

Multi-objective shape optimization of developable Bézier-like surfaces using non-dominated sorting genetic algorithm

Jing Lu^{1,2}, XiaoBo Su², Jingyu Zhong³, and Gang Hu^{2,*} 

¹ College of Mathematics and Computer Application, Shangluo University, Shangluo 726000, PR China

² Department of Applied Mathematics, Xi'an University of Technology, Xi'an 710054, PR China

³ Electronic Information and Electrical Engineering College, Shangluo University, Shangluo 726000, PR China

Received: 3 July 2022 / Accepted: 18 August 2023

Abstract. The shape optimization design of the developable surface is an important research topic in CAD/CAM, and it is widely used in engineering manufacturing. In this paper, NSGA-II (the elitist non-dominated sorting genetic algorithm) is used to study the multi-objective shape optimization problem of generalized cubic developable Bézier-like surfaces (GCDBLS, for short) to promote the application of GCDBLS in industrial software and engineering design. Firstly, the shape optimization of developable surfaces is transformed into the shape optimization of dyadic curves based on the point-to-plane duality theory. Secondly, a multi-objective shape parameter optimization model is developed based on three surface optimality criteria (the shortest arc length, the smallest energy, and the smallest curvature change rate of the dual curve). Finally, the results of shape parameter optimization of GCDBLS obtained by NSGA-II are compared with MSSA and MOGOA to verify the feasibility and superiority of NSGA-II in solving multi-objective shape optimization problems for developable surfaces and the flexibility of GCDBLS in the construction of developable surfaces.

Keywords: Developable Bézier-like surfaces / multi-objective shape optimization / the elitist non-dominated sorting genetic algorithm / shape parameter

1 Introduction

Developable surface is a special type of ruled surface in Computer-Aided Design (CAD) and Computer Aided Manufacturing (CAM). Because it can be expanded into a complete plane without stretching and tearing, and it has high application value in industrial manufacturing [1]. Compared with other curved surfaces, this is also its biggest advantage, so scholars have done a lot of research on developable surfaces. Specifically, there are the structural design of developable surfaces [2–16], interpolation fitting [17–19], mesh approximation [20,21], smooth stitching [22–24] and shape optimization [25].

There are two main methods for constructing developable surfaces: one is pointing geometric representation [2–7], and the other is line or surface geometric representation [8–16], which is also called dual representation. The point geometric representation method regards the developable surface as a ruled surface. It adds certain constraints to express it as a tensor product surface in Euclidean space. However, there are two defects in the

developable surface constructed by the point geometric representation method. On the one hand, the calculation of the nonlinear characteristic equation is more complicated in the construction process, and on the other hand, the constructed developable surface cannot give an explicit expression. Therefore, in order to solve the above mentioned two drawbacks, the surface geometry representation is proposed for constructing developable surfaces. The surface geometry representation method is to regard the developable surface as a curve in the 3D projective space for the construction. The developable surface constructed by this method can avoid the complicated calculation of the first method and has an explicit expression.

The surface geometric representation approach to surface construction is widely used. In 1993, Bodduluri et al. [8] used cubic Bernstein and B-spline basis functions to construct developable surfaces for the first time and gave explicit expressions for the constructed developable surfaces. However, the shape of the developable surface constructed by [8–10] is determined by its control plane and cannot be easily modified. Therefore, in order to control the shape flexibly, Zhou et al. [11] used C-Bézier basis functions to construct a type of C-Bézier developable

* e-mail: hugang@xaut.edu.cn

surface with adjustable shape, but the constructed developable surface only contains one shape parameter, which can only be adjusted the global shape of the curved surface, but the shape of the developable surface cannot be adjusted locally, and the shape adjustability is limited. In order to overcome this defect, Hu et al. [14–16] used Bézier-like basis functions and H-Bézier basis functions to construct two types of developable surfaces with three shape parameters, generalized Bézier-like developable surface and generalized H-Bézier developable surface. They construct developable surfaces that can be flexibly modified by changing the size of the shape parameters of the corresponding developable surface given the control surface coordinates. Based on this property, the shape optimization problem of developable surfaces has become an important research topic in CAD and CAM field. However, there is still little research work on the shape optimization of developable surfaces worldwide. The related work in Section 2 introduces the current status of some techniques for optimizing the shape of developable surfaces. However, the related research works only construct single-objective optimization problems based on a certain surface feature (shortest arc length, minimum energy, and minimum curvature rate of change) [25]. Therefore, this study extends the first shape parameter optimization problem for developable surfaces to multi-objective optimization.

