Enhanced Salp Swarm Algorithm for Optimizing the Shape of Developable Bézier-Like Surfaces

Jing Lu,1,2 XiaoBo Su,2 Jingyu Zhong,2 and Gang Hu2

1College of Mathematics and Computer Application, Shangluo University, Shangluo 726000, China
2Department of Applied Mathematics, Xi’an University of Technology, Xi’an 710054, China

Correspondence should be addressed to Gang Hu; hugang@xaut.edu.cn

Received 16 June 2022; Revised 19 July 2022; Accepted 22 July 2022; Published 30 August 2022

The shape optimization of developable surfaces is a key and difficult technology in CAD/CAM. This paper studies the shape optimization of generalized developable Bézier-like surfaces using the improved salp swarm algorithm. First, aiming at the problems of slow convergence speed and low calculation accuracy, quickly falling into the optimal local solution for the salp swarm algorithm, and by adjusting the control parameters, the added Morlet wavelet mutation strategy to the individual position updates operation. An improved salp swarm algorithm is proposed, called FMSSA for short. Numerical results show that FMSSA performed the best among 18 (78%) comparisons with other improved SSAs in the CEC2005 test set. In addition, FMSSA serves the best among 16 (69%) comparisons with other algorithms. Therefore, the convergence speed and computational accuracy of FMSSA are improved, and the computational results are significantly better than other intelligent algorithms. Second, according to the duality principle of point and plane, the shape optimization problem of generalized developable Bézier-like surface is transformed into the minimization problem of arc length, energy, and curvature change rate of the dual curve, and three shape optimization models of the developable surface are established. Finally, shape optimization of expandable surfaces is mathematically an optimization problem that the swarm intelligence algorithm can efficiently handle, so FMSSA is used to solve these three optimization models. Numerical experiments show that FMSSA obtains optimal generalized developable Bézier-like surfaces in solving these three optimization models, demonstrating the superiority of the proposed FMSSA in effectively solving the shape optimization models in terms of precision and robustness.

1. Introduction

The developable surface is a special kind of ruled surface that can be unfolded into a complete plane without stretching, folding, and tearing. Because of this excellent property, a developable curved surface has high application value in some manufacturing fields using nontelescopic materials. In recent years, scholars at home and abroad have studied on the theory of developable surfaces. The studies mainly include the design and construction of developable surfaces, fitting, mesh approximation, geometric constraints, developable mesh surfaces, and smooth splicing of developable surfaces. Generally speaking, the construction methods of developable surfaces can be divided into point geometric representation [1–9] and line/surface geometric representation [10–16], also called the dual model.

Point geometry represents a developable surface as a tensor product surface in Euclidean space. This method was first proposed by Weiss and Furtner [1] in 1988, but no specific expression of the developable surface was given. In 1991, Aumann [2] constructed developable Bézier surfaces using design and adjoint curves located on two parallel planes. The two curves are cubic and quartic Bézier curves, respectively. In 1997, Ye et al. [3] extended this method, and he used two Bézier curves of order \((n + 1)\) and order \((n + m + 1)\) as design and adjoint curves to construct developable surfaces. Therefore, the design curve and adjoint curve do not need to be confined to two parallel planes, and the construction conditions are simplified. In 2013, Li et al. [4] proposed a method for constructing interpolation developable surfaces using curvature lines. In addition, other scholars have done further research on constructing developable surfaces by the point geometry method [6–9].
The line/plane geometric representation regards the developable surface as a curve in 3D projective space and constructs the developable surface according to the duality principle between points and planes. In 1993, Boduluri and Ravani [10] used dual representation to construct cubic Bézier and cubic developable B-spline surfaces for the first time and gave explicit expressions. At the same time, they also studied the splicing problem of developable surfaces and put forward the splicing algorithm of Decaster Jau and Farin–Boehm. In 1995, Pottmann and Farin [11] used the same method to construct rational Bézier and rational developable B-spline surfaces. In 1999, Pottmann and Wallner [12] extended this method to constructing NURBS developable surfaces and achieved good results. In 2004 and 2007, Zhou et al. [13, 14] constructed the developable surfaces of Bézier and B-spline four times and the developable surfaces of Bézier and B-spline five times, respectively. In 2013, Zhou et al. [15] constructed a developable C-Bézier surface with a shape parameter by using the 3-degree C-Bézier basis function. Given the control plane coordinates, the overall shape of the developable surface can be modified by changing the size of the shape parameter, but the local shape cannot be adjusted. In 2017, Hu and Cao [16] constructed a generalized developable Bézier-like surface with three shape parameters. The whole and local shapes of developable surfaces can be flexibly modified, and the shape adjustability is significantly improved.

Many optimization problems exist in the modeling design and application research of developable surfaces. The most involved is the shape optimization of developable surfaces, so how to solve these optimization problems quickly and accurately is a research hotspot in CAD/CAM. Traditional optimization methods of single-point evolution require that the objective function satisfy convexity and differentiability. In theory, the algorithm is guaranteed to converge to the optimal solution, which has the advantage of high accuracy. However, it is difficult to use the optimization problem, which requires solid real-time performance, or the objective function does not meet the requirements. The expression of a generalized developable Bézier-like surface is a nonlinear function of its shape parameters. If the shape parameters are taken as optimization variables, it is not easy to solve the shape optimization model of generalized developable Bézier-like surfaces by traditional methods. Compared with conventional methods, the swarm intelligence optimization algorithm is simple, easy to operate, and flexible. It has lower requirements on the objective function and faster convergence and optimization speed, which can better solve the shape optimization problem of generalized developable Bézier-like surfaces.

Generally, swarm intelligence optimization algorithms are derived from imitating the behaviors of some biological groups in nature. Extensive applications include monarch butterfly optimization (MBO) [17]; MBO is an optimization algorithm inspired by the migration behavior of monarch butterflies, which mainly consists of two strategies to simulate the behavior. The MBO algorithm has a simple algorithm structure and iterates more rapidly, but it cannot guarantee the retention of better dimensions. The slime mould algorithm (SMA) [18] uses adaptive weights to simulate the process of generating positive and negative feedback from a bio-oscillator-based propagation wave of mucilaginous bacteria to form optimal pathways connecting foods with a propensity for good exploration and exploitation. SMA has fast convergence and excellent search capability. The moth search algorithm (MSA) [19] is a search algorithm inspired by the light-taming of moths and the flight of Lévy. The algorithm is capable of global superiority and less likely to fall into local extremes but suffers from the disadvantage of slow optimization. Hunger games search (HGS) [20] is designed by hunger-driven activities and behavioral choices of animals. The main features of the method are its dynamics, structural simplicity, and high performance in terms of acceptable solution quality. However, it is often prone to fall into local solutions. The Runge–Kutta method (RUN) [21] introduces the mathematical basis and ideas of a method widely known in mathematics as Runge Kutta. The algorithm has superior exploration and development trends. However, it is limited in dealing with high-dimensional problems. Colony predation algorithm (CPA) [22], the weighted mean of vectors (INFO) [23], and Harris Hawk Optimization (HHO) [24], the main inspiration for the HHO algorithm is the cooperative behavior and pursuit style of the Harris Hawk in nature, etc. Other representative metaheuristic intelligent optimization algorithms are detailed in the works of literature [25–29].

The salp swarm algorithm (SSA) studied in this paper was proposed by Mirjalili et al. [30] in 2017. It describes the foraging habits of a class of marine organisms. SSA has been widely noticed and applied in various fields since its proposal due to its advantages, such as simplicity and strong robustness, and has achieved good results. For example, Mohammad et al. proposed an improved salp swarm algorithm based on adversarial learning for feature selection [31]. Chaudhary et al. used the SSA algorithm to solve the problem of stochastic wind power multiregion economic dispatch [32]. Naderipour et al. used the SSA algorithm for multistage PV system carrier optimization to improve power quality in the industrial environment [33]. Gholami et al. introduced a mutated salp swarm algorithm and used it for the optimal allocation problem of active and reactive power sources in radial distribution systems [34]. Sultan et al. combined the SSA algorithm with the crossover strategy and greedy mechanism and used it as a parameter identification for proton exchange membrane fuel cells [35]. Lu et al. used a multiobjective salp swarm algorithm to optimize the relevance vector machine to train an ensemble model for predicting the burst pressure of corroded pipelines [36]. However, the application and study of the SSA algorithm have also shown the problem of slow convergence quickly falling into optimal local solutions. Therefore, one objective of this paper is to propose an improved SSA (i.e., FMSSA) with higher solution accuracy and better performance by adjusting the control parameters of SSA, adding the Morlet wavelet mutation strategy to the individual position update mechanism, proposing an improved salp swarm algorithm (FMSSA). FMSSA retains the original
algorithm’s advantages and improves the algorithm’s convergence speed and calculation accuracy. Using the proposed FMSSA, we can solve the shape optimization model of generalized developable Bézier-like surfaces. In summary, the main contributions of this paper are as follows:

(i) To effectively solve the shape optimization problem of generalized developable Bézier surfaces, three shape optimization models of developable surfaces are developed by transforming the shape optimization problem of generalized, extensible Bézier surfaces into the issue of minimizing the arc length, energy, and curvature rate of change of the pairwise curves based on the principle of the pairwise pair.

(ii) An enhanced salp swarm algorithm called FMSSA is developed, which combines a tuning control parameter strategy and a Morlet wavelet variation strategy to increase the diversity and exploitation capability of the population. The performance of FMSSA was evaluated in the prestigious CEC 2005.

(iii) The three optimization models are solved using the proposed FMSSA and fuzzy support vector machine and intelligent algorithms such as SCA, WOA, MFO, MVO, SSA, HBA, and WSO. The results show that FMSSA can effectively solve the developable surface shape and optimization models.

The rest of this paper is arranged as follows: In the second section, the explicit expressions of generalized developable Bézier-like surfaces are given; in the third section, the basic principle of SSA is introduced and puts forward an improved algorithm FMSSA; in the fourth section, the shape optimization models of three groups of generalized developable Bézier-like surfaces are established, and several representative experiments verify the validity and accuracy of the optimization model based by FMSSA; and the fifth section summarizes the main work of this paper.

2. Preliminaries

Given control points \( P \in \mathbb{R}^n (n = 2, 3) \), \( P = [P_0, P_1, P_2, P_3] \), the matrix expression of the generalized cubic Bézier-like curve can be written as

\[
\left\{ \prod_{t} \right\} : L(t; \Omega) = TMP^T, \quad 0 \leq t \leq 1, \quad (1)
\]

where

\[
M = \begin{pmatrix}
\lambda_1 & -\lambda_1 - \lambda_2 & \lambda_2 - \lambda_3 & \lambda_3 \\
-3\lambda_1 - 1 & 3\lambda_1 + 2\lambda_2 + 3 & -2\lambda_2 + \lambda_3 - 3 & 1 - \lambda_3 \\
3\lambda_1 + 3 & -6 - 3\lambda_1 - \lambda_2 & 3 + \lambda_2 & 0 \\
-\lambda_1 - 3 & 3 + \lambda_1 & 0 & 0 \\
1 & 0 & 0 & 0
\end{pmatrix}.
\]

\[
T = [t^4, t^3, t^2, t, 1], \quad \Omega = [\lambda; \lambda_1, \lambda_2, \lambda_3], \lambda_1, \lambda_3 \in [-3, 3], \lambda_2 \in [-3, 3], \text{and } t \text{ and } \lambda_i \text{ are the family parameter and shape control parameters of } [\prod_t], \text{respectively; } M \text{ is called coefficient matrix.}
\]

According to the point-to-planes dual theorem, combining (1), the generalized cubic Bézier-like plane family equation can be obtained as

\[
\left\{ \prod_{t} \right\} : U(t; \Omega) = TMQ^T, \quad 0 \leq t \leq 1. \quad (3)
\]

