3.00×10^{3}	2.89×10^{3}	2.88×10^{3}
F25	Std	6.07×10^{1}	5.33×10^{1}	1.54×10^{1}	3.37×10^{1}	1.47×10^{2}	2.00×10^{2}	2.13×10^{1}	1.12×10^{1}
	Mean	2.93×10^{3}	2.98×10^{3}	2.90×10^{3}	3.01×10^{3}	3.36×10^{3}	3.19×10^{3}	2.93×10^{3}	2.89×10^{3}
	Min	2.80×10^{3}	3.95×10^{3}	4.58×10^{3}	4.28×10^{3}	6.90×10^{3}	4.12×10^{3}	3.07×10^{3}	2.91×10^{3}
F26	Std	9.35×10^{2}	9.96×10^{2}	3.06×10^{2}	1.29×10^{3}	6.01×10^{2}	1.47×10^{3}	1.21×10^{3}	5.16×10^{2}
	Mean	4.77×10^{3}	7.04×10^{3}	5.01×10^{3}	8.12×10^{3}	8.38×10^{3}	6.69×10^{3}	6.36×10^{3}	4.69×10^{3}
	Min	3.21×10^{3}	3.22×10^{3}	3.21×10^{3}	3.23×10^{3}	3.30×10^{3}	3.27×10^{3}	3.23×10^{3}	3.20×10^{3}
F27	Std	7.48×10^{1}	1.17×10^{2}	1.37×10^{1}	2.26×10^{2}	1.12×10^{2}	8.28×10^{1}	7.43×10^{1}	1.93×10^{-4}
	Mean	3.29×10^{3}	3.40×10^{3}	3.23×10^{3}	3.56×10^{3}	3.49×10^{3}	3.40×10^{3}	3.34×10^{3}	3.20×10^{3}
	Min	3.23×10^{3}	3.23×10^{3}	3.21×10^{3}	3.36×10^{3}	3.53×10^{3}	3.36×10^{3}	3.21×10^{3}	3.26×10^{3}
F28	Std	1.19×10^{2}	4.86×10^{2}	5.37×10^{1}	6.91×10^{1}	3.50×10^{2}	3.80×10^{2}	2.78×10^{1}	1.01×10^{1}
	Mean	3.35×10^{3}	3.52×10^{3}	3.27×10^{3}	3.47×10^{3}	4.24×10^{3}	3.83×10^{3}	3.30×10^{3}	3.30×10^{3}
	Min	3.37×10^{3}	3.75×10^{3}	3.59×10^{3}	4.22×10^{3}	5.07×10^{3}	3.99×10^{3}	3.72×10^{3}	3.26×10^{3}
F29	Std	2.45×10^{2}	2.65×10^{2}	2.69×10^{2}	4.61×10^{2}	4.37×10^{2}	4.24×10^{2}	3.12×10^{2}	1.48×10^{2}
	Mean	3.79×10^{3}	4.47×10^{3}	4.01×10^{3}	4.94×10^{3}	5.71×10^{3}	4.73×10^{3}	4.29×10^{3}	3.51×10^{3}
	Min	8.84×10^{3}	3.24×10^{4}	2.45×10^{4}	7.63×10^{5}	2.99×10^{6}	2.09×10^{6}	9.31×10^{3}	3.23×10^{3}
F30	Std	1.58×10^{6}	6.06×10^{6}	7.86×10^{4}	1.21×10^{7}	2.72×10^{7}	1.60×10^{7}	2.19×10^{4}	5.10×10^{3}
	Mean	4.54×10^{5}	3.03×10^{6}	1.10×10^{5}	9.77×10^{6}	3.50×10^{7}	1.94×10^{7}	3.77×10^{4}	7.70×10^{3}

Table 3. Comparison of 8 algorithms of 50 dimension.