Generally speaking, most engineering design problems are multi-objective optimization problems. In the specific optimization process, the sub-objectives are mostly restricted to each other, so there is no way to get a single solution so that multiple sub-objectives can reach the optimum at the same time. The optimal solution set obtained by multi-objective optimization can guarantee that each solution in the solution set can make all sub-objectives as optimal as possible. According to the point-to-planes duality theory in 3D projective space, the shape optimization of a developable surface is transformed into the shape optimization of the corresponding dual curve in the dual space. Based on this conversion process, three optimization criteria (shortest arc length, minimum energy, and minimum curvature rate of change) related to the surface construction characteristics are introduced and subjected to multi-objective optimization [25]. Furthermore, in order to verify the effect of picking shape parameters optimized by different objective functions on the surface shape, three surface-related optimization objective functions are freely combined, resulting in three two-objective optimization models and one three-objective optimization model. However, it is complicated for traditional mathematical methods to solve multi-objective optimization problems. Therefore, this paper will use a multi-objective evolutionary algorithm to solve this problem and verify the effects of different objective functions on shape parameter optimization.

The earliest multi-objective evolutionary algorithm is the Vector Evaluation Genetic Algorithm (VEGA), which was proposed by David Schaffer et al. in [26]. Compared with the traditional weighting method to solve the multi-objective optimization problem, the results obtained by

VEGA have higher population diversity, but because the algorithm has not adopted the Pareto optimal mechanism, it is easy to converge to the local optimal and has certain defects. By 1995, Deb et al. [27] proposed a multi-objective genetic algorithm based on non-dominated sorting (NSGA), which improves the diversity and convergence of the understanding set, but the algorithm is more troublesome because of its high computational complexity, a large amount of calculation, lack of elite strategies, and need to specify shared parameters manually. On the basis of NSGA, Deb et al. [28] introduced the elite strategy, the crowding degree comparison operator and the fast non-dominated sorting method based on grading to overcome those shortcomings and obtained the fast non-dominated sorting genetic algorithm with the elite strategy (NSGA-II). NSGA-II greatly improves the calculation speed and optimization ability of NSGA, and there is no need to specify shared parameters manually. After that, scholars successively proposed Multi-objective Particle Swarm Optimization Algorithm [29] (MOPSO), Evolutionary Multi-objective Optimization Algorithm Based on an Artificial Immune System [30] (MISA), Multi-objective Evolutionary Algorithm Based on Decomposition [31] (MOEA/D), Multi-objective Salp Swarm Algorithm [32] (MSSA), Multi-objective Grasshopper Optimization Algorithm [33] (MOGOA), etc. However, with its powerful optimization capabilities, NSGA-II has been one of the most popular multi-objective evolutionary algorithms so far. This article is going to use NSGA-II to solve the optimization problem of GCBLDS. In addition, the optimization results of GCBLDS obtained by NSGA-II are experimentally compared with other multi-objective optimization algorithms in this study, which is used to verify the superiority of the NSGA-II algorithm in solving the optimization of GCBLDS shape parameters.

The main scientific contributions of this study are as follows:

- Based on the point-to-planes duality theory, the shape optimization of developable surfaces is transformed into the shape optimization of dual curves in the corresponding dual space.
- A multi-objective optimization model based on the shortest arc length, the smallest energy, and the smallest curvature change rate of the dual curve is established.
- Using NSGA-II to solve the multi-objective generalized cubic developable Bézier-like surfaces established with different combinations of objective functions, the optimal shape parameters satisfying the optimization criterion are obtained.