In equation (3), \( T \) and \( M \) have the same values as in equation (1), and \( Q = (a_i, b_i, c_i, d_i), i = 0, 1, 2, 3 \) represents the coordinates of the control plane. Then, equation (3) can also be expressed as

\[
U(t; \lambda_1, \lambda_2, \lambda_3) = \sum_{i=0}^{3} F_{i,t}(t)Q_i
\]

\[
= \left\{\sum_{i=0}^{3} F_{i,t}(t)a_i, \sum_{i=0}^{3} F_{i,t}(t)b_i, \sum_{i=0}^{3} F_{i,t}(t)c_i, \sum_{i=0}^{3} F_{i,t}(t)d_i\right\}. \quad (4)
\]

\( F_{i,t}(t) (i = 0, 1, 2, 3) \) is the generalized cubic Bézier-like basis function, its computational formula are as follows:

\[
\begin{align*}
F_{0,t}(t) &= 1 - (\lambda_1 + 3)t + (3\lambda_1 + 3)t^2 - (3\lambda_1 + 1)t^3 + \lambda_1 t^4, \\
F_{1,t}(t) &= (\lambda_1 + 3)t - (6 + 3\lambda_1 + \lambda_2)t^2 + (3\lambda_1 + 2\lambda_2 + 3)t^3 - (\lambda_1 + \lambda_2)t^4, \\
F_{2,t}(t) &= (3 + \lambda_2)t^2 - (2\lambda_2 - \lambda_3 + 3)t^3 + (\lambda_2 - \lambda_3)t^4, \\
F_{3,t}(t) &= (1 - \lambda_3)t^3 + \lambda_3 t^4.
\end{align*}
\]
Let
\[
\begin{aligned}
&u_0(t) = \sum_{i=0}^{3} F_{i,4}(t)a_i, \\
&u_1(t) = \sum_{i=0}^{3} F_{i,4}(t)b_i, \\
&u_2(t) = \sum_{i=0}^{3} F_{i,4}(t)c_i, \\
&u_3(t) = \sum_{i=0}^{3} F_{i,4}(t)d_i.
\end{aligned}
\]
Equation (3) can be written as
\[
\prod_{i=1}^{3} u(t; \Omega) = \{u_0(t), u_1(t), u_2(t), u_3(t)\}.
\]

2.1. Generalized Enveloping Developable Bézier-Like Surfaces. According to the definition of developable surface, the envelope surface of the single-parameter family of plane \(\prod_{i=1}^{3} u(t)\) is a developable surface, so the generalized enveloping developable Bézier-like surface can be expressed as
\[
H(k, t; \lambda_1, \lambda_2, \lambda_3) = kU(t) + \psi(t), \quad k \in (-\infty, +\infty).
\]

2.2. Generalized Spine Curve Developable Bézier-Like Surfaces. Generalized spine curve developable Bézier-like surfaces can be defined as
\[
S(v, t; \lambda_1, \lambda_2, \lambda_3) = \rho(t) + \nu v',
\]
where
\[
\rho(t) = u_0''(t)&(\nu t \times \nu') + u_1(t)[\nu'(t) \times \nu''(t)] + u_2''(t)[\nu''(t) \times \nu(t)] \\
&\nu'(t) \cdot \nu''(t).
\]

3. Improved Salp Swarm Algorithm

3.1. Salp Swarm Algorithm. Salp swarm is a kind of invertebrate in the ocean, which mainly lives in the Southern Ocean and swings forward by absorbing water. They usually form a group chain in the deep sea, divided into two parts as leader and followers, which is called a salp swarm chain. Leaders are individuals at the forefront of the group chain, and the rest are followers. While looking for food, followers move harmoniously under the guidance of leaders, constantly update their positions, and explore and develop better food sources in the search space as the group’s goal. The mathematical model expresses the foraging activity, and the initial position of the leader is
\[
x_i^1 = \left\{ \begin{array}{ll}
F_j^1 + c_1\left( [b_j^u - b_j^l]c_2 + b_j^l \right), & c_3 \geq 0, \\
F_j^1 - c_1\left( [b_j^u - b_j^l]c_2 + b_j^l \right), & c_3 < 0.
\end{array} \right.
\]

The follower’s position update formula is
\[
x_i^j = \frac{1}{2}(x_i^{j-1} + x_i^{j-1})
\]
In which, \(x_i^1\) said the leader, is the first \(j\). The first individual in the ascidian chain; \(x_i^j\) represents the first \(j\) The first in the dimension group \(i\) The position of a follower individual \((i \geq 2); F_i^j\) For the first \(j\) The food source position of dimension. \(b_j^u\) and \(b_j^l\) are the upper and lower bounds of the search space; \(c_1\) is used to control the population exploration and development parameters, defined as
\[
c_1 = 2e^{-(4L)^2},
\]
where \(c_2\) and \(c_3\) are uniformly distributed random numbers in \([0,1]\), \(L\) is the iteration times of the current population, and \(L\) is the maximum number of iterations.

3.2. An Improved Salp Swarm Algorithm

3.2.1. Control Parameter Adjustment Strategy. In SSA, control parameter \(c_1\) is essential in balancing the algorithm’s global exploration and local development ability, which weighs the parameter of population exploration and development. Give \(c_1\) an appropriate coefficient \(\delta\) which will accelerate the convergence speed of the algorithm. After improvement, \(c_1\) can be expressed as
\[
c_1 = \delta \cdot 2e^{-(4L)^2}.
\]

Note: After many tests, it is found that the performance of the algorithm improves the most when \(\delta\) takes 0.01.

3.2.2. Morlet Wavelet Mutation Strategy. Because optimization problems with dense distribution of extreme points easily fall into local optimum, jumping out of local optimum is the key to the swarm intelligence optimization algorithm. Mutation is the fundamental reason for biological evolution. Common mutation strategies include Gaussian mutation, Cauchy mutation, and so on, while wavelet mutation realizes the dynamic adjustment of mutation space through the wavelet function’s
translation and expansion properties. Improve the stability of the solution, integrate wavelet mutation into SSA, reduce the amplitude of the function by controlling the expansion parameters of a wavelet function, make the mutation space constrained by iteration times, realize the fine-tuning effect of mutation operation, enhance the ability of the algorithm to jump out of local optimum, and improve the convergence speed and calculation accuracy of the algorithm. The leader position after Morlet wavelet mutation is still expressed by formula (12), and the follower population position’s update formula becomes

\[ x_j^l = \frac{1}{2} \sigma \left( x_j^l + x_j^{l-1} \right), \tag{16} \]

where \( \sigma \) is the wavelet coefficient of variation as is follows

\[ \sigma = \frac{1}{\sqrt{a}} \psi \left( \frac{\phi}{a} \right). \tag{17} \]

Among them, \( \psi(x) \) is Morlet wavelet function, which is defined as

\[ \psi(x) = e^{-x^2/2} \cos(5x). \tag{18} \]

Because of wavelet function \( \psi(x) \), more than 99% of the energy is contained in the interval \([-2.5, 2.5]\); therefore, \( \phi \) for \([-2.5a, 2.5a]\) is a random number within. The scalation parameter \( a \) is

\[ a = s \cdot \left( \frac{1}{2} \right{l} \right)^{(1-\frac{l}{L})}, \tag{19} \]

where \( l \) is the current iteration number, \( L \) is the maximum number of iterations, and \( s \) is a random number.

3.3. FMSSA Algorithm Steps and Pseudo Code. To improve the calculation accuracy of the SSA algorithm convergence rate, the control parameters are modified based on the SSA algorithm \( c_1 \). This paper proposes an improved fusion parameter adjustment and Morlet wavelet mutation’s salp swarm algorithm by adding the Morlet wavelet mutation strategy to the individual position update operation. Table 1 provides the pseudo code for FMSSA. The algorithm steps are as follows:

**Step 1.** Initialize control parameters \( c_2, c_3 \), maximum number of iterations \( L \), population size \( N \), function dimension dim, and search. The upper and lower bounds of cable space are \( b_j^U \) and \( b_j^L \) \((j = 1, 2, \ldots, \text{dim})\).

**Step 2.** Generate initialization population \( X_0(x_j^i, i = 1, 2, \ldots, N) \). The fitness value of each individual in the population calculated is \( F(x_j^i) \).

**Step 3.** Order \( l = 1 \) according to the type (15) to generate control parameters \( c_1 \).

**Step 4.** When \( i = 1 \), update the leaders’ position by formula (12). When \( i > 1 \), update the follower’s position by formula (16).

**Step 5.** Check whether the updated individual position exceeds the search space, and modify it to get a new population \( X_l \), calculate the new population \( X_l \). The fitness value of each individual in the system is sorted.

**Step 6.** When \( l < L \), make \( x_j^l = x_j^{l+1}, l = l + 1 \), and return to Step 4.

**Step 7.** When \( l = L \), output the global optimal value, and the algorithm ends.

3.4. Experimental Set-up. To verify the improvement of SSA performance by parameter adjustment strategy and Morlet wavelet mutation strategy, we conducted a two-part experiment: (1). FMSSA was compared with the original SSA and three other improved SSA (ISSA, ESSA, and MESSA). (2). FMSSA is further compared with excellent, intelligent algorithms such as WOA, SCA, MFO, MVO, and MPA. The parameter settings are given in Table 2. We selected the famous CEC2005 test function [25] to compare FMSSA with other algorithms. For fairness, in the comparative experiment, other conditions are consistent except the algorithm and the population’s size \( N = 30 \). The maximum number of iterations of the algorithm \( t_{\text{max}} = 500 \). For each test function, each algorithm runs independently 20 times, and the calculation formula of the average accuracy (Ave) and standard deviation (STD), which measures the advantages and disadvantages of the algorithm is as follows:

\[ \text{Ave} = \frac{1}{\text{runs}} \sum_{i=1}^{\text{runs}} \text{value}_i, \]

\[ \text{STD} = \sqrt{\frac{1}{\text{runs} - 1} \sum_{i=1}^{\text{runs}} (\text{value}_i - \text{Ave})^2}, \tag{20} \]

Among them, \( \text{value}_i \) represents the results of the \( i \)-th run of the algorithm.

3.5. Experimental Set-up. Table 3 records the average value and standard deviation of FMSSA, SSA, ISSA, ESSA, and MESSA, then independently run 20 times on the test function F1–F23, and the bold data are the optimal values and the results of the Friedman test. First, it can be found from Table 3 that FMSSA ranks first with a mean rank of 1.6. Observing Table 3, it can be found that FMSSA outperforms the other algorithms in terms of mean and standard deviation on the other 18 test functions except for F8, F14, F21, F22, and F23. FMSSA shows optimal performance in all the unimodal functions from F1–F7. It indicates that FMSSA has good exploration and development capability, and for the multimodal functions of F8–F13, five functions of FMSSA can reach the optimum, which shows that the proposed algorithm can jump out of the local optimum solution well and explore the optimum solution quickly. There are still six functions that exhibit optimality in the face of fixed-dimension functions, which also show the excellent exploration ability of FMSSA. The parameter adjustment strategy increases the algorithm’s flexibility to search for optimal solutions by adjusting the parameters of the foraging activity phase of the SSA algorithm and therefore improves the search ability in the face of unimodal functions. On the other hand, the Morlet
Algorithm 1: FMSSA

Begin
Initialize FMSSA related parameters, such as $L$, $N$, $\text{dim}$, $c^2$, $c^3$;
Randomly generate initial population $x_i^j$ ($i = 1, 2, \ldots, N; j = 1, 2, \ldots, \text{dim}$);
Calculate the fitness of each search agent (salp) $F$ is the best search agent;
$l = 1$;
While ($l < L$) Update $c_i$ by equation (14);
For each salp ($x_i^j$) If ($i$ $<$ 1)
Update the position of the leading salp by equation (11);
Else Update the position of the follower salp by equation (15);
End If
End For
Amend the salps based on the upper and lower bounds of variables;
$l = l + 1$;
End While
Return $F$;
end

Table 1: Pseudo code of FMSSA.