Function		PSO	DBO	SMA	HHO	SABO	SCSO	TSO	SLTSO
F1	Min	1.20×10^{9}	3.66×10^{8}	2.19×10^{6}	2.19×10^{9}	2.49×10^{10}	1.48×10^{10}	3.12×10^{8}	1.75×10^{2}
	Std	5.83×10^{9}	1.05×10^{10}	2.67×10^{6}	1.55×10^{9}	1.04×10^{10}	6.99×10^{9}	5.57×10^{8}	6.13×10^{5}
	Mean	7.50×10^{9}	6.57×10^{9}	7.26×10^{6}	5.36×10^{9}	4.32×10^{10}	2.77×10^{10}	1.02×10^{9}	1.25×10^{5}
F3	Min	1.26×10^{5}	1.78×10^{5}	1.11×10^{5}	1.41×10^{5}	1.56×10^{5}	9.28×10^{4}	1.51×10^{5}	2.45×10^{5}
	Std	5.42×10^{4}	6.59×10^{4}	5.67×10^{4}	1.87×10^{4}	1.50×10^{4}	2.26×10^{4}	3.25×10^{4}	4.52×10^{4}
	Mean	2.22×10^{5}	2.57×10^{5}	2.05×10^{5}	1.75×10^{5}	1.83×10^{5}	1.36×10^{5}	2.04×10^{5}	3.47×10^{5}
F4	Min	5.73×10^{2}	7.37×10^{2}	5.16×10^{2}	1.05×10^{3}	3.73×10^{3}	1.30×10^{3}	7.22×10^{2}	4.51×10^{2}
	Std	5.59×10^{2}	1.31×10^{3}	6.20×10^{1}	3.77×10^{2}	2.17×10^{3}	1.41×10^{3}	9.90×10^{1}	6.43×10^{1}
	Mean	1.14×10^{3}	1.57×10^{3}	6.17×10^{2}	1.69×10^{3}	7.56×10^{3}	4.02×10^{3}	8.55×10^{2}	5.55×10^{2}
F5	Min	6.51×10^{2}	7.48×10^{2}	7.30×10^{2}	8.80×10^{2}	1.02×10^{3}	8.67×10^{2}	7.63×10^{2}	6.96×10^{2}
	Std	4.99×10^{1}	9.11×10^{1}	5.50×10^{1}	2.95×10^{1}	4.90×10^{1}	3.88×10^{1}	3.73×10^{1}	2.56×10^{1}
	Mean	7.21×10^{2}	9.61×10^{2}	8.21×10^{2}	9.33×10^{2}	1.11×10^{3}	9.50×10^{2}	8.39×10^{2}	7.42×10^{2}
F6	Min	6.09×10^{2}	6.34×10^{2}	6.19×10^{2}	6.74×10^{2}	6.61×10^{2}	6.57×10^{2}	6.49×10^{2}	6.00×10^{2}
	Std	6.61×10^{0}	1.13×10^{1}	1.26×10^{1}	3.45×10^{0}	1.10×10^{1}	7.62×10^{0}	7.98×10^{0}	2.26×10^{0}
	Mean	6.19×10^{2}	6.62×10^{2}	6.49×10^{2}	6.79×10^{2}	6.87×10^{2}	6.75×10^{2}	6.63×10^{2}	6.02×10^{2}
F7	Min	9.60×10^{2}	1.14×10^{3}	1.05×10^{3}	1.65×10^{3}	1.51×10^{3}	1.52×10^{3}	1.26×10^{3}	9.87×10^{2}
	Std	9.28×10^{1}	1.54×10^{2}	8.94×10^{1}	8.43×10^{1}	1.04×10^{2}	1.03×10^{2}	1.35×10^{2}	3.87×10^{1}
	Mean	1.09×10^{3}	1.40×10^{3}	1.17×10^{3}	1.89×10^{3}	1.66×10^{3}	1.72×10^{3}	1.62×10^{3}	1.06×10^{3}
F8	Min	9.46×10^{2}	1.06×10^{3}	9.58×10^{2}	1.15×10^{3}	1.32×10^{3}	1.15×10^{3}	1.01×10^{3}	9.80×10^{2}
	Std	4.78×10^{1}	1.16×10^{2}	5.33×10^{1}	3.68×10^{1}	5.70×10^{1}	4.93×10^{1}	4.53×10^{1}	2.79×10^{1}
	Mean	1.02×10^{3}	1.28×10^{3}	1.09×10^{3}	1.23×10^{3}	1.45×10^{3}	1.30×10^{3}	1.12×10^{3}	1.05×10^{3}
F9	Min	1.67×10^{3}	1.10×10^{4}	7.93×10^{3}	2.63×10^{4}	1.82×10^{4}	1.40×10^{4}	7.24×10^{3}	1.57×10^{3}
	Std	8.27×10^{3}	6.69×10^{3}	3.85×10^{3}	2.70×10^{3}	4.68×10^{3}	4.00×10^{3}	4.32×10^{3}	3.09×10^{3}
	Mean	1.11×10^{4}	3.05×10^{4}	1.58×10^{4}	3.16×10^{4}	3.14×10^{4}	2.17×10^{4}	1.39×10^{4}	7.42×10^{3}
F10	Min	6.32×10^{3}	7.54×10^{3}	6.63×10^{3}	8.83×10^{3}	1.41×10^{4}	8.43×10^{3}	6.91×10^{3}	8.27×10^{3}
	Std	1.02×10^{3}	2.51×10^{3}	8.59×10^{2}	9.50×10^{2}	5.07×10^{2}	9.19×10^{2}	1.77×10^{3}	5.71×10^{2}
	Mean	8.17×10^{3}	1.24×10^{4}	8.44×10^{3}	1.04×10^{4}	1.50×10^{4}	1.04×10^{4}	1.03×10^{4}	9.19×10^{3}
F11	Min	1.37×10^{3}	2.00×10^{3}	1.29×10^{3}	1.84×10^{3}	7.04×10^{3}	3.15×10^{3}	1.45×10^{3}	1.30×10^{3}
	Std	3.72×10^{2}	4.68×10^{3}	8.00×10^{1}	5.80×10^{2}	2.42×10^{3}	2.85×10^{3}	2.12×10^{2}	2.50×10^{2}
	Mean	1.72×10^{3}	5.77×10^{3}	1.43×10^{3}	3.02×10^{3}	1.12×10^{4}	7.89×10^{3}	1.75×10^{3}	1.59×10^{3}
F12	Min	1.59×10^{7}	2.63×10^{8}	1.05×10^{7}	3.63×10^{8}	3.47×10^{9}	3.74×10^{8}	1.24×10^{7}	2.16×10^{6}
	Std	2.73×10^{9}	7.38×10^{8}	1.73×10^{7}	5.56×10^{8}	5.06×10^{9}	3.45×10^{9}	3.18×10^{7}	1.58×10^{7}
	Mean	2.58×10^{9}	1.16×10^{9}	3.66×10^{7}	8.49×10^{8}	1.09×10^{10}	4.41×10^{9}	4.90×10^{7}	1.95×10^{7}
F13	Min	4.15×10^{4}	2.37×10^{5}	6.65×10^{4}	6.35×10^{6}	3.39×10^{8}	1.37×10^{7}	1.53×10^{4}	2.11×10^{3}
	Std	1.61×10^{9}	1.18×10^{8}	7.42×10^{4}	3.22×10^{7}	1.85×10^{9}	4.70×10^{8}	3.80×10^{4}	7.69×10^{3}
	Mean	9.04×10^{8}	1.08×10^{8}	1.79×10^{5}	2.59×10^{7}	2.05×10^{9}	5.38×10^{8}	6.98×10^{4}	1.00×10^{4}
F14	Min	2.23×10^{5}	3.18×10^{5}	1.51×10^{5}	3.63×10^{5}	3.65×10^{5}	9.14×10^{4}	1.21×10^{4}	1.32×10^{5}
	Std	6.21×10^{6}	4.03×10^{6}	6.75×10^{5}	4.51×10^{6}	6.59×10^{6}	5.48×10^{6}	2.59×10^{5}	3.75×10^{5}
	Mean	2.30×10^{6}	4.03×10^{6}	9.15×10^{5}	4.92×10^{6}	7.15×10^{6}	3.41×10^{6}	2.71×10^{5}	6.21×10^{5}
F15	Min	3.80×10^{3}	3.86×10^{4}	1.18×10^{4}	4.18×10^{5}	5.74×10^{6}	3.54×10^{4}	4.07×10^{3}	1.71×10^{3}
	Std	1.30×10^{7}	2.30×10^{8}	2.79×10^{4}	8.83×10^{5}	2.47×10^{8}	5.61×10^{8}	8.91×10^{3}	3.99×10^{4}
	Mean	2.41×10^{6}	6.15×10^{7}	4.73×10^{4}	1.37×10^{6}	2.06×10^{8}	3.44×10^{8}	1.50×10^{4}	2.32×10^{4}
F16	Min	2.46×10^{3}	3.44×10^{3}	2.67×10^{3}	4.01×10^{3}	4.81×10^{3}	3.50×10^{3}	3.20×10^{3}	2.98×10^{3}
	Std	5.04×10^{2}	6.49×10^{2}	4.64×10^{2}	6.60×10^{2}	5.06×10^{2}	5.01×10^{2}	3.77×10^{2}	3.26×10^{2}
	Mean	3.55×10^{3}	4.84×10^{3}	3.59×10^{3}	4.90×10^{3}	5.95×10^{3}	4.68×10^{3}	3.93×10^{3}	3.83×10^{3}
F17	Min	2.61×10^{3}	3.31×10^{3}	2.44×10^{3}	3.34×10^{3}	3.63×10^{3}	3.14×10^{3}	2.86×10^{3}	2.89×10^{3}
	Std	3.78×10^{2}	5.32×10^{2}	4.11×10^{2}	4.29×10^{2}	5.37×10^{2}	7.09×10^{2}	2.95×10^{2}	2.29×10^{2}
	Mean	3.32×10^{3}	4.20×10^{3}	3.37×10^{3}	3.90×10^{3}	4.73×10^{3}	4.00×10^{3}	3.45×10^{3}	3.54×10^{3}
F18	Min	3.31×10^{5}	3.82×10^{5}	1.04×10^{6}	1.54×10^{6}	2.95×10^{6}	3.06×10^{5}	1.59×10^{5}	9.43×10^{5}
	Std	7.38×10^{6}	9.51×10^{6}	4.22×10^{6}	1.09×10^{7}	2.47×10^{7}	1.52×10^{7}	1.60×10^{6}	4.91×10^{6}
	Mean	6.70×10^{6}	9.41×10^{6}	7.12×10^{6}	1.19×10^{7}	3.73×10^{7}	1.57×10^{7}	1.96×10^{6}	5.78×10^{6}
F19	Min	1.15×10^{4}	6.83×10^{4}	3.65×10^{3}	3.07×10^{5}	7.71×10^{6}	1.49×10^{5}	4.46×10^{3}	2.04×10^{3}
	Std	3.16×10^{6}	7.56×10^{6}	1.71×10^{4}	2.35×10^{6}	6.45×10^{7}	2.16×10^{7}	1.34×10^{4}	1.61×10^{4}
	Mean	1.56×10^{6}	7.93×10^{6}	2.63×10^{4}	2.35×10^{6}	6.26×10^{7}	1.23×10^{7}	2.52×10^{4}	1.90×10^{4}
F20	Min	2.24×10^{3}	3.04×10^{3}	2.95×10^{3}	2.88×10^{3}	3.61×10^{3}	2.83×10^{3}	2.60×10^{3}	2.97×10^{3}
	Std	4.11×10^{2}	3.48×10^{2}	2.89×10^{2}	3.21×10^{2}	2.21×10^{2}	3.17×10^{2}	3.70×10^{2}	1.78×10^{2}
	Mean	3.14×10^{3}	3.79×10^{3}	3.42×10^{3}	3.57×10^{3}	4.24×10^{3}	3.60×10^{3}	3.35×10^{3}	3.44×10^{3}
F21	Min	2.44×10^{3}	2.66×10^{3}	2.48×10^{3}	2.77×10^{3}	2.84×10^{3}	2.67×10^{3}	2.53×10^{3}	2.48×10^{3}
	Std	5.72×10^{1}	8.41×10^{1}	5.54×10^{1}	8.53×10^{1}	4.58×10^{1}	6.23×10^{1}	6.01×10^{1}	3.47×10^{1}
	Mean	2.54×10^{3}	2.88×10^{3}	2.57×10^{3}	2.94×10^{3}	2.94×10^{3}	2.79×10^{3}	2.65×10^{3}	2.56×10^{3}
F22	Min	3.94×10^{3}	9.20×10^{3}	7.75×10^{3}	9.98×10^{3}	1.54×10^{4}	1.01×10^{4}	8.64×10^{3}	9.57×10^{3}
	Std	1.92×10^{3}	2.28×10^{3}	9.69×10^{2}	8.65×10^{2}	4.95×10^{2}	1.17×10^{3}	1.55×10^{3}	7.37×10^{2}
	Mean	1.04×10^{4}	1.25×10^{4}	9.48×10^{3}	1.25×10^{4}	1.71×10^{4}	1.23×10^{4}	1.13×10^{4}	1.10×10^{4}