The structure of the article is as follows: In the first section, we give the display expressions of the generalized Bézier-like curve family and surface family and the optimization criteria of the generalized Bézier-like developable surface. And establish four multi-objective optimization models according to the optimization criteria. Section 2 introduces the basic knowledge related to the multi-objective optimization problem. Section 3 explains the ideology and algorithm steps of the NSGA-II algorithm. Section 4 gives a set of control plane coordinates and then

uses NSGA-II to solve the optimization model established in the first section and record the experimental results. At last, the conclusion is outlined in Section 5.

2 Related work

Shape optimization of extensible surfaces constructs shape-optimal extensible surfaces based on certain requirements or objectives. The implementation of this technique relies on the shape parameters that are available at the time of construction of the spreading surface. However, the conventional method of constructing an expandable surface lacks additional shape parameter. In addition, considering the complexity of the optimization process of spreading surfaces constructed based on shape parameters, many studies have introduced metaheuristic algorithms into the optimization of spreading surfaces and achieved effective optimization results. Some studies of metaheuristic algorithms for spreading surface optimization are presented below to describe the current state of the art.

Hu and Wu et al. invoked an adaptive cuckoo search algorithm with the shape optimization of generalized developable H-Bézier surfaces. They developed three shape optimization models for generalized developable H-Bézier surfaces using the duality between points and planes. Then, the adaptive cuckoo search algorithm is applied to solve the solution of the shape optimization model for generalized developable H-Bézier surfaces [25].

An improved marine predator algorithm is proposed and used for the shape optimization problem of the spreading spherical surface. Hu et al. establish a class of novel shape-adjustable generalized cubic Ball basis functions and construct shape optimization models for the envelope and spine curve spreading surfaces with the energy of the minimized Ball surface as the evaluation criterion. Then, they optimize the shape parameters of Ball surfaces by an improved marine predator algorithm based on a quasi-opposition strategy and a differential evolutionary algorithm established [35].

Lu and Su et al. developed an enhanced Salp Swarm algorithm and used it to optimize the shape of the developable Bézier-like surfaces. First, the shape optimization problem of the generalized developable class of Bézier-like surfaces is transformed into the problem of minimizing the arc length, energy, and rate of change of curvature of the dyadic curves based on the duality principle of point and plane. Then the shape optimization model based on three developable surfaces with arc length, energy, and curvature rate of change is developed. In addition, the Morlet wavelet mutation strategy is introduced into the Salp Swarm algorithm to improve its optimization capability and is used to solve the shape optimization of developable surfaces [36].

Hu and Zhu et al. apply the multi-strategy marine predator algorithm to the optimization of approximately developable surfaces. First, Lévy, Brownian motions, and optimal encounter rate strategies are introduced into the marine predator algorithm to improve the diversity of the algorithm's population. Then, a model for designing an approximately developable surface by interpolating two

given spatial curves is developed with the evaluation criterion of the maximum developability degree of the spreading surface. Finally, the proposed algorithm is used to solve the optimized model to obtain the approximate developable surface with the maximum developability degree [37].

An enhanced JS algorithm for shape optimization of combinatorial cubic generalized ball surfaces is proposed. First, a new combinatorial cubic generalized ball surface is constructed, and the shape optimization of the Ball surface is investigated by an enhanced jellyfish search algorithm. In this study, the surface shape optimization models with G^1 and G^2 geometric continuity are developed separately by minimizing the Ball surface energy as the evaluation criterion [38].

A particle swarm optimization algorithm based on GHT-Bézier developable surface shape optimization was proposed by BiBi et al. First, the GHT-Bézier developable surface shape optimization model is established with the developability degree of the straight grain surface as the objective function and the shape parameters as the optimization variables. Then, the PSO technique is used to search for the optimal shape control parameters within the value range of shape parameters and then visualize the developable surface with high accuracy exploitability [39].