Table 2: Parameter setting.

| Algorithms | Parameter’s settings |
|------------|----------------------|
| SCA        | $a$ (constant) = 2, $a$ = decreased from 2 to 0, $b$ = 2. |
| WOA        | $a$ (The convergence constant) = $[-2, -1]$, $b$ (spiral factor) = 1. |
| MFO        | Wormhole existence probability $WEP_{\text{Max}} = 1$, $WEP_{\text{Min}} = 0.2$. |
| MVO        | $\gamma > 1$, $P = 0.0$. |
| SSA        | Leader position update probability $c_3 = 0.5$. |
| FMSSA      | $\delta = 0.01$, $\varphi = [-2.5a, 2.5a]$, leader position update probability $c_3 = 0.5$. |
| HBA        | $\beta = 6$, $C = 2$. |
| WSO        | $f_{\text{min}} = 0.07$, $f_{\text{max}} = 0.75$, $\tau = 4.125$. |

Table 3: Compare the results of FMSSA and other improved SSA.

| Functions | Index | SSA | ISSA | ESSA | MSSA | FMSSA |
|-----------|-------|-----|------|------|------|-------|
| F1        | Ave   | 2.44E-07 | 2.61E-18 | 3.42E-37 | 1.12E-12 | 1.69E-109 |
|           | STD   | 3.62E-11 | 6.33E-24 | 1.08E-40 | 3.11E-13 | 2.11E-110 |
|           | RANK  | 5    | 5    | 2    | 4    | 1     |
| F2        | Ave   | 2.5007 | 4.95E-10 | 6.03E-28 | 0.5043 | 6.35E-58 |
|           | STD   | 0.2293 | 1.99E-12 | 4.29E-30 | 0.0791 | 4.85E-59 |
|           | RANK  | 5    | 3    | 2    | 4    | 1     |
| F3        | Ave   | 1.35E+03 | 1.84E-17 | 7.76E-28 | 1.778  | 1.18E-110 |
|           | STD   | 6.89E+02 | 1.19E-20 | 9.46E-31 | 0.3505 | 4.24E-115 |
|           | RANK  | 5    | 3    | 2    | 4    | 1     |
| F4        | Ave   | 1.08E+01 | 1.27E-09 | 1.28E-23 | 2.5699 | 1.95E-56 |
|           | STD   | 2.9539 | 2.84E-10 | 5.23E-25 | 0.8804 | 1.85E-57 |
|           | RANK  | 5    | 3    | 2    | 4    | 1     |
| F5        | Ave   | 3.77E+02 | 2.88E+02 | 8.76E+01 | 4.39E+01 | 2.80E+01 |
|           | STD   | 9.5945 | 7.7411 | 6.5912 | 3.73E+02 | 3.71E-01 |
|           | RANK  | 5    | 4    | 3    | 2    | 1     |
| F6        | Ave   | 1.37E+07 | 3.57E+06 | 1.3272 | 9.08E-13 | 9.08E-13 |
|           | STD   | 1.75E+10 | 0.1132 | 0.4541 | 1.77E-14 | 1.77E-14 |
|           | RANK  | 3    | 5    | 4    | 3    | 1     |
| F7        | Ave   | 0.1463 | 7.92E-05 | 9.64E-05 | 8.28E-02 | 1.43E-05 |
|           | STD   | 0.0538 | 4.09E-06 | 5.38E-07 | 3.72E-05 | 1.46E-07 |
|           | RANK  | 5    | 2    | 3    | 4    | 1     |
| F8        | Ave   | -7.63E+03 | -3.84E+03 | -5.93E+03 | -7.43E+03 | -6.57E+03 |
|           | STD   | 1.27E+02 | 2.21E+02 | 3.27E+02 | 4.51E+02 | 5.29E+02 |
|           | RANK  | 1    | 5    | 4    | 2    | 3     |
wavelet variation strategy enhances the capability of the development phase of the SSA algorithm and enables the algorithm to find the optimal solution steadily, mainly in the face of multimodal and fixed-dimension functions.

Figure 1 shows their convergence curves on some test functions. From the figure, it can be found that FMSSA has a faster convergence rate in the early stages of the iteration than the other three improved algorithms for the unimodal functions F1, F2, F3, F4, F5, F6, and F7 of the test functions. This result is due to the parameter adjustment strategy that helps the SSA algorithm to locate the optimal solution better. While the unimodal function is mainly a comparison of accuracy, FMSSA has better convergence accuracy, which is also the result of the Morlet wavelet variation strategy that improves the development capability of SSA algorithm. In the face of multimodal function, F10, F12, and F13 all converge to the optimum and converge very fast, and the convergence is poor when facing fixed-dimension functions (F14, F21, F22, and F23), which may be due to the speedier convergence speed and not converging to the optimal solution. Also, these are some of the algorithm’s problems, which tend to fall into local situations. Table 4 provides the results of the rank-sum test for FMSSA and other improved SSAs. Besides, “+” indicates that the number of comparison algorithms is superior to FMSSA, “−” suggests otherwise, and “0” shows the number of FMSSA. The comparison algorithms have the same performance in terms of statistics. The significance level is 0.05. From the results, it can be found that FMSSA outperforms the other enhanced algorithms.

| Functions | Index | SSA | ISSA | ESSA | MSSA | FMSSA |
|-----------|-------|-----|------|------|------|-------|
| F9        | Ave   | 4.95E+01 | 0 | 0 | 1.29E+02 | 0 |
|           | STD   | 0.2391 | 0 | 0 | 1.86E+01 | 0 |
|           | RANK  | 4 | 1 | 1 | 5 | 1 |
| F10       | Ave   | 2.5458 | 9.71E-10 | 8.88E-16 | 4.34E-16 | 8.88E-16 |
|           | STD   | 0.5851 | 7.15E-11 | 6.89E-17 | 3.26E-17 | 3.26E-17 |
|           | RANK  | 4 | 3 | 1 | 5 | 1 |
| F11       | Ave   | 0.0183 | 1.11E-16 | 1.29E-14 | 0.1394 | 0 |
|           | STD   | 0.0029 | 9.24E-17 | 6.46E-15 | 3.90E-02 | 0 |
|           | RANK  | 4 | 2 | 3 | 5 | 1 |
| F12       | Ave   | 6.6836 | 0.2814 | 0.2691 | 6.8743 | 3.86E-07 |
|           | STD   | 0.0249 | 0.0057 | 0.0987 | 0.3061 | 7.96E-09 |
|           | RANK  | 4 | 3 | 2 | 5 | 1 |
| F13       | Ave   | 1.76E+01 | 2.4363 | 2.1912 | 3.29E-03 | 1.59E-13 |
|           | STD   | 1.6233 | 9.06E-02 | 0.1621 | 5.31E-04 | 8.88E-16 |
|           | RANK  | 5 | 5 | 3 | 2 | 1 |
| F14       | Ave   | 1.2463 | 1.0011 | 3.9683 | 1.1458 | 4.56E+05 |
|           | STD   | 0.0546 | 2.11E-06 | 0.0146 | 0.0673 | 1.27E-01 |
|           | RANK  | 3 | 1 | 4 | 2 | 5 |
| F15       | Ave   | 0.0039 | 5.89E-04 | 4.76E-04 | 5.24E-04 | 4.13E-04 |
|           | STD   | 0.0001 | 1.43E-05 | 7.13E-07 | 1.68E-05 | 8.28E-05 |
|           | RANK  | 4 | 2 | 1 | 3 | 2 |
| F16       | Ave   | 1.68E-14 | -1.0316 | -1.0314 | -1.0314 | -1.0314 |
|           | STD   | 1.68E-14 | 2.59E-07 | 3.28E-14 | 5.19E-04 | 0 |
|           | RANK  | 1 | 5 | 1 | 1 | 1 |
| F17       | Ave   | 0.3979 | 0.4008 | 0.3979 | 0.3979 | 0.3979 |
|           | STD   | 1.08E-05 | 3.47E-08 | 7.63E-07 | 8.14E-07 | 0 |
|           | RANK  | 1 | 5 | 1 | 1 | 1 |
| F18       | Ave   | 1.70E-13 | 3.0061 | 3.47E-13 | 6.79E-12 | 0 |
|           | STD   | 1.70E-13 | 3.21E-14 | 3.47E-13 | 6.79E-12 | 0 |
|           | RANK  | 1 | 5 | 1 | 1 | 1 |
| F19       | Ave   | -3.8628 | -3.828 | -3.856 | -3.8628 | -3.8628 |
|           | STD   | 3.28E-16 | 7.58E-03 | 4.58E-11 | 3.28E-16 | 3.28E-16 |
|           | RANK  | 1 | 5 | 4 | 1 | 1 |
| F20       | Ave   | -3.225 | -2.8624 | -3.1164 | -3.1838 | -3.322 |
|           | STD   | 0.0585 | 6.16E-02 | 5.85E-02 | 6.14E-02 | 1.12E-02 |
|           | RANK  | 2 | 5 | 4 | 3 | 1 |
| F21       | Ave   | -8.1464 | -4.29627 | -1.01E+01 | -1.0E+00 | -8.3904 |
|           | STD   | 0.3209 | 2.72E-03 | 3.21E-02 | 1.37E-04 | 2.9169 |
|           | RANK  | 4 | 5 | 1 | 1 | 3 |
| F22       | Ave   | -9.159 | -4.6361 | -9.159 | -1.0E+00 | -8.6761 |
|           | STD   | 2.5938 | 0.31131 | 0.1554 | 4.42E-03 | 2.8058 |
|           | RANK  | 2 | 5 | 2 | 1 | 4 |
| F23       | Ave   | -7.5267 | -4.8592 | -10.5 | -1.0E+01 | -5.3145 |
|           | STD   | 0.5519 | 0.0957 | 0.5748 | 9.97E-03 | 0.3661 |
|           | RANK  | 3 | 5 | 2 | 1 | 4 |

Mean rank 3.3913 3.7391 2.3478 2.6956 1.6087

Ranking 5 4 2 3 1

Table 3: Continued.
Figure 1: Continued.
Figure 1: The convergence curve of partial test functions.
Table 4: FMSSA and modified SSA rank-sum test results.

| P-value | SSA vs FMSSA | ISSA vs FMSSA | ESSA vs FMSSA | MSSA vs FMSSA |
|---------|--------------|---------------|---------------|---------------|
| F1      | 6.78E-08     | 6.80E-08      | 6.80E-08      | 6.80E-08      |
| F2      | 6.80E-08     | 6.80E-08      | 6.80E-08      | 6.80E-08      |
| F3      | 6.80E-08     | 6.80E-08      | 6.80E-08      | 6.80E-08      |
| F4      | 6.80E-08     | 6.80E-08      | 6.80E-08      | 6.80E-08      |
| F5      | 1.05E-06     | 6.80E-08      | 6.80E-08      | 6.80E-08      |
| F6      | 6.80E-08     | 6.80E-08      | 6.80E-08      | 0.005115262   |
| F7      | 6.80E-08     | 0.86340974    | 0.903116496   | 6.80E-08      |
| F8      | 0.228694354  | 6.80E-08      | 3.94E-07      | 0.735268153   |
| F9      | 8.01E-09     | NaN           | NaN           | 7.99E-09      |
| F10     | 8.01E-09     | 8.01E-09      | 0.342112253   | 8.01E-09      |
| F11     | 8.01E-09     | 0.019781508   | NaN           | 8.01E-09      |
| F12     | 6.80E-08     | 6.80E-08      | 6.80E-08      | 6.80E-08      |
| F13     | 0.000130693  | 0.592858508   | 2.94E-06      |               |
| F14     | 1.60E-08     | 6.35E-08      | 8.89E-08      | 0.043657508   |
| F15     | 0.000274329  | 0.0060397959  | 0.002559421   | 0.000835245   |
| F16     | NaN          | 8.01E-09      | 3.49E-07      | NaN           |
| F17     | NaN          | 8.01E-09      | 8.01E-09      | NaN           |
| F18     | 3.98E-02     | 8.30E-04      | 8.30E-04      | 0.393874081   |
| F19     | NaN          | 8.01E-09      | 8.01E-09      | NaN           |
| F20     | 0.015834725  | 3.48E-08      | 1.33E-05      | 0.375297687   |
| F21     | 9.51E-04     | 3.53E-07      | 3.53E-07      | 0.515801039   |
| F22     | 0.005671208  | 4.78E-06      | 7.12E-06      | 0.397076      |
| F23     | 0.441669394  | 7.13E-06      | 4.36E-07      | 0.954583801   |
| +/=/-   | 2/5/16       | 1/2/20        | 4/5/14        | 1/10/12       |