Table 3. *Cont.*

Function		PSO	DBO	SMA	HHO	SABO	SCSO	TSO	SLTSO
F23	Min	3.11×10^{3}	3.35×10^{3}	2.92×10^{3}	3.69×10^{3}	3.61×10^{3}	3.20×10^{3}	3.12×10^{3}	2.89×10^{3}
	Std	1.70×10^{2}	1.37×10^{2}	5.72×10^{1}	2.12×10^{2}	1.85×10^{2}	1.06×10^{2}	1.83×10^{2}	5.64×10^{1}
	Mean	3.35×10^{3}	3.57×10^{3}	3.04×10^{3}	4.01×10^{3}	3.91×10^{3}	3.39×10^{3}	3.43×10^{3}	3.01×10^{3}
F24	Min	3.27×10^{3}	3.46×10^{3}	3.02×10^{3}	3.96×10^{3}	3.61×10^{3}	3.23×10^{3}	3.39×10^{3}	3.11×10^{3}
	Std	1.96×10^{2}	1.24×10^{2}	9.09×10^{1}	2.16×10^{2}	1.59×10^{2}	1.25×10^{2}	3.11×10^{2}	5.05×10^{1}
	Mean	3.53×10^{3}	3.67×10^{3}	3.19×10^{3}	4.31×10^{3}	3.93×10^{3}	3.54×10^{3}	3.78×10^{3}	3.24×10^{3}
F25	Min	3.08×10^{3}	3.16×10^{3}	3.04×10^{3}	3.34×10^{3}	4.26×10^{3}	4.26×10^{3}	3.14×10^{3}	2.94×10^{3}
	Std	1.12×10^{2}	1.74×10^{3}	3.88×10^{1}	2.23×10^{2}	1.15×10^{3}	5.79×10^{2}	9.71×10^{1}	4.86×10^{1}
	Mean	3.23×10^{3}	3.79×10^{3}	3.10×10^{3}	3.79×10^{3}	6.96×10^{3}	5.15×10^{3}	3.32×10^{3}	3.02×10^{3}
F26	Min	3.71×10^{3}	8.28×10^{3}	2.94×10^{3}	7.59×10^{3}	1.10×10^{4}	6.36×10^{3}	8.07×10^{3}	5.33×10^{3}
	Std	1.66×10^{3}	1.35×10^{3}	2.28×10^{3}	1.39×10^{3}	9.68×10^{2}	2.24×10^{3}	1.32×10^{3}	5.11×10^{2}
	Mean	7.15×10^{3}	1.10×10^{4}	5.42×10^{3}	1.18×10^{4}	1.36×10^{4}	1.08×10^{4}	1.11×10^{4}	6.64×10^{3}
F27	Min	3.45×10^{3}	3.60×10^{3}	3.32×10^{3}	3.87×10^{3}	3.94×10^{3}	3.73×10^{3}	3.58×10^{3}	3.20×10^{3}
	Std	2.45×10^{2}	2.42×10^{2}	1.05×10^{2}	7.13×10^{2}	4.89×10^{2}	2.87×10^{2}	2.39×10^{2}	1.78×10^{-4}
	Mean	3.74×10^{3}	4.01×10^{3}	3.54×10^{3}	4.94×10^{3}	4.74×10^{3}	4.31×10^{3}	3.95×10^{3}	3.20×10^{3}
F28	Min	3.42×10^{3}	3.52×10^{3}	3.32×10^{3}	3.99×10^{3}	6.01×10^{3}	4.30×10^{3}	3.49×10^{3}	3.30×10^{3}
	Std	1.04×10^{3}	2.49×10^{3}	4.71×10^{1}	3.91×10^{2}	8.66×10^{2}	6.50×10^{2}	1.34×10^{2}	2.05×10^{-4}
	Mean	4.47×10^{3}	6.67×10^{3}	3.38×10^{3}	4.77×10^{3}	7.52×10^{3}	5.55×10^{3}	3.84×10^{3}	3.30×10^{3}
F29	Min	3.82×10^{3}	4.69×10^{3}	4.26×10^{3}	5.69×10^{3}	7.09×10^{3}	5.32×10^{3}	4.86×10^{3}	3.99×10^{3}
	Std	4.26×10^{2}	8.66×10^{2}	3.00×10^{2}	8.53×10^{2}	1.70×10^{3}	8.62×10^{2}	4.73×10^{2}	2.11×10^{2}
	Mean	4.66×10^{3}	6.20×10^{3}	4.98×10^{3}	7.06×10^{3}	9.83×10^{3}	6.86×10^{3}	5.52×10^{3}	4.37×10^{3}
F30	Min	1.31×10^{6}	4.69×10^{6}	7.83×10^{6}	3.61×10^{7}	2.02×10^{8}	9.52×10^{7}	1.46×10^{6}	3.34×10^{3}
	Std	8.05×10^{6}	6.09×10^{7}	3.82×10^{6}	7.68×10^{7}	2.23×10^{8}	1.92×10^{8}	1.69×10^{6}	6.54×10^{3}
	Mean	7.40×10^{6}	6.37×10^{7}	1.35×10^{7}	1.38×10^{8}	5.29×10^{8}	2.84×10^{8}	3.75×10^{6}	9.58×10^{3}

Tables 1–3 show that the SLTSO algorithm consistently beats the other seven algorithms in terms of mean, Std, and best scores when tested with benchmark Functions F1–F5 across dimensions of 10, 30, and 50. The mean and Std values of the SLTSO algorithm are clearly more than 30% higher than those of the DBO algorithm after a thorough comparison of the mean, Std, and best values. When compared to the other seven algorithms, the SLTSO algorithm performs better overall when taking into account the combined performance on F1 through F5. In the evaluation of benchmark Functions F6-F17, the SLTSO algorithm achieves favorable mean and best parameter values, signifying good performance. On other multi-modal benchmark functions, the SLTSO algorithm surpasses the PSO, DBO, SMA, HHO, SABO, and SCSO algorithms.