3 Preliminaries

3.1 Generalized cubic developable Bézier-like surfaces

Generalized cubic developable Bézier-like surfaces are constructed based on control planes with Bernstein basis functions and contain three shape parameters. By controlling the three shape parameters, the generalized cubic developable Bézier-like surface allows the position and shape of the developable surface to be freely adjusted, thus increasing the flexibility of developable surface design and improving the ability to solve problems in engineering appearance design. However, the introduction of multiple shape parameters also implies whether the selection of parameters is reasonable for different surface design cases. Therefore, based on the above-mentioned advantages of the surfaces and the problems that arise, this study considers a multi-objective optimization of the control parameters of the generalized cubic developable Bézier-like surface to obtain the optimal parameters.

Given control points $\mathbf{P} \in \mathbf{R}^u (u = 2, 3)$, $\mathbf{P} = [\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3]$, the matrix expression of the generalized cubic Bézier-like curves can be written as [14]

$$\{\prod_t\} : \mathbf{L}(t; \Omega) = \mathbf{TMP}^T, 0 \leq t \leq 1, \quad (1)$$

where

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

$\mathbf{T} = [t^4, t^3, t^2, t, 1]$ $\mathbf{\Omega} = \{\Pi\lambda: \lambda_1, \lambda_2, \lambda_3\}, \lambda_1, \lambda_3 \in [3, 1], \lambda_2 \in [3, 3]$ t and λ_i are the family parameter and shape control parameters of $\{\tilde{\Pi}_t\}$, respectively $\mathbf{M}$ is called coefficient matrix.

According to the point-to-planes dual theorem, combining equation (1), the generalized cubic Bézier-like plane family equation can be obtained as [14]

$$\{\tilde{\Pi}_t\} : \mathbf{U}(t; \cdot) = \mathbf{T}\mathbf{M}\mathbf{Q}^T, 0 \leq t \leq 1, \quad (2)$$

in equation (2), $\mathbf{T}$ and $\mathbf{M}$ have the same values as in equation (1), and $\mathbf{Q}_i = (a_i, b_i, c_i, d_i), i = 0, 1, 2, 3$ represents the coordinates of the control plane. Equation (2) can also be expressed as:

$$\begin{aligned} \mathbf{U}(t; \lambda_1, \lambda_2, \lambda_3) &= \sum_{i=0}^3 F_{i,4}(t) \mathbf{Q}_i \\ &= \left\{ \sum_{i=0}^3 F_{i,4}(t) a_i, \sum_{i=0}^3 F_{i,4}(t) b_i, \sum_{i=0}^3 F_{i,4}(t) c_i, \sum_{i=0}^3 F_{i,4}(t) d_i \right\}, \end{aligned} \quad (3)$$

$F_{i,4}(t) (i = 0, 1, 2, 3)$ are the generalized cubic Bézier-like basis functions, its computational formula are as follows:

See equation (4) below

Let

$$\begin{cases} u_0(t) = \sum_{i=0}^3 F_{i,4}(t) a_i, u_1(t) = \sum_{i=0}^3 F_{i,4}(t) b_i, \\ u_2(t) = \sum_{i=0}^3 F_{i,4}(t) c_i, u_3(t) = \sum_{i=0}^3 F_{i,4}(t) d_i. \end{cases} \quad (5)$$

Equation (2) can be written as:

$$\{\tilde{\Pi}_t\} : \mathbf{U}(t; \mathbf{\Omega}) = \{u_0(t), u_1(t), u_2(t), u_3(t)\}. \quad (6)$$

(a) Generalized Bézier-like envelope developable surfaces

According to the definition of developable surface, the envelope surface of the single-parameter family of

plane $\{\tilde{\Pi}_t\}$ is a developable surface, so the generalized Bézier-like envelope developable surface can be expressed as

$$\mathbf{H}(k, t; \lambda_1, \lambda_2, \lambda_3) = k\mathbf{v}(t) + \psi(t), k \in (-\infty, +\infty) \quad (7)$$

where

$$\begin{cases} \mathbf{u}(t) = (u_0(t), u_1(t), u_2(t)), \\ \mathbf{v}(t) = \mathbf{u}(t) \times \mathbf{u}'(t), \\ \tau = u'_3(t)\mathbf{u}(t) - u_3(t)\mathbf{u}'(t), \\ \psi(t) = \frac{\mathbf{v}(t) \times \tau(t)}{\mathbf{v}(t) \cdot \mathbf{v}(t)}, t \in (0, 1). \end{cases} \quad (8)$$

(b) Generalized Bézier-like spine curve developable surfaces

Generalized Bézier-like spine curve developable surfaces can be defined as:

$$\mathbf{S}(v, t; \lambda_1, \lambda_2, \lambda_3) = \rho(t) + v\rho'(t), \quad (9)$$

where

See equation (10) below

$v \in (-\infty, +\infty), t \in [0, 1]$, the value of $\mathbf{u}(t)$ is the same as equation (8).