Table 5: Compare the results of FMSSA and other algorithms.

| Functions | Index | SCA     | WOA     | MFO     | MVO     | MPA     | FMSSA  |
|-----------|-------|---------|---------|---------|---------|---------|--------|
|           | Ave   | 1.08E+01| 2.33E-76| 2.00E+03| 0.3465  | 5.74E-23| 1.69E-109|
| F1        | STD   | 1.2513  | 6.09E-85| 3.23E+02| 0.0169  | 3.11E-33| 2.11E-110|
| RANK      | 5     | 2       | 6       | 4       | 3       | 1       |
|           | Ave   | 0.0323  | 2.16E-50| 3.11E+01| 0.4396  | 1.04E-15| 6.35E-58 |
| F2        | STD   | 0.0043  | 9.09E-52| 7.97    | 0.0159  | 3.14E-17| 4.85E-59 |
| RANK      | 4     | 2       | 6       | 5       | 3       | 1       |
|           | Ave   | 8.28E+03| 4.55E+04| 2.26E+04| 2.82E+01| 2.79E-05| 1.18E-110|
| F3        | STD   | 5.32E+02| 9.99E+02| 1.77E+03| 1.1902  | 6.32E-06| 4.24E-115|
| RANK      | 4     | 6       | 5       | 3       | 2       | 1       |
|           | Ave   | 3.55E+01| 5.17E+01| 6.85E+01| 0.7381  | 3.18E-09| 1.95E-96 |
| F4        | STD   | 3.7109  | 2.108   | 0.7105  | 0.0836  | 9.51E-13| 1.85E-57 |
| RANK      | 4     | 5       | 6       | 3       | 2       | 1       |
|           | Ave   | 4.71E+04| 2.80E+01| 4.00E+02| 3.68E+02| 2.80E+01| 2.80E+01 |
| F5        | STD   | 7.02E+02| 3.93E-01| 1.79E+01| 1.74E+01| 2.725   | 3.71E-01 |
| RANK      | 6     | 1       | 5       | 4       | 3       | 1       |
|           | Ave   | 1.54E+01| 3.99E-01| 2.39E+03| 0.411   | 2.46E-08| 6.44E-12 |
| F6        | STD   | 1.6047  | 1.94E-02| 2.35E+02| 0.0132  | 4.87E-10| 1.77E-13 |
| RANK      | 5     | 3       | 6       | 4       | 2       | 1       |
|           | Ave   | 0.1001  | 5.18E-02| 4.5024  | 1.46E-02| 1.39E-03| 1.43E-05 |
| F7        | STD   | 0.0055  | 2.27E-04| 0.2812  | 7.09E-04| 3.72E-05| 1.46E-07 |
| RANK      | 5     | 4       | 6       | 3       | 2       | 1       |
|           | Ave   | -3.87E+03|-1.03E+04|-8.63E+03|-7.43E+03|-8.29E+03|-6.57E+03|
| F8        | STD   | 3.31E+01| 2.07E+02| 7.14E+02| 1.21E+02| 1.51E+02| 5.29E+02|
| RANK      | 6     | 1       | 2       | 4       | 3       | 5       |
|           | Ave   | 5.16E+01| 0       | 1.59E+02| 6.34E+02| 0       | 0       |
| F9        | STD   | 3.7878  | 0       | 1.77E+01| 2.45E+01| 0       | 0       |
| RANK      | 4     | 1       | 5       | 6       | 1       | 1       |
|           | Ave   | 1.65E+01| 4.44E-15| 1.61E+01| 1.0016  | 1.24E-12| 8.88E-16 |
| F10       | STD   | 2.0339  | 2.83E-18| 0.7368  | 0.0398  | 3.34E-14| 0       |
| RANK      | 6     | 2       | 5       | 4       | 3       | 1       |
The results of FMSSA, WOA, SCA, MFO, MVO, and MPA runs on the test functions are given in Table 5. From the data in the table, it can be seen that FMSSA obtains the best case in all of the unimodal functions except F8, F12, F14, F18, F21, F22, and F23, while it brings the best chance in four of the six multimodal functions and four of the nine fixed-dimension functions. This illustration shows that FMSSA demonstrates dominance in solving unimodal and multimodal functions, obtaining the best case in almost all of these two types of functions. This is also due to the enhancement of the SSA algorithm by the two improvement strategies. The only drawback is that only four functions demonstrate the best case in the face of fixed-dimension functions, and even in the F18 test functions, all algorithms except FMSSA exhibit the best solution. This result indicates that FMSSA still has some shortcomings when facing fixed-dimension functions. The main reason is that the very fast convergence rate leads to the tendency to fall into local solutions, and the optimal solution cannot be found. The last part of Table 5 provides the results of the Friedman rank-sum test from which it can be seen that FMSSA ranks first with 1.47. Figure 2 shows the convergence curve of FMSSA on some test functions. It can also be seen that FMSSA has the fastest convergence speed and the highest calculation accuracy in most test number functions. From the figure, it can be seen that FMSSA converges the quickest and the optimal solution all the time without the problem of iterative stopping, thanks to the improvement of the exploration ability by the parameter adjustment strategy. FMSSA has good
Figure 2: Continued.
Figure 2: The convergence curve of partial test functions.
Table 6: FMSSA and other algorithms rank-sum test results.

| P-value | SCA vs FMSSA | WOA vs FMSSA | MFO vs FMSSA | MVO vs FMSSA | MPA vs FMSSA |
|---------|--------------|--------------|--------------|--------------|--------------|
| F1      | 6.80E-08     | 6.80E-08     | 6.80E-08     | 6.80E-08     | 6.80E-08     |
| F2      | 6.80E-08     | 1.20E-06     | 6.80E-08     | 6.80E-08     | 6.80E-08     |
| F3      | 6.80E-08     | 6.80E-08     | 6.80E-08     | 6.80E-08     | 6.80E-08     |
| F4      | 6.80E-08     | 1.33E-07     | 6.80E-08     | 6.80E-08     | 6.80E-08     |
| F5      | 6.80E-08     | 0.00920913   | 6.80E-08     | 6.80E-08     | 6.80E-08     |
| F6      | 6.80E-08     | 6.80E-08     | 6.80E-08     | 6.80E-08     | 6.79E-08     |
| F7      | 6.80E-08     | 5.23E-07     | 1.81E-05     | 0.006557193  | 1.43E-07     |
| F8      | 8.01E-09     | NaN          | 8.01E-09     | 8.01E-09     | NaN          |
| F9      | 8.01E-09     | 1.91E-07     | 8.01E-09     | 8.01E-09     | 8.01E-09     |
| F10     | 8.01E-09     | 0.34211253   | 8.01E-09     | 8.01E-09     | NaN          |
| F11     | 6.80E-08     | 6.80E-08     | 6.80E-08     | 1.81E-05     | NaN          |
| F12     | 4.49E-08     | 0.27871029   | 4.49E-08     | 0.27871029   | 2.27E-05     |
| F13     | 3.96E-08     | 4.39E-05     | 7.68E-08     | 3.99E-09     | 3.99E-09     |
| F14     | 0.00621631   | 0.001480216  | 0.000411615  | 0.0003046    | 7.99E-09     |
| F15     | 8.01E-09     | 0.162568851  | NaN          | 8.01E-09     | NaN          |
| F16     | 8.01E-09     | 8.01E-09     | NaN          | 8.01E-09     | NaN          |
| F17     | 8.01E-09     | 8.01E-09     | NaN          | 8.01E-09     | NaN          |
| F18     | 0.000830361  | 0.000830361  | 0.03981673   | 0.000830361  | 0.03981673   |
| F19     | 8.01E-09     | 8.01E-09     | NaN          | 8.01E-09     | NaN          |
| F20     | 3.48E-08     | 0.000308717  | 0.018233007  | 0.010548668  | 0.009584368  |
| F21     | 4.03E-08     | 0.000370199  | 0.268838065  | 0.045489262  | 4.57E-09     |
| F22     | 1.53E-05     | 0.008340797  | 0.135207931  | 0.132056163  | 2.41E-08     |
| F23     | 0.001040475  | 0.767564353  | 0.457049626  | 0.0335664    | 6.33E-09     |

Figure 3: Box plot of partial test functions. (a) F1. (b) F3. (c) F6. (d) F13.
convergence in F10 and F13 functions in the case of multimodal functions. In the face of fixed-dimension functions, it shows a mediocre level. Table 6 provides the results of the rank-sum test for FMSSA and the other algorithms. From the results, it can be found that FMSSA outperforms the other algorithms.

Figure 3 shows the box diagram of each algorithm on some test functions (F1, F3, F6, and F13). The ordinate corresponding to the upper and lower sides of the blue rectangle in the figure represents the upper quartile and the lower quartile of the solution, and the ordinate corresponding to the red straight line in the rectangle is the median. The black lines at the top and bottom are the upper and lower bounds of the value range, and the red plus sign indicates abnormal values.

A comparison of FMSSA with some improved algorithms (CSOAOA [25], LHIO [37], LICRSA [38], G-QPSO [39]) and the latest proposed algorithms (HBA [40], WSO [41], GO [42]) is given in Table 7. The optimal mean and standard deviation are given in bold. From the table, it can be found that FMSSA ranks first with an average rank of 3.13. The results show that FMSSA has good optimization results compared to the advanced optimization algorithms when facing multimodal functions. In contrast, both have good convergence results when facing unimodal functions. Both converge to the optimal solution, but the optimization effect of WSO is inferior in the face of unimodal functions. Both converge to the optimal solution, but the optimization of WSO is very poor in the face of unimodal functions. This result also illustrates the weakness of FMSSA in the face of fixed-dimension problems. The rank-sum tests of FMSSA and the improved and proposed algorithms are given in Table 8. From the table, we can see that CSOAOA, LHIO, LICRSA, G-QPSO, HBA, WSO, and GO are 6/3/14, 10/7/6, 7/11/5, 3/3/17, 8/10/5, 5/4/14, and 5/4/14, respectively.

3.6. In-Depth Analysis of Algorithmic Complexity. Considering that the control parameter adjustment strategy and the Morlet wavelet variation strategy are modifications of the foraging activity and the pilot position of the original SSA algorithm, no more time complexity is added. Since the complexity of the objective function part, where \( m \) stands for the number of objectives.

From the algorithm complexity in the table, we can find that SSA has the same complexity as FMSSA because FMSSA does not add complexity to SSA. In addition, FMSSA has the same complexity as SCA, WOA, and MPA and is also the smallest, and all other improved SSA algorithms increase the complexity on top of SSA.

4. Shape Optimization of Generalized Developable Bézier-Like Surfaces

4.1. Optimization Criteria. First, suppose a generalized developable Bézier-like surface \( \{ \Pi_i \} \), then the dual curve is

\[
r_s(t; \lambda_1, \lambda_2, \lambda_3) = \sum_{i=0}^{3} F_i, 0 \leq t \leq 1,
\]

provided by some algorithm literature does not consider the complexity of the objective function, and for fairness principle and uniformity, we add the complexity of the objective function part, where \( m \) stands for the number of objectives.