Each of the eight methods achieves the theoretically ideal answer for the best value in the mixed benchmark function testing. While all eight methods for the mixed benchmark Functions F12–F16 attain the ideal mean and best values in theory, the SLTSO algorithm performs better on the standard deviation than the other seven techniques. The SLTSO method converges more quickly and needs fewer iterations than the mean and best values of the other algorithms, even if they both approach the theoretical optimum. The SLTSO algorithm outperforms the others in the mean and best scores across all three benchmark function categories. The SLTSO algorithm uses fewer iterations and performs best overall while obtaining the same degree of accuracy.

6.2. Comparison of Engineering Application

To verify the effectiveness and feasibility of the SLTSO algorithm in engineering applications, welding beam design problems and gear system design problems were selected for analysis.

(1) Welding beam design problem

The welding beam design (WBD) problem is a minimization problem in which the optimization algorithm aims to reduce the manufacturing cost of the design [59]. This optimization problem can be described as finding a solution that satisfies the shear stress (τ), bending stress (θ), beam bending load (Pc), and end deviation (δ). The four design variables constrained by boundary conditions, namely, the length (l), curvature (t), thickness (b), and

weld thickness (h) of the beam, make the cost of manufacturing welded beams the highest. Therefore, the welded beam problem is a typical nonlinear programming problem. The mathematical description of the WBD problem is as follows:

$$\text{Variable}: \vec{l} = [l_1, l_2, l_3, l_4] = [hltb] = [x_1 x_2 x_3 x_4] \tag{36}$$

$$\text{Standard function}: f(\vec{l}) = 1.10471 l_1^2 l_2 + 0.04811 l_3 l_4 (14.0 + l_2) \tag{37}$$

where the objective function represents the cost of manufacturing welded beams, and it is required to solve the problem of cost maximization. The range of decision variable values is as follows: $0.1 \le l_1 \le 2$, $0.1 \le l_2 \le 10$, $0.1 \le l_3 \le 10$, $0.1 \le l_4 \le 2$.

The constraints are $s_1(\vec{l}) = \tau(\vec{l}) - \tau_{\max} \le 0$, $s_2(\vec{l}) = \sigma(\vec{l}) - \sigma_{\max} \le 0$, $s_3(\vec{l}) = \delta(\vec{l}) - \delta_{\max} \le 0$, $s_4(\vec{l}) = l_1 - l_4 \le 0$, $s_5(\vec{l}) = 0.125 - l_1 \le 0$. $s_6(\vec{l}) = 1.10471 l_1^2 + 0.04811 l_3 l_4 (14.0 + l_2) - 5.0 \le 0$,

$s_7(\vec{l}) = 1.10471 l_1^2 + 0.04811 l_3 l_4 (14.0 + l_2) - 5.0 \le 0$, where σ_{max} = 30,000 psi, P = 6000 lb, L = 14 in., $\delta_{\max}$ = 0.25 in., $E = 3 \times 10^6$ psi, τ_{max} = 136,000 psi, and G = 1.2×10^7 psi. The expressions of various functions in the constraint conditions refer to Equations (38)–(42).

$$\tau\left(\vec{l}\right) = \sqrt{\tau\prime^2 + 2\tau\prime\tau'' (l_2/R) + (\tau'')^2} \tag{38}$$

$$\tau = \frac{\mathrm{P}}{\sqrt{2} l_1 l_2}, \ \tau'' = \mathrm{MR/J}, \ \mathrm{M} = \mathrm{p(L} + l_2/2) \tag{39}$$

$$\mathrm{R} = \sqrt{\frac{\left[l_2^2 + (l_1 + l_3)^2\right]}{4}} \tag{40}$$

$$\mathrm{J} = 2\left\{\sqrt{2} l_1 l_2 [\frac{l_2^2}{12} + \frac{(l_1 + l_3)^2}{14}]\right\} \tag{41}$$

$$\mathrm{P_c}\left(\vec{l}\right) = \frac{4.013 \mathrm{E} l_3 l_4^2}{6\mathrm{L}^2}\left(1 - \frac{l_3\sqrt{\mathrm{E}}}{8\mathrm{LG}}\right) \tag{42}$$

In this article, the penalty coefficient transforms constrained optimization problems into unconstrained optimization problems and is used to compare the optimization results with those of the seven other algorithms. The iterative effects of the eight optimization algorithms are shown in Figure 5. A comparison of the results of the welding beam design problem is shown in Table 4.

Table 4. Comparison of optimized design algorithms for welded beams.

Algorithm	l	t	b	h	$f_{\min}$
PSO	0.125	6.3202	8.4483	0.2661	3.4293
DBO	0.3469	3.8941	5.6022	0.5353	2.7293
SMA	0.1956	3.4175	3.4175	0.1989	1.6925
HHO	0.1708	3.7218	9.5616	0.1953	1.7859
SABO	0.2759	2.7468	7.8187	0.2889	2.5531
SLTSO	0.1981	3.3528	9.1916	0.1988	1.6713
TSO	0.1962	3.4175	9.1995	0.1989	1.6931
SCSO	0.1913	3.5671	9.1924	0.1985	1.6897

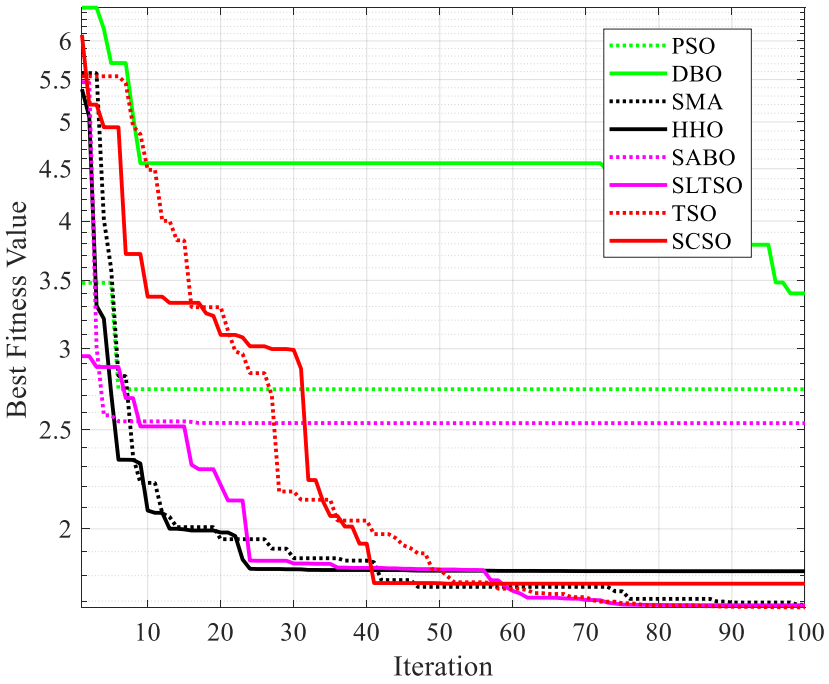


Figure 5. Iterative effect of welding beam design using 8 optimization algorithms.

From the comparison results in Figure 5 and Table 4, it can be deduced that although the SLTSO algorithm may not yield the best individual attribute optimizations for the welded beam, the overall optimization result significantly outperforms the other algorithms. The optimal solution obtained by the SLTSO algorithm for the welding beam is $[x_1, x_2, x_3, x_4]$ = [0.1981, 3.3528, 9.1916, 0.1988], with the optimal objective function value being $f(X)$ = 1.6713. This indicates that the SLTSO algorithm can deliver the most cost-effective optimal design for the welding beam, thereby validating the SLTSO algorithm's superior applicability and stability in practical engineering optimization problems.