3.2 Optimization criteria

Based on the shape adjustability of the GCBLDS, under the premise of keeping the control plane of the developable surface unchanged, the shape parameters of the developable surface can be adjusted to make the developable surface have any shape. However, in many practical engineering problems, it is often necessary to make the developable surface meet certain requirements based on specific constraints. At this time, it is necessary to determine the corresponding criteria to determine the value of the shape parameter. For the GCBLDS designed by the duality theory in 3D projective space, if its control plane is regarded as the control vertex of the curve, the optimization of the developable surface can be converted

$$\begin{cases} F_{0,4}(t) = 1 - (\lambda_1 + 3)t + (3\lambda_1 + 3)t^2 - (3\lambda_1 + 1)t^3 + \lambda_1 t^4, \\ F_{1,4}(t) = (\lambda_1 + 3)t - (6 + 3\lambda_1 + \lambda_2)t^2 + (3\lambda_1 + 2\lambda_2 + 3)t^3 - (\lambda_1 + \lambda_2)t^4, \\ F_{2,4}(t) = (3 + \lambda_2)t^2 - (2\lambda_2 - \lambda_3 + 3)t^3 + (\lambda_2 - \lambda_3)t^4, \\ F_{3,4}(t) = (1 - \lambda_3)t^3 + \lambda_3 t^4. \end{cases} \quad (4)$$

$$\rho(t) = \frac{u''_3(t)[\mathbf{u}(t) \times \mathbf{u}'(t)] + u_3(t)[\mathbf{u}'(t) \times \mathbf{u}''(t)] + u_{\eta 3}(t)[\mathbf{u}''(t) \times \mathbf{u}(t)]}{\mathbf{u}(t) \cdot [\mathbf{u}'(t) \times \mathbf{u}''(t)]} \quad (10)$$

to the corresponding dual curve optimization. The following three optimization criteria [25] for curves are given:

(a) The shortest arc length of dual curves

The arc length of the surface has an important effect on the construction of the surface, so the arc length is minimized to obtain a better-constructed surface. The arc length of a generalized cubic Bézier-like curve $\mathbf{L}(t; \boldsymbol{\Omega})$, $\boldsymbol{\Omega} = \{\lambda_1, \lambda_2, \lambda_3\}$ can be calculated based on the principle of simplicity by the following formula:

$$m_1(\lambda) = \int_0^1 \left\| \mathbf{L}'(t; \omega)^2 \right\| dt. \quad (11)$$

(b) Minimum energy of dual curves

The energy value of the parametric surface is used as an approximate measure of the smoothness of the surface. The smaller the energy value, the smoother the curve. The constructed surface should be as smooth as possible for the practical application process. The energy function of the curve is:

$$m_2(\lambda) = \int_0^1 \left\| \mathbf{L}''(t; \omega) \right\|^2 dt. \quad (12)$$

(c) Dual curves hold smallest curvature variation

Since curvature variation has an important effect on the shape of the surface, a better surface is obtained by combining curvature variation with energy minimization. The curvature change rate of the curve can be measured by the following equation (13):

$$m_3(\lambda) = \int_0^1 \left\| \mathbf{L}'''(t; \omega)^2 \right\| dt. \quad (13)$$

Equations (11)–(13) are three criteria for curve optimization.

3.3 Optimization models

All three criteria (shortest arc length, minimum energy, and minimum curvature variation) significantly affect the shape of generalized cubic developable Bézier-like surfaces but focus on different surface configurations. In order to verify the effect of selecting shape parameters optimized by different objective functions on the surface shape and to select a suitable optimization model, three surface-related optimization objective functions are freely combined to form three two-objective optimization models and one three-objective optimization model. In the following, we present these models.