From the algorithm complexity in the table, we can find that SSA has the same complexity as FMSSA because FMSSA does not add complexity to SSA. In addition, FMSSA has the same complexity as SCA, WOA, and MPA and is also the smallest, and all other improved SSA algorithms increase the complexity on top of SSA.

\[
r_s(t; \lambda_1, \lambda_2, \lambda_3) = \lambda_1 a(t) \Delta Q_0 + \lambda_2 b(t) \Delta Q_1 + \lambda_3 c(t) \Delta Q_2 + S(t),
\]

where

\[
\Delta Q_0 = Q_1 - Q_0, \Delta Q_1 = Q_2 - Q_1, \Delta Q_2 = Q_3 - Q_2,
\]

\[
a(t) = t(1 - t)^2, b(t) = t^2(1 - t)^2, c(t) = -t^3(1 - t),
\]

\[
S(t) = (1 - t)^3 Q_0 + 3t(1 - t)^2 Q_1 - 3t^2(1 - t) Q_2 + t^3 Q_3.
\]

The following three optimization criteria of generalized Bézier-like curves are given.

4.1.1. The Arc Length of Dual Curve is the Shortest. Given a generalized Bézier-like curve, its arc length can be calculated by the following formula:

\[
R(\lambda_1, \lambda_2, \lambda_3) = \int_0^1 \| r_s'(t; \lambda_1, \lambda_2, \lambda_3) \|^2 \, dt.
\]

Substituting formula (22) into formula (24), we get

\[
R(\lambda_1, \lambda_2, \lambda_3) = \int_0^1 \left( \| \lambda_1 a'(t) \Delta Q_0 + \lambda_2 b'(t) \Delta Q_1 + \lambda_3 c'(t) \Delta Q_2 + S'(t) \|^2 \right) \, dt
\]

\[
= \int_0^1 \left( \lambda_1^2 \| a'(t) \|^2 + \lambda_2^2 \| b'(t) \|^2 + \lambda_3^2 \| c'(t) \|^2 \right) + 2\lambda_1 \lambda_2 (a'(t) \Delta Q_0) \cdot (b'(t) \Delta Q_1) + 2\lambda_1 \lambda_3 (a'(t) \Delta Q_0) \cdot (c'(t) \Delta Q_2) + 2\lambda_2 \lambda_3 (b'(t) \Delta Q_1) \cdot (c'(t) \Delta Q_2) + 2\lambda_1 \lambda_2 (a'(t) \Delta Q_0) \cdot (S'(t) + 2\lambda_2 b'(t) \Delta Q_1 + 2\lambda_3 c'(t) \Delta Q_2) - \lambda_2 \lambda_3 (S'(t) + \| S'(t) \|^2) \, dt.
\]
| Functions | Index | CSOAOA | LHIO | LICRSA | G-QPSO | HBA | WSO | GJO | FMSSA |
|-----------|-------|--------|-------|--------|--------|-----|-----|-----|-------|
| F1        | Ave   | 1.37E−13 | 7.55E−147 | 0 | 1.58E−74 | 2.48E−135 | 270.4733 | 2.96E−54 | 1.69E−109 |
|           | STD   | 4.26E−13 | 2.59E−146 | 0 | 6.47E−74 | 9.82E−135 | 148.6705 | 8.91E−54 | 2.11E−110 |
|           | RANK  | 7       | 2     | 1     | 5     | 3   | 8   | 6   | 4     |
| F2        | Ave   | 4.39E−10 | 4.78E−74 | 0 | 6.48E−42 | 4.79E−72 | 4.0138 | 2.32E−60 | 6.35E−58 |
|           | STD   | 4.22E−10 | 2.14E−73 | 0 | 6.74E−42 | 1.63E−71 | 1.2953 | 5.22E−60 | 4.85E−59 |
|           | RANK  | 7       | 2     | 1     | 6     | 3   | 8   | 4   | 5     |
| F3        | Ave   | 2.22E−04 | 3.03E−83 | 0 | 8.66E−24 | 1.89E−97 | 1195.8201 | 3.54E−52 | 1.18E−110 |
|           | STD   | 1.42E−04 | 1.35E−82 | 0 | 3.07E−23 | 6.45E−97 | 507.3181 | 1.58E−51 | 4.24E−115 |
|           | RANK  | 7       | 4     | 1     | 6     | 3   | 8   | 5   | 2     |
| F4        | Ave   | 0.0015  | 1.29E−74 | 0 | 2.52E−29 | 1.80E−56 | 13.3528 | 6.79E−41 | 1.95E−56 |
|           | STD   | 0.0007  | 4.63E−74 | 0 | 8.95E−29 | 6.79E−56 | 2.3898 | 1.54E−40 | 1.85E−57 |
|           | RANK  | 7       | 2     | 1     | 6     | 3   | 8   | 5   | 4     |

Table 7: Comparison of FMSSA with the latest and improved algorithms.
Table 7: Continued.

| Functions | Index | CSOAOA | LHHO | LICRSA | G-QPSO | HBA | WSO | GJO | FMSSA |
|-----------|-------|--------|------|--------|--------|-----|-----|-----|-------|
| F20       | Ave   | -3.2563| -3.1848| -3.2863| -2.9929| -3.2469| -3.2923| -3.1389| -3.3220 |
|           | STD   | 6.06E–02 | 0.0684 | 0.0559 | 0.0128 | 0.0967 | 5.28E–02 | 0.1418 | 0.0112 |
|           | RANK  | 4      | 6     | 3     | 8     | 5   | 2   | 7   | 1     |
| F21       | Ave   | -10.1532 | -8.8763 | -5.6350 | -4.4333 | -9.6896 | -9.4062 | -9.3762 | -8.3904 |
|           | STD   | 3.43E–15 | 2.26E+00 | 3.2015 | 0.2511 | 2.0731 | 2.30E+00 | 1.8625 | 2.9169 |
|           | RANK  | 1      | 5     | 7     | 8     | 5   | 2   | 3   | 4     |
| F22       | Ave   | -10.4029 | -9.0688 | -6.3161 | -4.4434 | -10.0211 | -10.0690 | -9.8601 | -8.6761 |
|           | STD   | 1.57E–07 | 2.36E+00 | 3.5080 | 0.2179 | 1.7077 | 1.49E+00 | 1.6256 | 2.8058 |
|           | RANK  | 1      | 5     | 7     | 8     | 5   | 2   | 3   | 4     |
| F23       | Ave   | -10.5364 | -9.7221 | -6.4979 | -4.5633 | -9.3899 | -10.1531 | -9.8472 | -5.3145 |
|           | STD   | 4.92E–08 | 1.98E+00 | 3.8627 | 0.2539 | 2.81E+00 | 1.71E+00 | 2.1199 | 0.3661 |
|           | RANK  | 1      | 4     | 6     | 8     | 5   | 2   | 3   | 7     |

Table 8: FMSSA and the latest and improved algorithms rank-sum test results.

| P-value | CSOAOA vs FMSSA | LHHO vs FMSSA | LICRSA vs FMSSA | G-QPSO vs FMSSA | HBA vs FMSSA | WSO vs FMSSA | GJO vs FMSSA |
|---------|-----------------|---------------|-----------------|-----------------|--------------|--------------|--------------|
| F1      | 6.80E–08        | 6.80E–08      | +               | 8.01E–09        | +            | 6.80E–08     | +            | 6.80E–08     | -            |
| F2      | 6.80E–08        | 6.80E–08      | +               | 8.01E–09        | +            | 6.80E–08     | +            | 6.80E–08     | -            |
| F3      | 6.80E–08        | 7.41E–05      | -               | 8.01E–09        | +            | 6.80E–08     | -            | 1.92E–07     | -            |
| F4      | 6.80E–08        | 6.80E–08      | +               | 8.01E–09        | +            | 6.80E–08     | -            | 0.27329      | -            |
| F5      | 0.01667         | 6.80E–08      | +               | 6.80E–08        | +            | 6.80E–08     | +            | 6.80E–08     | -            |
| F7      | 0.00214         | 0.56085       | =               | 0.07643         | =            | 0.03605      | =            | 0.00037      | =            |
| F8      | 6.80E–08        | 6.79E–08      | +               | 9.17E–08        | -            | 6.80E–08     | -            | 0.00022      | +            |
| F9      | 6.54E–09        | NaN            | NaN             | NaN             | NaN          | NaN          | =            | 8.01E–09     | NaN          |
| F10     | 8.01E–09        | NaN            | NaN             | NaN             | 8.64E–08     | 0.34211      | =            | 8.01E–09     | 1.77E–08     |
| F11     | 8.01E–09        | NaN            | NaN             | NaN             | 0.34211      | NaN          | =            | 8.01E–09     | 0.34211      |
| F12     | 7.41E–05        | 0.02944        | +               | 6.80E–08        | +            | 6.80E–08     | -            | 2.56E–07     | -            |
| F13     | 4.14E–05        | 1.57E–05      | -               | 4.49E–08        | +            | 0.27876      | =            | 0.27876      | 4.49E–08     |
| F14     | 0.66772         | 3.99E–09      | +               | 2.43E–06        | +            | 7.39E–08     | +            | 1.58E–08     | +            |
| F15     | 0.00110         | 0.02563        | +               | 0.00856         | +            | 0.01143      | +            | 0.65425      | 7.99E–09     |
| F16     | NaN             | NaN            | NaN             | 8.01E–09        | -            | NaN          | =            | 0.08063      | 7.99E–09     |
| F17     | NaN             | 0.34211        | =               | 8.01E–09        | -            | NaN          | =            | 0.08063      | 8.01E–09     |
| F18     | 0.03982         | 0.52625        | =               | 0.13756         | =            | 0.00083      | =            | 0.03982      | 0.00083      |
| F19     | 0.00017         | 0.01376        | =               | 0.034211        | =            | 0.81E–09     | =            | 0.34211      | =            |
| F20     | 0.02796         | 0.00025        | =               | 0.06708         | =            | 3.48E–08     | =            | 0.26917      | 0.55990      |
| F21     | 4.57E–09        | 4.68E–05      | +               | 0.96437         | =            | 3.53E–07     | =            | 2.08E–07     | +            |
| F22     | 1.96E–07        | 0.00226        | +               | 0.96577         | =            | 4.35E–07     | =            | 8.03E–07     | +            |
| F23     | 1.02E–07        | 4.21E–05      | +               | 0.37449         | =            | 7.13E–06     | =            | 7.08E–05     | +            |

Table 9: Time complexity of different algorithms.