(2) Gear train design problem

The gear train design problem is an unconstrained discrete design problem in mechanical engineering, which refers to the most simplified gear system [60]. The gear system is defined as the ratio of the output shaft angular velocity to the input shaft angular velocity. The number of teeth $nA(=x_1)$, $nB(=x_2)$, $nC(=x_3)$ and $nD(=x_4)$ of the gear are considered as design variables, and their mathematical models are as follows:

$$\text{Standard function}: \ f(\mathrm{X}) \ = \ \left(\frac{1}{6.931} - \frac{x_3 x_2}{x_1 x_4}\right)^2 \tag{43}$$

$$\text{Boundary constraints}: \ 12 \leq x_i \leq 60, \ i \ = \ 1,2,3,4 \tag{44}$$

The iterative effect of the gear system design problem using the eight optimization algorithms is shown in Figure 6. A comparison of the results of the gear system design problem is shown in Table 5.

From Figure 6 and Table 5, it can be concluded that in the gear system design benchmark engineering example, the optimal solution of the function optimized by the SLTSO algorithm is $[l, t, b, h]$ = [12, 24.0508, 38.6095, 51.8098], and the optimal value is $f(X) = 1.9259 \times 10^{-32}$. Compared with the standard TSO algorithm, the PSO, DBO, SMA, HHO, SABO, and SCSO algorithms have the best computational results and more expressive power, resulting in the best results. The SLTSO algorithm is thus validated to have superior applicability and stability in practical engineering optimization problems for welding beam design and gear system design.

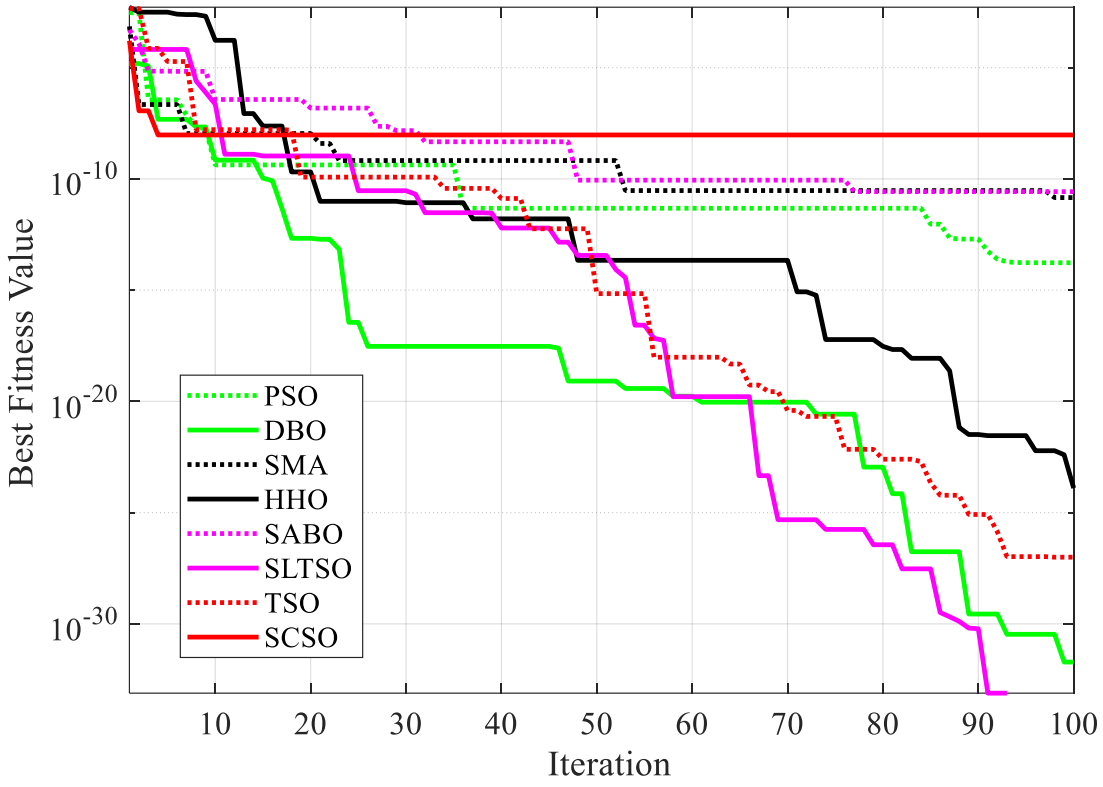


Figure 6. Iterative effect of gear system design using 8 optimization algorithms.

Table 5. Comparison of optimized design algorithms for gear train design.

Algorithm	l	t	b	h	$f_{\min}$
PSO	12	23.869	51.1725	38.7949	1.7009×10^{-14}
DBO	16.592	12	48.651	28.3652	9.9416×10^{-28}
SMA	12	16.3463	36.245	37.5114	2.6689×10^{-11}
HHO	15.2728	27.4423	53.8562	53.9388	1.2646×10^{-24}
SABO	27.5857	14.3694	49.5786	55.4162	1.4442×10^{-11}
SLTSO	12	24.0508	38.6095	51.8098	1.9259×10^{-32}
TSO	15.2097	34.1496	54.3547	55.1563	1.2646×10^{-27}
SCSO	21.0187	33.6516	78.4484	62.5337	9.4023×10^{-9}

6.3. Comparison of UAV Path Planning

A model map of a mountainous environment is an example, where the terrain is characterized by severe undulations and numerous gullies. Environmental disturbances are randomly generated in this terrain, as shown in Figure 7. The starting and ending points of the UAV are set as (10,10,50) and (950950450), respectively (in meters). The number of individuals in the algorithm is uniformly set to 30, and the maximum number of iterations is 50. The SLTSO algorithm and seven other algorithms, including PSO, DBO, SMA, HHO, SABO, SCSO, and standard TSO, were used to solve the model 30 times for each scenario, and the statistical results were obtained. The eight algorithms for path planning and 3D drawing are shown in Figure 8. The top view of path planning for the eight algorithms is shown in Figure 9.

In Figures 8 and 9, it can be seen that the PSO algorithm has the worst quality of the UAV flight route obtained in this flight scenario and the weakest optimization ability, and it is prone to getting stuck in local search, resulting in the worst optimization performance. The TSO algorithm has a stronger evolutionary ability compared with the algorithms included in the comparison, but its local exploration ability needs to be improved. DBO is prone to getting stuck in local search and lacks the ability to jump out of local search,

resulting in an uneven flight path. The flight routes obtained by the other algorithms included in the comparison are relatively smooth, but the paths are longer, resulting in higher flight costs. Seven algorithms, including PSO, DBO, SMA, HHO, SABO, SCSO, and standard TSO, have poor performance in solving unmanned aerial vehicle path planning. The obtained unmanned aerial vehicle flight routes are prone to high flight costs and even flight hazards. The search performance of the SABO algorithm is relatively stable, and compared with the other algorithms, it can obtain better UAV flight routes. However, the optimization performance of the algorithm needs to be improved. According to Figure 9, it can be seen that the SLTSO algorithm has a fast convergence speed for solving the UAV path planning problem in this scenario, and the quality of the solution obtained is significantly higher than that of other the algorithms included in the comparison.

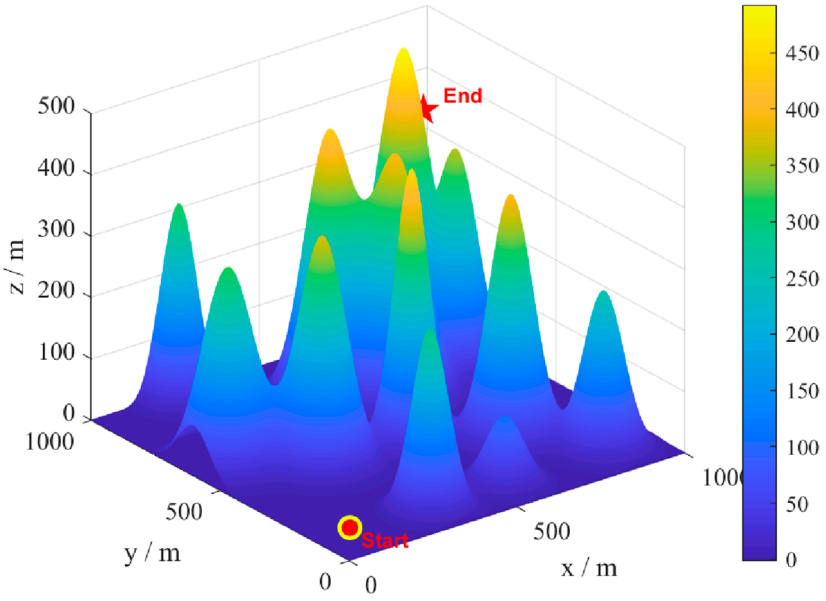


Figure 7. The flight scenario of UAVs.

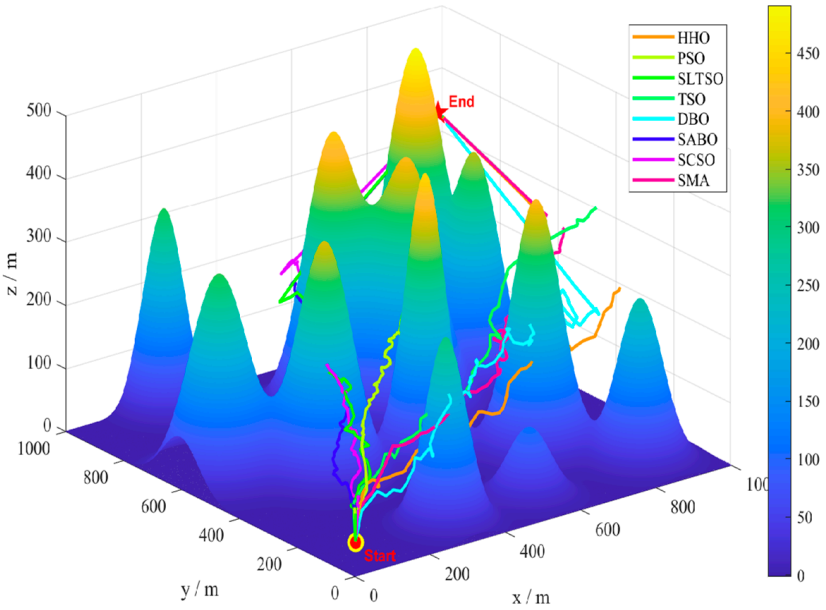


Figure 8. Three-dimensional drawing of path planning for 8 algorithms.

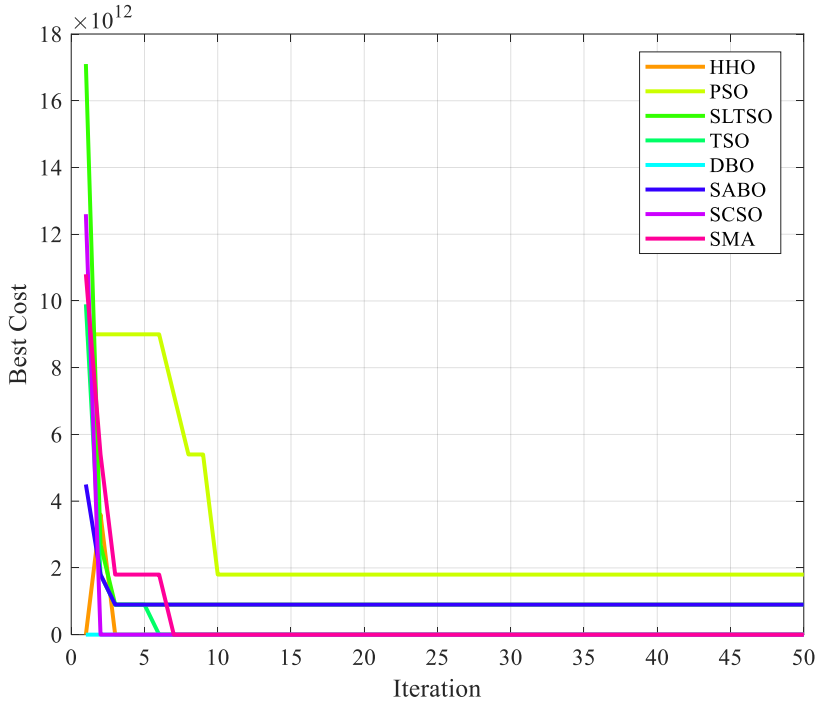


Figure 9. Iterative effect diagram of path planning for 8 algorithms.

The long flight path avoided environmental disturbances, but the low-flying height did not provide safe ground clearance, and the original TSO algorithm iterated into local optima, as shown by comparing the flight routes of TSO and SLTSO. The SLTSO algorithm's fly route is smooth, eliminating environmental interruptions and providing a safe distance and safe ground clearance from impediments. Figures 8 and 9 show how slowly the original TSO algorithm converges, indicating that it is prone to becoming stuck in local optima. This demonstrates that the SLTSO algorithm can escape local optima. The approach presented in this research, which has a faster convergence time and may quickly leap out of local optimal solutions, based on various performances, is feasible for use in planning high-quality paths in challenging hilly terrain.

7. Conclusions

In order to solve the issue of UAV trajectory planning, this paper suggests a sine strategy and a tuna swarm optimization approach based on Levy flight guidance (SLTSO). This method works well for three-dimensional UAV trajectory planning in challenging mission locations. In this algorithm, the population variation in the TSO algorithm is increased by employing the elite opposition-based learning mechanism. The use of a golden sine approach, which effectively synchronizes local excavation ability with global search capacity, accelerates convergence speed. We propose the Levy method to enhance the algorithm's escape from local optima. A lateral crossover with an adaptive technique is employed to improve the convergence accuracy and global optimization potential of the algorithm. These improved strategies enhance the performance of the SLTSO algorithm in three-dimensional UAV route planning. Experimental data and trajectory planning experiments demonstrate that the SLTSO algorithm can effectively find a safe and effective path. Comparative statistics show that the SLTSO approach outperforms other optimization strategies in terms of addressing UAV route planning difficulties. Regarding UAV route planning, the SLTSO algorithm is a useful and effective method.

Even though the SLTSO approach yields higher-quality UAV route planning solutions, it increases computing complexity. It is necessary to improve the performance of optimization algorithms in the future while preserving a simplified algorithmic mechanism

and process. The SLTSO method is expected to find applications in domains such as UAV collaborative route planning and dynamic collision avoidance in the future.

Author Contributions: Conceptualization, Q.W.; writing—review and editing, M.X.; validation, Z.H.; data curation, M.X.; writing—original draft preparation, Q.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Wenzhou Municipal Science and Technology Bureau G20210009, and the Zhejiang Provincial Natural Science Foundation under Grant LD21F020001, LSZ19F020001, and Z20F020022.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon request. There are no restrictions on data availability.

Conflicts of Interest: The authors declare no competing interests.