| Alg. | Ref. | Algorithmic complexity |
|------|------|------------------------|
| SSA  | [30] | O (SSA) = O (L × (dim × N × Cof × N)) |
| ISSA | [31] | O (ISSA) = O (L × (dim + 2 × dim × N + 2 × Cof × N)) |
| ESSA | [43] | O (ESSA) = O (L × (dim × N + m × N² + Cof × N)) |
| MIP  | [44] | O (MIP) = O (L × (N² + L + dim × N + L × Cof × N)) |
| SCA  | [45] | O (SCA) = O (L × (dim × N + Cof × N)) |
| WOA  | [46] | O (WOA) = O (L × (dim × N + Cof × N)) |
| MFO  | [47] | O (MFO) = O (L × (N² + L × dim × N + L × Cof × N)) |
| MVO  | [48] | O (MVO) = O (L × (N² + dim × logN × Cof × N)) |
| MPA  | [49] | O (MPA) = O (L × (dim × N + Cof × N)) |
| FMSSA|      | O (FMSSA) = O (L × (dim × N + Cof × N)) |
It can be obtained by integrating the items on the right side of the equal sign in the above formula (25) one by one

\[
\begin{align*}
A_0 &= \int_0^1 \left( a'(t) \| \Delta Q_0 \| \right)^2 dt = \int_0^1 \left( a'(t) \right)^2 dt \cdot \| \Delta Q_0 \|^2 = \frac{3}{5} \| \Delta Q_0 \|^2, \\
A_1 &= \int_0^1 \left( b'(t) \| \Delta Q_1 \| \right)^2 dt = \int_0^1 \left( b'(t) \right)^2 dt \cdot \| \Delta Q_1 \|^2 = \frac{2}{105} \| \Delta Q_1 \|^2, \\
A_2 &= \int_0^1 \left( c'(t) \| \Delta Q_2 \| \right)^2 dt = \int_0^1 \left( c'(t) \right)^2 dt \cdot \| \Delta Q_2 \|^2 = \frac{3}{35} \| \Delta Q_2 \|^2, \\
A_3 &= \int_0^1 \left( d'(t) \Delta Q_3 \right) \cdot \left( b'(t) \Delta Q_1 \right) dt = \int_0^1 d'(t) \cdot b'(t) dt \cdot \Delta Q_3 \cdot \Delta Q_1 = \frac{1}{70} \Delta Q_3 \cdot \Delta Q_1, \\
A_4 &= \int_0^1 \left( a'(t) \Delta Q_0 \right) \cdot \left( c'(t) \Delta Q_2 \right) dt = \int_0^1 a'(t) \cdot c'(t) dt \cdot \Delta Q_0 \cdot \Delta Q_2 = \frac{1}{70} \Delta Q_0 \cdot \Delta Q_2, \\
A_5 &= \int_0^1 \left( b'(t) \Delta Q_1 \right) \cdot \left( c'(t) \Delta Q_2 \right) dt = \int_0^1 b'(t) \cdot c'(t) dt \cdot \Delta Q_1 \cdot \Delta Q_2 = \frac{1}{70} \Delta Q_1 \cdot \Delta Q_2, \\
A_6 &= \int_0^1 a'(t) \Delta Q_0 \cdot \cdot \cdot (t) dt = \int_0^1 a'(t) \Delta Q_0 \cdot \cdot \cdot (t) dt \cdot \Delta Q_0 = \left( \frac{1}{5} Q_0 + \frac{3}{10} Q_1 - \frac{1}{10} Q_2 \right) \cdot \Delta Q_0, \\
A_7 &= \int_0^1 b'(t) \Delta Q_1 \cdot \cdot \cdot (t) dt = \int_0^1 b'(t) \Delta Q_1 \cdot \cdot \cdot (t) dt \cdot \Delta Q_1 = \left( \frac{1}{10} Q_0 - \frac{3}{10} Q_2 + \frac{1}{5} Q_3 \right) \cdot \Delta Q_1, \\
A_8 &= \int_0^1 c'(t) \Delta Q_2 \cdot \cdot \cdot (t) dt = \int_0^1 c'(t) \Delta Q_2 \cdot \cdot \cdot (t) dt \cdot \Delta Q_2 = \left( \frac{1}{5} Q_0 - \frac{3}{5} Q_2 - \frac{6}{5} Q_3 \right) \cdot \Delta Q_2, \\
A_9 &= \int_0^1 \| S'(t) \|^2 dt = \frac{9}{5} \| Q_0 \|^2 + \frac{6}{5} \| Q_1 \|^2 - \frac{6}{5} \| Q_2 \|^2 + \frac{9}{5} \| Q_1 \|^2 - \frac{9}{5} \| Q_2 \| \cdot \| Q_3 \| - \frac{9}{5} \| Q_2 \| \cdot \| Q_3 \| - \frac{9}{5} \| Q_2 \| \cdot \| Q_3 \|. 
\end{align*}
\]

Equation (25) can be simplified as follows:

\begin{equation}
R(\lambda_1, \lambda_2, \lambda_3) = \lambda_1^2 A_0 + \lambda_2^2 A_1 + \lambda_3^2 A_2 + 2 \lambda_1 \lambda_2 A_3 + 2 \lambda_1 \lambda_3 A_4 + 2 \lambda_2 \lambda_3 A_5 + 2 \lambda_1 A_6 + 2 \lambda_2 A_7 + 2 \lambda_3 A_8 + A_9. 
\end{equation}

(27)

4.1.2. The Dual Curve has the Smallest Energy. The energy value function of parameter curve is

\[
E(\lambda_1, \lambda_2, \lambda_3) = \int_0^1 \| r'(t; \lambda_1, \lambda_2, \lambda_3) \|^2 dt.
\]

(28)

Substituting formula (22) into formula (28), we get

\[
E(\lambda_1, \lambda_2, \lambda_3) = \int_0^1 \left( \| a''(t) \| \Delta Q_0 + \lambda_2 b''(t) \Delta Q_1 + \lambda_3 c''(t) \Delta Q_2 + S''(t) \right)^2 dt \\
= \int_0^1 \left( \lambda_1^2 (a''(t) \| \Delta Q_0 \|)^2 + \lambda_2^2 (b''(t) \| \Delta Q_1 \|)^2 + \lambda_3^2 (c''(t) \| \Delta Q_2 \|)^2 + 2 \lambda_1 \lambda_2 (a''(t) \Delta Q_0) \cdot (b''(t) \Delta Q_1) \\
+ 2 \lambda_1 \lambda_3 (a''(t) \Delta Q_0) \cdot (c''(t) \Delta Q_2) \\
+ 2 \lambda_2 \lambda_3 (b''(t) \Delta Q_1) \cdot (c''(t) \Delta Q_2) + 2 \lambda_1 a''(t) \Delta Q_0 \cdot S''(t) + 2 \lambda_2 b''(t) \Delta Q_1 \cdot S''(t) \\
+ 2 \lambda_3 c''(t) \Delta Q_2 \cdot S''(t) + \| S''(t) \|^2 \right) dt.
\]

(29)

Integrating the items on the right side of formula (29) one by one, we get
Combined with formula (30), formula (29) can be simplified as follows:

\[
E(\lambda_1, \lambda_2, \lambda_3) = \lambda_1^2 B_0 + \lambda_2^2 B_1 + \lambda_3^2 B_2 + 2\lambda_1\lambda_2 B_3 + 2\lambda_1\lambda_3 B_4 + 2\lambda_2\lambda_3 B_5 + 2\lambda_1 B_6 + 2\lambda_2 B_7 + 2\lambda_3 B_8 + B_6.
\]

\[
(31)
\]

4.1.3. The Curvature Change Rate of Dual Curve is the Smallest. The curvature change rate of parametric curve can be measured by the following formula:

\[
T(\lambda_1, \lambda_2, \lambda_3) = \int_0^1 \left( \lambda_1 a''(t)\Delta Q_0 + \lambda_2 b''(t)\Delta Q_1 + \lambda_3 c''(t)\Delta Q_2 + S''(t) \right)^2 dt.
\]

\[
(32)
\]

Substituting formula (22) into formula (32), we get
Successive integration of the items on the right can be expressed as the following equation:

\[
\begin{align*}
C_0 &= \int_0^1 \left( a^{\prime\prime}(t) \right) \left\| \Delta \mathbf{Q}_0 \right\| dt = \int_0^1 \left( a^{\prime\prime}(t) \right) \left\| \Delta \mathbf{Q}_0 \right\|^2 dt = 84 \cdot \left\| \Delta \mathbf{Q}_0 \right\|^2, \\
C_1 &= \int_0^1 \left( b^{\prime\prime}(t) \right) \left\| \Delta \mathbf{Q}_0 \right\| dt = \int_0^1 \left( b^{\prime\prime}(t) \right) \left\| \Delta \mathbf{Q}_0 \right\|^2 dt = 48 \cdot \left\| \Delta \mathbf{Q}_0 \right\|^2, \\
C_2 &= \int_0^1 \left( c^{\prime\prime}(t) \right) \left\| \Delta \mathbf{Q}_0 \right\| dt = \int_0^1 \left( c^{\prime\prime}(t) \right) \left\| \Delta \mathbf{Q}_0 \right\|^2 dt = 84 \cdot \left\| \Delta \mathbf{Q}_0 \right\|^2, \\
C_3 &= \int_0^1 \left( a^{\prime\prime}(t) \right) \left\| b^{\prime\prime}(t) \right\| \left\| \Delta \mathbf{Q}_0 \right\| dt = \int_0^1 a^{\prime\prime}(t) \cdot b^{\prime\prime}(t) \cdot \Delta \mathbf{Q}_0 \cdot \Delta \mathbf{Q}_1 = -48 \cdot \Delta \mathbf{Q}_0 \cdot \Delta \mathbf{Q}_1, \\
C_4 &= \int_0^1 \left( a^{\prime\prime}(t) \right) \left\| c^{\prime\prime}(t) \right\| \left\| \Delta \mathbf{Q}_0 \right\| dt = \int_0^1 a^{\prime\prime}(t) \cdot c^{\prime\prime}(t) \cdot \Delta \mathbf{Q}_0 \cdot \Delta \mathbf{Q}_2 = -12 \cdot \Delta \mathbf{Q}_0 \cdot \Delta \mathbf{Q}_2, \\
C_5 &= \int_0^1 \left( b^{\prime\prime}(t) \right) \left\| c^{\prime\prime}(t) \right\| \left\| \Delta \mathbf{Q}_0 \right\| dt = \int_0^1 b^{\prime\prime}(t) \cdot c^{\prime\prime}(t) \cdot \Delta \mathbf{Q}_1 \cdot \Delta \mathbf{Q}_2 = 48 \cdot \Delta \mathbf{Q}_1 \cdot \Delta \mathbf{Q}_2, \\
C_6 &= \int_0^1 a^{\prime\prime}(t) \left\| \Delta \mathbf{Q}_0 \right\| \cdot \Delta S^{\prime}(t) dt = \int_0^1 a^{\prime\prime}(t) \cdot \Delta S^{\prime}(t) \cdot \Delta \mathbf{Q}_0 = 36 \cdot (-Q_0 + 3Q_1 - 3Q_2 + Q_3) \cdot \Delta \mathbf{Q}_0, \\
C_7 &= \int_0^1 b^{\prime\prime}(t) \left\| \Delta \mathbf{Q}_1 \right\| \cdot \Delta S^{\prime}(t) dt = \int_0^1 b^{\prime\prime}(t) \cdot \Delta S^{\prime}(t) \cdot \Delta \mathbf{Q}_1 = 0, \\
C_8 &= \int_0^1 c^{\prime\prime}(t) \left\| \Delta \mathbf{Q}_2 \right\| \cdot \Delta S^{\prime}(t) dt = \int_0^1 c^{\prime\prime}(t) \cdot \Delta S^{\prime}(t) \cdot \Delta \mathbf{Q}_2 = 36 \cdot (-Q_0 + 3Q_1 - 3Q_2 + Q_3) \cdot \Delta \mathbf{Q}_2, \\
C_9 &= \int_0^1 \left\| \Delta S^{\prime\prime}(t) \right\|^2 dt = 36 \left\| \mathbf{Q}_0 \right\|^2 + 324 \left| \mathbf{Q}_1 \right|^2 + 324 \left| \mathbf{Q}_2 \right|^2 + 36 \left| \mathbf{Q}_3 \right|^2 - 216Q_0 \cdot Q_1 + 216Q_0 \cdot Q_2 - 72Q_0 \cdot Q_3 - 648Q_1 \cdot Q_2 + 216Q_1 \cdot Q_3 - 216Q_2 \cdot Q_3, \\
\end{align*}
\]

Using formula (34), formula (33) can be simplified as

\[
T(\lambda_1, \lambda_2, \lambda_3) = \lambda_1^2 C_0 + \lambda_2^2 C_1 + \lambda_3^2 C_2 + 2\lambda_1\lambda_2 C_3 + 2\lambda_1\lambda_3 C_4 + 2\lambda_2\lambda_3 C_5 + 2\lambda_1\lambda_6 C_6 + 2\lambda_2\lambda_7 C_7 + 2\lambda_3\lambda_8 C_8 + C_9. 
\]

(35)

\[
M_1: \begin{cases}
\min & R(\lambda_1, \lambda_2, \lambda_3) = \lambda_1^2 A_0 + \lambda_2^2 A_1 + \lambda_3^2 A_2 + 2\lambda_1\lambda_2 A_3 + 2\lambda_1\lambda_3 A_4 + 2\lambda_2\lambda_3 A_5 + 2\lambda_1 A_6 + 2\lambda_2 A_7 + 2\lambda_3 A_8 + A_9, \\
\text{s.t.} & \lambda_1, \lambda_3 \in [-3, 1], \lambda_2 \in [-3, 3].
\end{cases}
\]