References

1. Jones, M.; Djahel, S.; Welsh, K. Path-planning for unmanned aerial vehicles with environment complexity considerations: A survey. *ACM Comput. Surv.* **2023**, *55*, 1–39. [CrossRef]
2. Jayaweera, H.M.P.C.; Hanoun, S. Path planning of unmanned aerial vehicles (UAVs) in windy environments. *Drones* **2022**, *6*, 101. [CrossRef]
3. Shahid, N.; Abrar, M.; Ajmal, U.; Masroor, R.; Amjad, S.; Jeelani, M. Path planning in unmanned aerial vehicles: An optimistic overview. *Int. J. Commun. Syst.* **2022**, *35*, e5090. [CrossRef]
4. Maheswaran, S.; Murugesan, G.; Duraisamy, P.; Vivek, B.; Selvapriya, S.; Vinith, S.; Vasantharajan, V. Unmanned ground vehicle for surveillance. In Proceedings of the 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India, 1–3 July 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–5.
5. Wang, J.; Zhao, Z.; Qu, J.; Chen, X. APPA-3D: An autonomous 3D path planning algorithm for UAVs in unknown complex environments. *Sci. Rep.* **2024**, *14*, 1231. [CrossRef]
6. Guo, Y.; Liu, X.; Zhang, W.; Yang, Y. 3D path planning method for UAV based on improved artificial potential field. *Xibei Gongye Daxue Xuebao J. Northwestern Polytech. Univ.* **2020**, *38*, 977–986. [CrossRef]
7. Lee, S.; Yu, H.; Lee, H. Multiagent Q-learning-based multi-UAV wireless networks for maximizing energy efficiency: Deployment and power control strategy design. *IEEE Internet Things J.* **2021**, *9*, 6434–6442. [CrossRef]
8. Zhao, Y.; Zheng, Z.; Liu, Y. Survey on computational-intelligence-based UAV path planning. *Knowl.-Based Syst.* **2018**, *158*, 54–64. [CrossRef]
9. Tang, W.; Cao, L.; Chen, Y.; Yue, Y. Solving Engineering Optimization Problems Based on Multi-Strategy Particle Swarm Optimization Hybrid Dandelion Optimization Algorithm. *Biomimetics* **2024**, *9*, 298. [CrossRef]
10. Yue, Y.; Cao, L.; Lu, D.; Hu, Z.; Xu, M.; Wang, S.; Li, B.; Ding, H. Review and empirical analysis of sparrow search algorithm. *Artif. Intell. Rev.* **2023**, *56*, 10867–10919. [CrossRef]
11. Cao, L.; Chen, H.; Chen, Y.; Yue, Y.; Zhang, X. Bio-Inspired Swarm Intelligence Optimization Algorithm-Aided Hybrid TDOA/AOA-Based Localization. *Biomimetics* **2023**, *8*, 186. [CrossRef] [PubMed]
12. Wang, S.; Cao, L.; Chen, Y.; Chen, C.; Yue, Y.; Zhu, W. Gorilla optimization algorithm combining sine cosine and cauchy variations and its engineering applications. *Sci. Rep.* **2024**, *14*, 7578. [CrossRef] [PubMed]
13. Yue, Y.; Cao, L.; Chen, H.; Chen, Y.; Su, Z. Towards an Optimal KELM Using the PSO-BOA Optimization Strategy with Applications in Data Classification. *Biomimetics* **2023**, *8*, 306. [CrossRef] [PubMed]
14. Cao, L.; Wang, Z.; Wang, Z.; Wang, X.; Yue, Y. An Energy-Saving and Efficient Deployment Strategy for Heterogeneous Wireless Sensor Networks Based on Improved Seagull Optimization Algorithm. *Biomimetics* **2023**, *8*, 231. [CrossRef] [PubMed]
15. Chen, B.; Cao, L.; Chen, C.; Chen, Y.; Yue, Y. A comprehensive survey on the chicken swarm optimization algorithm and its applications: State-of-the-art and research challenges. *Artif. Intell. Rev.* **2024**, *57*, 170. [CrossRef]
16. Zhu, H.; Wang, Y.; Li, X. UCAV path planning for avoiding obstacles using cooperative co-evolution spider monkey optimization. *Knowl.-Based Syst.* **2022**, *246*, 108713. [CrossRef]
17. Shi, J.; Li, T.; Zhang, H.; Lian, X.; Xu, T. Adaptive multi-UAV path planning method based on improved gray wolf algorithm. *Comput. Electr. Eng.* **2022**, *104*, 108377.
18. Awad, A.; Kamel, S.; Hassan, M.H.; Mohamed, F. An enhanced tuna swarm algorithm for optimizing FACTS and wind turbine allocation in power systems. *Electr. Power Compon. Syst.* **2024**, *52*, 863–878. [CrossRef]
19. Yan, Z.; Yan, J.; Wu, Y.; Cai, S.; Wang, H. A novel reinforcement learning based tuna swarm optimization algorithm for autonomous underwater vehicle path planning. *Math. Comput. Simul.* **2023**, *209*, 55–86. [CrossRef]
20. Lv, J.X.; Yan, L.J.; Chu, S.C.; Cai, Z.M.; Pan, J.S.; He, X.K.; Xue, J.K. A new hybrid algorithm based on golden eagle optimizer and grey wolf optimizer for 3D path planning of multiple UAVs in power inspection. *Neural Comput. Appl.* **2022**, *34*, 11911–11936. [CrossRef]