(36)

(2) Establishing an optimization model according to the optimal criterion formula (31) of minimum energy of dual curve, we get

\[
M_2: \begin{cases}
\min & E(\lambda_1, \lambda_2, \lambda_3) = \lambda_1^2 B_0 + \lambda_2^2 B_1 + \lambda_3^2 B_2 + 2\lambda_1\lambda_2 B_3 + 2\lambda_1\lambda_3 B_4 + 2\lambda_2\lambda_3 B_5 + 2\lambda_1 B_6 + 2\lambda_2 B_7 + 2\lambda_3 B_8 + B_9, \\
\text{s.t.} & \lambda_1, \lambda_3 \in [-3, 1], \lambda_2 \in [-3, 3].
\end{cases}
\]

(37)

4.2. Optimization Model
(3) Establishing an optimization model based on the optimization criterion formula (35) with the minimum curvature change rate of the dual curve, we get

\[
M_3: \begin{cases} 
\min & T(\lambda_1, \lambda_2, \lambda_3) = \lambda_1^3C_0 + \lambda_2^3C_1 + \lambda_3^3C_2 + 2\lambda_1\lambda_2C_3 + 2\lambda_1\lambda_3C_4 + 2\lambda_2\lambda_3C_5 + 2\lambda_1C_6 + 2\lambda_2C_7 + 2\lambda_3C_8 + C_9, \\
\text{s.t.} & \lambda_1, \lambda_3 \in [-3, 1], \lambda_2 \in [-3, 3].
\end{cases}
\] (38)

Equations (36) to (38) are optimization models based on the shortest arc length, the smallest energy, and the smallest curvature change rate of the dual curve of the generalized developable Bézier-like surface, respectively. The concrete numerical examples are given in the next section.

\[
\begin{align*}
Q_{0,1} &= (5\sqrt{2}, -5\sqrt{2}, 5, 125), \\
Q_{2,1} &= (-5\sqrt{2}, 5\sqrt{2}, 5, 125), \\
Q_{1,1} &= (-5\sqrt{2}, -5\sqrt{2}, 5, 125), \\
Q_{3,1} &= (5\sqrt{2}, 5\sqrt{2}, 5, 125).
\end{align*}
\] (39)

Figures 4–6 show the optimization illustrations when the control plane is given as formula (39). In which (a) is the optimized generalized envelope developable Bézier-like surface, and (b) is the corresponding objective function value convergence graph. In Table 10, the optimal shape parameters and the corresponding minimum objective function values obtained by solving three optimization models with different intelligent algorithms are given. The data in Tables 10 and 11 below have the same meanings and will not be explained again. Table 12.

Example 1. The control plane coordinates of a set of generalized enveloping developable Bézier-like surface are given as follows:

\[
\begin{align*}
Q_{0,2} &= (0, -20 \sin 60^\circ, 20 \cos 60^\circ, 400), \\
Q_{2,2} &= (0, 20 \cos 60^\circ, 20 \sin 60^\circ, 400), \\
Q_{1,2} &= (0, -20 \cos 60^\circ, 20 \sin 60^\circ, 400), \\
Q_{3,2} &= (0, 20 \sin 60^\circ, 20 \cos 60^\circ, 400).
\end{align*}
\] (40)

Example 2. Given a control plane coordinate of another group of generalized enveloping developable Bézier-like surface:

\[
\begin{align*}
Q_{0,3} &= (2\sqrt{3}, -3\sqrt{3}, 4\sqrt{3}, 5\sqrt{3}), \\
Q_{2,3} &= (2\sqrt{3}, 3\sqrt{3}, 4\sqrt{3}, 5\sqrt{3}), \\
Q_{1,3} &= (6, 9, -2, -15), \\
Q_{3,3} &= (50, 135, 240, 375).
\end{align*}
\] (41)

The subgraphs in Figure 7–9 have the same meaning as those in Figures 4–6. Here, two sets of envelope developable surfaces are given to verify the generality of the optimization method. Given below is an example of shape optimization of the generalized spine curve developable Bézier-like surface.

Example 3. Given is a group of generalized spine curve developable Bézier-like surface control planes:

In Figures 10–12, (a) is based on the optimization models $M_1, M_2, M_3$ when the control plane is formula (41). The optimized generalized spine curve developable Bézier-like surface is obtained, and Fig. (b) is the convergence curve of the objective function of these three optimization models.
Figure 4: The shape optimization of generalized enveloping developable Bézier-like surface based on the optimization model $(M)_1$ in Example 1. (a) The optimized developable surface. (b) The arc length of the dual curve.

Figure 5: The shape optimization of generalized enveloping developable Bézier-like surface based on the optimization model $(M)_2$ in Example 1. (a) The optimized developable surface. (b) The energy value of the dual curve.

Figure 6: The shape optimization of generalized enveloping developable Bézier-like surface based on the optimization model $(M)_3$ in Example 1. (a) The optimized developable surface. (b) The curvature variation of the dual curve.
Table 10: Optimization results in Example 1.

| Algorithm | $\lambda_1$ | $\lambda_2$ | $\lambda_3$ | Optimization model | Objective function value |
|-----------|-------------|-------------|-------------|--------------------|------------------------|
| SCA       | -3.0000     | -3.199e -07 | -3.0000     | $M_1$              | -37422.8571             |
|           | -1.7024     | -0.12397    | -1.6444     | $M_2$              | 5604.5423               |
|           | -1.4165e-09 | -7.1822e-10| -7.7124e-09| $M_3$              | 28809.0358              |
| WOA       | -3.0000     | -2.5069e -07| -3.0000     | $M_1$              | -37422.8571             |
|           | -1.6667     | -0.001474   | -1.6666     | $M_2$              | 5600.0000               |
|           | -1.4942e-09 | 1.002e -08  | 1.6452e -09 | $M_3$              | 28800.0000              |
| MFO       | -3.0000     | 3.1428e -07 | -3.0000     | $M_1$              | -37423.4396             |
|           | -1.6667     | 6.5647e -09 | -1.6666     | $M_2$              | 5600.0000               |
|           | -4.7566e -09| 9.5784e -09 | 5.6876e -09 | $M_3$              | 28800.0000              |
| MVO       | -3.0000     | 3.1765e -06 | -3.0000     | $M_1$              | -37422.8571             |
|           | -1.6666     | -1.7066e-04 | -1.6666     | $M_2$              | 56000.0028              |
|           | -3.6833-04  | -1.4438-04  | 4.4849e-05  | $M_3$              | 28800.0024              |
| SSA       | -3.0000     | 1.401e -08  | -3.0000     | $M_1$              | -37444.3675             |
|           | -1.6667     | 3.0326e -08 | -1.6667     | $M_2$              | 5600.0000               |
|           | -5.3237e-08 | 2.044e -08  | -4.5363e-08 | $M_3$              | 28802.1563              |
| FMSSA     | -3.0000     | -3.2109e -07| -3.0000     | $M_1$              | -37422.8571             |
|           | -1.6667     | 2.8354e -09 | -1.6667     | $M_2$              | 5600.0000               |
|           | 6.6325e -09 | 8.9055e -09 | 5.4171e -10 | $M_3$              | 28800.0000              |
Table 11: Optimization results in Example 2.

| Algorithm | $\lambda_1$ | $\lambda_2$ | $\lambda_3$ | Optimization model | Objective function value |
|-----------|-------------|-------------|-------------|---------------------|--------------------------|
| SCA       | $-0.73802$  | $7.4365e-03$| $-0.74496$  | $M_1$               | $-383577.127$            |
|           | $0.45971$   | $8.8832e-03$| $0.45963$   | $M_2$               | $3643.1074$              |
|           | $0.74568$   | $1.7061e-03$| $0.75077$   | $M_3$               | $13229.7157$             |
| WOA       | $-0.74013$  | $1.7456e-03$| $-0.73915$  | $M_1$               | $-383577.1281$           |
|           | $0.46095$   | $0.015132$  | $0.74764$   | $M_2$               | $3643.1391$              |
|           | $0.73695$   | $-0.0039$   | $0.45385$   | $M_3$               | $13229.2551$             |
| MFO       | $-0.73964$  | $2.0882e-07$| $-0.73964$  | $M_1$               | $-383577.1281$           |
|           | $0.45753$   | $1.0268e-08$| $0.45753$   | $M_2$               | $3658.1547$              |
|           | $0.74231$   | $-2.7356e-09$| $0.74231$  | $M_3$               | $13229.0344$             |
| MVO       | $-0.73978$  | $-1.611e-05$| $-0.73977$  | $M_1$               | $-383577.1281$           |
|           | $0.45730$   | $-4.3935e-04$| $0.45746$  | $M_2$               | $3643.0781$              |
|           | $0.74232$   | $-1.2972e-04$| $0.74271$  | $M_3$               | $13229.0355$             |
| SSA       | $-0.73964$  | $1.519e-06$  | $-0.73964$  | $M_1$               | $-383577.1281$           |
|           | $0.45753$   | $1.2823e-08$| $0.45753$   | $M_2$               | $3643.0781$              |
|           | $0.74234$   | $2.6578e-09$ | $0.74231$  | $M_3$               | $13229.0344$             |
| FMSSA     | $-0.73964$  | $-1.2839e-06$| $-0.73964$  | $M_1$               | $-383577.1281$           |
|           | $0.45753$   | $3.8258e-09$ | $0.45753$   | $M_2$               | $3643.0781$              |
|           | $0.74231$   | $2.0434e-09$ | $0.74231$  | $M_3$               | $13229.0344$             |
Table 12: Optimization results in Example 3.

| Algorithm | $\lambda_1$ | $\lambda_2$ | $\lambda_3$ | Optimization model | Objective function value |
|-----------|-------------|-------------|-------------|---------------------|--------------------------|
| SCA       | 0.89266     | 1.0000      | -0.68645    | $M_1$               | -32114.7894              |
|           | 0.66749     | 1.0000      | -0.5064     | $M_2$               | 12916.2623               |
|           | -0.059772   | -5.8243e-04 | -0.037139   | $M_3$               | 67204.5337               |
| WOA       | 0.86537     | 1.0000      | -0.68072    | $M_1$               | -32115.2576              |
|           | 0.67187     | 1.0000      | -0.51976    | $M_2$               | 12901.3285               |
|           | 5.4609e-02  | 8.4218e-04  | -4.0912e-02 | $M_3$               | 67208.8402               |
| MFO       | 0.86536     | 1.0000      | -0.68072    | $M_1$               | -32115.2576              |
|           | 0.67188     | 1.0000      | -0.51977    | $M_2$               | 12903.4478               |
|           | 6.3486e-02  | 5.3667e-03  | -0.035349   | $M_3$               | 67203.6321               |
| MVO       | 0.86509     | 1.0000      | -0.68058    | $M_1$               | -32115.0007              |
|           | 0.67235     | 1.0000      | -0.51996    | $M_2$               | 12901.3373               |
|           | 6.2921e-02  | 4.6365e-03  | 3.5674e-02  | $M_3$               | 67203.6418               |
| SSA       | 0.86536     | 1.0000      | -0.68072    | $M_1$               | -32110.2633              |
|           | 0.67188     | 1.0000      | -0.51977    | $M_2$               | 12918.3285               |
|           | -0.063208   | 4.9172e-03  | -3.552e-02  | $M_3$               | 67212.3395               |
| FMSSA     | 0.86536     | 1.0000      | -0.68072    | $M_1$               | -32115.2576              |
|           | 0.67188     | 1.0000      | -0.51977    | $M_2$               | 12901.3285               |
|           | 6.3208e-02  | 4.9175e-04  | -3.552e-02  | $M_3$               | 67203.6265               |
Comprehensive observation of Figures 4(b)–12(b) shows that when the iteration time is less than 20 times, the objective function curve corresponding to FMSSA tends to be minimum and stable, which indicates that FMSSA is superior to solving the shape optimization problem of this kind of developable surface, and this conclusion can be further verified according to the data in Tables 10–12.

Figure 7: The shape optimization of generalized enveloping developable Bézier-like surface based on optimization model (M)_1 in Example 2. (a) The optimized developable surface. (b) The arc length of the dual curve.

Figure 8: The shape optimization of generalized enveloping developable Bézier-like surface based on optimization model (M)_2 in Example 2. (a) The optimized developable surface. (b) The energy value of the dual curve.
Figure 9: The shape optimization of generalized enveloping developable Bézier-like surface based on the optimization model (M)_{3} in Example 2. (a) The optimized developable surface. (b) The curvature variation of the dual curve.

Figure 10: The shape optimization of generalized spine curve developable Bézier-like surface based on the optimization model (M)_{1} in Example 3. (a) The optimized developable surface. (b) The arc length of the dual curve.
5. Conclusions

This paper mainly studies the single-objective shape optimization of generalized developable Bézier-like surfaces. The main research contents can be divided into two parts: (1) The control parameter adjustment strategy and Morlet wavelet mutation strategy are added to the salp swarm algorithm, and an improved salp swarm algorithm (FMSSA) is proposed. The improved algorithm improves the exploration-exploitation performance and convergence accuracy of the original SSA, but it is easy to fall into local solutions when dealing with fixed-dimension problems. (2) According to the dual principle of point and plane in 3D projective space, the single-objective shape optimization models of three generalized developable Bézier-like surfaces are established, and FMSSA is used to solve the established models. Given are several groups of shape optimization examples of generalized developable Bézier-like surfaces. The experimental results verify the effectiveness and accuracy of FMSSA in solving the single-objective shape optimization problem of developable surfaces.

6. Future Work

In future work, first, given the excellent performance of FMSSA in solving the shape optimization problem generalized, the proposed method can be extended to other surface modeling examples [50–52]. Second, given the algorithm deficiencies mentioned in the paper, we will further improve its engineering optimization capabilities by mixing...
and incorporating different improvement strategies and addressing the poverty that the algorithm tends to fall into local solutions. Alternatively, the proposed improved algorithm can be further considered to solve more complex and demanding multiobjective optimization of shape optimization problems of generalized. Finally, because of the breadth and superiority of the salp swarm algorithm in solving problems in various fields such as engineering applications, FMSSA can be applied to research in more widely used and technically demanding areas such as feature selection [53], path planning, and predictive modeling.

Data Availability

All data generated or analyzed during this study are included in this published article.

Conflicts of Interest

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

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. 51875454).

References

[1] G. Weiss and P. Furtner, “Computer-aided treatment of developable surfaces,” Computers & Graphics, vol. 12, no. 1, pp. 39–51, 1988.
[2] G. Aumann, “Interpolation with developable Bézier patches,” Computer Aided Geometric Design, vol. 8, no. 5, pp. 409–420, 1991.
[3] Z. L. Ye, Y. Q. Meng, and K. X. Liu, “Developable Bézier surface,” J. Num. Method. Comp. Appl., vol. 18, no. 2, pp. 81–86, 1997.
[4] C. Y. Li, R. H. Wang, and C. G. Zhu, “An approach for designing a developable surface through a given line of curvature,” Computer-Aided Design, vol. 45, no. 3, pp. 621–627, 2013.
[5] X. W. Zhang and G. J. Wang, “A new algorithm for designing developable Bézier surfaces,” Journal of Zhejiang University - Science, vol. 7, no. 12, pp. 2050–2056, 2006.
[6] L. Fernández-Jambrina, “B-spline control nets for developable surfaces,” Computer Aided Geometric Design, vol. 24, no. 4, pp. 189–199, 2007.
[7] C. H. Chu, C. C. Wang, and C. R. Tsai, “Computer aided geometric design of strip using developable Bézier patches,” Computers in Industry, vol. 59, no. 6, pp. 601–611, 2008.
[8] H. D. Hwang and S. H. Yoon, “Constructing developable surfaces by wrapping cones and cylinders,” Computer-Aided Design, vol. 58, pp. 230–235, 2015.
[9] G. Hu, J. Wu, and X. Wang, “Constructing local controlled developable H-Bézier surfaces by interpolating characteristic curves,” Computational and Applied Mathematics, vol. 40, no. 6, p. 216, 2021.
[10] R. Bodduluri and B. Ravani, “Design of developable surfaces using duality between plane and point geometries,” Computer-Aided Design, vol. 25, no. 10, pp. 621–632, 1993.
[11] H. Pottmann and G. Farin, “Developable rational Bézier and B-spline surfaces,” Computer Aided Geometric Design, vol. 12, no. 5, pp. 513–531, 1995.
[12] H. Pottmann and J. Wallner, “Approximation algorithms for developable surfaces,” Computer Aided Geometric Design, vol. 16, no. 6, pp. 539–556, 1999.
[13] M. Zhou, G. Peng, and Z. Ye, “Design of developable surfaces by using duality between plane and point geometries,” Journal of Computer Aided Design and Computer Graphics, vol. 16, no. 10, pp. 1401–1406, 2004.
[14] M. Zhou, G. Peng, and Z. Ye, “Geometric design of understandably developable surfaces,” China Mechanical Engineering, vol. 17, no. 24, pp. 2554–2557, 2006.
[15] M. Zhou, J. Q. Yang, H. C. Zheng, and W. Song, “Design and shape adjustment of developable surfaces,” Applied Mathematical Modelling, vol. 37, no. 6, pp. 3789–3801, 2013.
[16] G. Hu, H. X. Cao, S. X. Zhang, and G. Wei, “Developable Bézier-like surfaces with multiple shape parameters and its continuity conditions,” Applied Mathematical Modelling, vol. 45, pp. 728–747, 2017.
[17] G.-G. Wang, S. Deb, and Z. Cui, “Monarch butterfly optimization,” Neural Computing & Applications, vol. 31, no. 7, pp. 1995–2014, 2019.
[18] S. Li, H. Chen, M. Wang, and S. HeidariMirjalilì, “Slime mould algorithm: a new method for stochastic optimization,” Future Generation Computer Systems, vol. 111, pp. 300–323, 2020.
[19] G.-G. Wang, “Moth search algorithm: a bio-inspired meta-heuristic algorithm for global optimization problems,” Memetic Computing, vol. 10, no. 2, pp. 151–164, 2018.
[20] Y. Yang, H. Chen, A. A. Heidari, and A. H. Gandomi, “Hunger games search: visions, conception, implementation, deep analysis, perspectives, and towards performance shifts,” Expert Systems with Applications, vol. 177, Article ID 114864, 2021.
[21] I. Ahmadianfar, A. A. Heidari, A. H. Gandomi, and H. ChuChen, “RUN beyond the metaphor: an efficient optimization algorithm based on Runge Kutta method,” Expert Systems with Applications, vol. 181, Article ID 110579, 2021.
[22] J. Tu, H. Chen, M. Wang, and A. H. Gandomi, “The colony predation algorithm,” Journal of Bionics Engineering, vol. 18, no. 3, pp. 674–710, 2021.
[23] I. Ahmadianfar, A. A. Heidari, S. Noshadian, and A. H. ChenGandomi, “INFO: an efficient optimization algorithm based on weighted mean of vectors,” Expert Systems with Applications, vol. 195, Article ID 116516, 2022.
[24] A. A. Heidari, S. Mirjalili, H. Faris, and H. AljarahMafarjaChen, “Harris hawks optimization: algorithm and applications,” Future Generation Computer Systems, vol. 97, pp. 849–872, 2019.
[25] G. Hu, J. Zhong, B. Du, and G. Wei, “An enhanced hybrid arithmetic optimization algorithm for engineering applications,” Computer Methods in Applied Mechanics and Engineering, vol. 394, Article ID 114901, 2022.
[26] A. Faramarzi, M. Heidarinejad, B. Stephens, and S. Mirjalili, “Equilibrium optimizer: a novel optimization algorithm,” Knowledge-Based Systems, vol. 191, Article ID 105190, 2020.
[27] H. Eskandar, A. Sadollah, A. Bahreininejad, and M. Hamdi, “Water cycle algorithm - a novel metaheuristic optimization method for solving constrained engineering optimization problems,” Computers & Structures, vol. 110-111, pp. 151–166, 2012.
[28] A. Chakri, R. Khelif, M. Benouaret, and X. S. Yang, “New directional bat algorithm for continuous optimization
problems,” Expert Systems with Applications, vol. 69, pp. 159–175, 2017.
[29] G. Hu, X. Zhu, G. Wei, and C. T. Chang, “An improved marine predators algorithm for shape optimization of developable surface designs,” Engineering Applications of Artificial Intelligence, vol. 105, Article ID 104417, 2021.
[30] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, and S. M. SaremiFarisMirjalili, “Salp Swarm Algorithm: a bio-inspired optimizer for engineering design problems,” Advances in Engineering Software, vol. 114, no. dec, pp. 163–191, 2017.
[31] M. Tubishat, N. Idris, and S. ShuihAbusabirahMiraljali, “Improved Salp Swarm Algorithm based on opposition based learning and novel local search algorithm for feature selection,” Expert Systems with Applications, vol. 145, Article ID 113122, 2020.
[32] V. Chaudhary, H. M. Dubey, M. Pandit, and J. C. Bansal, “Multi-area economic dispatch with stochastic wind power using Salp Swarm Algorithm,” Array, vol. 8, Article ID 100044, 2020.
[33] A. Naderipour, Z. Abdol-Malek, Z. A. Noorden et al., “Carrier wave optimization for multi-level photovoltaic system to improvement of power quality in industrial environments based on Salp swarm algorithm,” Environmental Technology & Innovation, vol. 21, Article ID 101197, 2021.
[34] K. Gholami and M. H. Parvaneh, “A mutated salp swarm algorithm for optimum allocation of active and reactive power sources in radial distribution systems,” Applied Soft Computing, vol. 85, Article ID 105833, 2019.
[35] H. M. Sultan, A. S. Menesy, S. Kamel, A. Selim, and F. Jurado, “Parameter identification of proton exchange membrane fuel cells using an improved salp swarm algorithm,” Energy Conversion and Management, vol. 224, Article ID 113341, 2020.
[36] H. Lu, T. Iseley, J. Matthews, and M. LiaoAzimi, “An ensemble model based on relevance vector machine and multi-objective salp swarm algorithm for predicting burst pressure of corroded pipelines,” Journal of Petroleum Science and Engineering, vol. 203, Article ID 105855, 2021.
[37] M. K. Naik, R. Panda, A. Wunnava, and A. JenaAbraham, “A leader Harris hawks optimization for 2-D Masi entropy-based multilevel image thresholding,” Multimedia Tools and Applications, vol. 80, no. 28-29, pp. 35543–35583, 2021.
[38] L. Huang, Y. Wang, Y. Guo, and G. Hu, “An improved reptile search algorithm based on levy flight and interactive crossover strategy to engineering application,” Mathematics, vol. 10, no. 13, p. 2329, 2022.
[39] L. d. S. Coelho, “Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems,” Expert Systems with Applications, vol. 37, no. 2, pp. 1676–1683, 2010.
[40] F. A. Hashim, E. H. Houssein, K. Hussain, and W. MabroukAl-Atabany, “Honey Badger Algorithm: new metaheuristic algorithm for solving optimization problems,” Mathematics and Computers in Simulation, vol. 192, pp. 84–110, 2022.
[41] M. Braik, A. Hammouri, J. Atwan, and M. A. Al-Betar-Awadallah, “White Shark Optimizer: a novel bio-inspired meta-heuristic algorithm for global optimization problems,” Knowledge-Based Systems, vol. 243, Article ID 108457, 2022.
[42] N. Chopra and M. Mohsin Ansari, “Golden jackal optimization: a novel nature-inspired optimizer for engineering applications,” Expert Systems with Applications, vol. 198, Article ID 116924, 2022.