21. Kiani, F.; Seyyedabbasi, A.; Nematzadeh, S.; Candan, F.; Çevik, T.; Anka, F.A.; Randazzo, G.; Lanza, S.; Muzirafuti, A. Adaptive metaheuristic-based methods for autonomous robot path planning: Sustainable agricultural applications. *Appl. Sci.* **2022**, *12*, 943. [CrossRef]
22. Liu, X.; Li, G.; Yang, H.; Zhang, N.; Wang, L.; Shao, P. Agricultural UAV trajectory planning by incorporating multi-mechanism improved grey wolf optimization algorithm. *Expert Syst. Appl.* **2023**, *233*, 120946. [CrossRef]
23. Jarray, R.; Al-Dhaifallah, M.; Rezk, H.; Bouallègue, S. Parallel cooperative coevolutionary grey wolf optimizer for path planning problem of unmanned aerial vehicles. *Sensors* **2022**, *22*, 1826. [CrossRef] [PubMed]
24. Feng, J.; Sun, C.; Zhang, J.; Du, Y.; Liu, Z.; Ding, Y. A UAV Path Planning Method in Three-Dimensional Space Based on a Hybrid Gray Wolf Optimization Algorithm. *Electronics* **2023**, *13*, 68. [CrossRef]
25. Kumar, R.; Singh, L.; Tiwari, R. Novel Reinforcement Learning Guided Enhanced Variable Weight Grey Wolf Optimization (RLV-GWO) Algorithm for Multi-UAV Path Planning. *Wirel. Pers. Commun.* **2023**, *131*, 2093–2123. [CrossRef]
26. Kelner, J.M.; Burzynski, W.; Stecz, W. Modeling UAV swarm flight trajectories using Rapidly-exploring Random Tree algorithm. *J. King Saud Univ. Comput. Inf. Sci.* **2024**, *36*, 101909. [CrossRef]
27. Cao, L.; Wang, L.; Liu, Y.; Yan, S. 3D trajectory planning based on the Rapidly-exploring Random Tree–Connect and artificial potential fields method for unmanned aerial vehicles. *Int. J. Adv. Robot. Syst.* **2022**, *19*, 17298806221118867. [CrossRef]
28. Blasi, L.; D'Amato, E.; Mattei, M.; Notaro, I. Uav path planning in 3d constrained environments based on layered essential visibility graphs. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, *59*, 2359–2375. [CrossRef]
29. Chowdhury, A.; De, D. RGSO-UAV: Reverse Glowworm Swarm Optimization inspired UAV path-planning in a 3D dynamic environment. *Ad Hoc Netw.* **2023**, *140*, 103068. [CrossRef]
30. Phung, M.D.; Ha, Q.P. Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization. *Appl. Soft Comput.* **2021**, *107*, 107376. [CrossRef]
31. Wang, B.H.; Wang, D.B.; Ali, Z.A. A Cauchy mutant pigeon-inspired optimization–based multi-unmanned aerial vehicle path planning method. *Meas. Control* **2020**, *53*, 83–92. [CrossRef]
32. Saeed, R.A.; Omri, M.; Abdel-Khalek, S.; Ali, E.S.; Alotaibi, M.F. Optimal path planning for drones based on swarm intelligence algorithm. *Neural Comput. Appl.* **2022**, *34*, 10133–10155. [CrossRef]
33. Gugan, G.; Haque, A. Path planning for autonomous drones: Challenges and future directions. *Drones* **2023**, *7*, 169. [CrossRef]
34. Vazquez-Carmona, E.V.; Vasquez-Gomez, J.I.; Herrera-Lozada, J.C.; Antonio-Cruz, M. Coverage path planning for spraying drones. *Comput. Ind. Eng.* **2022**, *168*, 108125. [CrossRef]
35. Li, L.L.; Ji, B.X.; Lim, M.K.; Tseng, M.L. Active distribution network operational optimization problem: A multi-objective tuna swarm optimization model. *Appl. Soft Comput.* **2024**, *150*, 111087. [CrossRef]
36. Xie, L.; Han, T.; Zhou, H.; Zhang, Z.-R.; Han, B.; Tang, A. Tuna swarm optimization: A novel swarm-based metaheuristic algorithm for global optimization. *Comput. Intell. Neurosci.* **2021**, *2021*, 9210050. [CrossRef]
37. Fan, C.; Wang, W.; Tian, J. Flexible job shop scheduling with stochastic machine breakdowns by an improved tuna swarm optimization algorithm. *J. Manuf. Syst.* **2024**, *74*, 180–197. [CrossRef]
38. Fu, C.; Zhang, L. A novel method based on tuna swarm algorithm under complex partial shading conditions in PV system. *Sol. Energy* **2022**, *248*, 28–40. [CrossRef]
39. Gou, Y.; Guo, C.; He, M.; Jiang, Y. A Method for Predicting Typical Characteristics of Voltage Sags Based on TSO-XGBoost Algorithm. In Proceedings of the 2023 2nd Asian Conference on Frontiers of Power and Energy (ACFPE), Chengdu, China, 20–22 October 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 99–103.
40. Nanda, B.; Muni, B.P.; Jena, R.K. Enhancing Power Quality in Microgrids with Hybrid Tuna-Glowworm Swarm Optimization Strategy for Renewable Energy Sources. *Energy Technol.* **2023**, *12*, 2300067. [CrossRef]
41. Sheeja, R.; Iqbal, M.M.; Sivasankar, C. Multi-objective-derived energy efficient routing in wireless sensor network using adaptive black hole-tuna swarm optimization strategy. *Ad Hoc Netw.* **2023**, *144*, 103140. [CrossRef]
42. Sun, C.; Shao, Q.; Zhou, Z.; Zhang, J. An Enhanced FCM Clustering Method Based on Multi-Strategy Tuna Swarm Optimization. *Mathematics* **2024**, *12*, 453. [CrossRef]
43. Guo, S.M.; Guo, J.K.; Gao, Y.G.; Guo, P.Y.; Wang, S.C.; Lou, Z.C.; Zhang, X. Research on engine speed control based on tuna swarm optimization. *J. Eng. Res. Rep.* **2022**, *23*, 272–280. [CrossRef]
44. Xiao, R.; Liu, G.; Yi, D.; Liu, B.; Zhuang, Q. Study on prediction model of liquid holdup based on back propagation neural network optimized by tuna swarm algorithm. *Energy Sources Part A Recovery Util. Environ. Eff.* **2023**, *45*, 8623–8641. [CrossRef]
45. Yao, Y.; Li, H.; Zeng, Z.; Wang, C.; Zhang, Y. Clustering Routing Protocol Based on Tuna Swarm Optimization and Fuzzy Control Theory in Wireless Sensor Networks. *IEEE Sens. J.* **2024**, *24*, 17102–17115. [CrossRef]
46. Sihwail, R.; Omar, K.; Ariffin, K.A.Z.; Tubishat, M. Improved harris hawks optimization using elite opposition-based learning and novel search mechanism for feature selection. *IEEE Access* **2020**, *8*, 121127–121145. [CrossRef]
47. Choi, T.J.; Lee, J.H.; Youn, H.Y.; Ahn, C.W. Adaptive differential evolution with elite opposition-based learning and its application to training artificial neural networks. *Fundam. Informaticae* **2019**, *164*, 227–242. [CrossRef]
48. Kaidi, W.; Khishe, M.; Mohammadi, M. Dynamic levy flight chimp optimization. *Knowl.-Based Syst.* **2022**, *235*, 107625. [CrossRef]
49. Jain, A.; Rao, A.C.S.; Jain, P.K.; Hu, Y.C. Optimized levy flight model for heart disease prediction using CNN framework in big data application. *Expert Syst. Appl.* **2023**, *223*, 119859. [CrossRef]

50. Li, M.; Liu, Z.; Song, H. An improved algorithm optimization algorithm based on RungeKutta and golden sine strategy. *Expert Syst. Appl.* **2024**, *247*, 123262. [CrossRef]
51. Tanyildizi, E.; Demir, G. Golden Sine Algorithm: A Novel Math-Inspired Algorithm. *Adv. Electr. Comput. Eng.* **2017**, *17*, 71–78. [CrossRef]
52. Li, Z. A local opposition-learning golden-sine grey wolf optimization algorithm for feature selection in data classification. *Appl. Soft Comput.* **2023**, *142*, 110319. [CrossRef]
53. Chen, C.; Cao, L.; Chen, Y.; Chen, B.; Yue, Y. A comprehensive survey of convergence analysis of beetle antennae search algorithm and its applications. *Artif. Intell. Rev.* **2024**, *57*, 141. [CrossRef]
54. Shen, Q.; Zhang, D.; Xie, M.; He, Q. Multi-Strategy Enhanced Dung Beetle Optimizer and Its Application in Three-Dimensional UAV Path Planning. *Symmetry* **2023**, *15*, 1432. [CrossRef]
55. Kopar, M.; Yıldız, A.R.; Yıldız, B.S. Optimum design of a composite drone component using slime mold algorithm. *Mater. Test.* **2023**, *65*, 1857–1864. [CrossRef]
56. Shi, H.; Lu, F.; Wu, L.; Yang, G. Optimal trajectories of multi-UAVs with approaching formation for target tracking using improved Harris Hawks optimizer. *Appl. Intell.* **2022**, *52*, 14313–14335. [CrossRef]
57. Mukhtar, R.; Chang, C.Y.; Raja, M.A.Z.; Chaudhary, N.I. Design of intelligent neuro-supervised networks for brain electrical activity rhythms of Parkinson's disease model. *Biomimetics* **2023**, *8*, 322. [CrossRef]
58. Niu, Y.; Yan, X.; Wang, Y.; Niu, Y. An improved sand cat swarm optimization for moving target search by UAV. *Expert Syst. Appl.* **2024**, *238*, 122189. [CrossRef]
59. Kaveh, A.; Almasi, P.; Khodagholi, A. Optimum design of castellated beams using four recently developed meta-heuristic algorithms. *Iran. J. Sci. Technol. Trans. Civ. Eng.* **2023**, *47*, 713–725. [CrossRef]
60. Marciniec, A.; Sobolak, M.; Połowniak, P. Graphical method for the analysis of planetary gear trains. *Alex. Eng. J.* **2022**, *61*, 4067–4079. [CrossRef]