– Establish a two-objective optimization model *K1* based on the shortest arc length and minimum energy of the dual curve:

$$K1 = \begin{cases} \min_{\lambda \in \Omega} M_1(\lambda) = \{m_1(\lambda), m_2(\lambda)\}, \\ s.t., \lambda_1, \lambda_3 \in [-3, 1], \lambda_2 \in [-3, 3]. \end{cases} \quad (14)$$

– A two-objective optimization model *K2* is established by combining the minimum arc length of the dual curve and the minimum rate of curvature change criterion:

$$K2 = \begin{cases} \min_{\lambda \in \Omega} M_2(\lambda) = \{m_1(\lambda), m_3(\lambda)\}, \\ s.t., \lambda_1, \lambda_3 \in [-3, 1], \lambda_2 \in [-3, 3]. \end{cases} \quad (15)$$

– Establish a two-objective optimization model *K3* based on the minimum energy of the dual curve and the minimum rate of curvature change:

$$K3 = \begin{cases} \min_{\lambda \in \Omega} M_3(\lambda) = \{m_2(\lambda), m_3(\lambda)\}, \\ s.t., \lambda_1, \lambda_3 \in [-3, 1], \lambda_2 \in [-3, 3]. \end{cases} \quad (16)$$

– A three-objective optimization model *K4* is established based on the shortest arc length, the smallest energy, and the smallest curvature change rate of the dual curve:

$$K4 = \begin{cases} \min_{\lambda \in \Omega} M_4(\lambda) = \{m_1(\lambda), m_2(\lambda), m_3(\lambda)\}, \\ s.t., \lambda_1, \lambda_3 \in [-3, 1], \lambda_2 \in [-3, 3]. \end{cases} \quad (17)$$

The models *K1*–*K4* established by combining different optimization criteria, which will be solved later.

4 Elitist non-dominated sorting genetic algorithm (NSGA-II)

4.1 Multi-objective optimization problems

For common multi-objective optimization problems, its objective function can be expressed as:

$$\min_{\lambda \in \Omega} M(\lambda) = \{m_1(\lambda), m_2(\lambda) \dots m_n(\lambda)\}, \quad (18)$$

where $M(\lambda)$ is called the objective function, and $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_D)$ are the decision variable of $M(\lambda)$, $\Omega = [L_i, U_i] \subset R_D$ is search space, L_i and U_i are the lower bound and upper bound of the i th dimension decision variable, D is the dimension of the decision variables, n is the number of sub-objective functions.

Definition 1. Pareto domination

For any $\lambda_1, \lambda_2 \in \Omega$, λ_1 is said to dominates λ_2 if and only if $\forall i \in \{1, 2, \dots, n\}$ makes $m_i(\lambda_1) \leq m_i(\lambda_2)$, and $\exists j \in \{1, 2, \dots, n, j \neq i\}$ makes $m_j(\lambda_1) < m_j(\lambda_2)$, which marked as $\lambda_1 < \lambda_2$, λ_1 is called non-dominated solution, and λ_2 is called dominated solution.

Definition 2. Pareto optimal solution

$\lambda^* \in \Omega$ is called pareto optimal solution if and only if $\lambda \in \Omega$ makes $\lambda < \lambda^*$.

Definition 3. Pareto-optimality Set (PS)

The set consisting of all Pareto optimal solutions is called the Pareto-optimality Set, denoted as PS:

$$PS = \{\lambda^* \in \Omega | \exists \lambda \in \Omega, \lambda < \lambda^*\}. \quad (19)$$

Definition 4. Pareto-optimality Front (PF)

Given a multi-objective optimization problem $M(\lambda)$, PF is defined as: $PF = \{ M(\lambda^*) \mid \lambda^* \in PS \}$,
 PF is the image of PS in the target space.

4.2 Algorithm-related concepts

NSGA-II is obtained by adding some improvement strategies on the basis of NSGA. These strategies can not only improve the calculation progress of the algorithm but also speed up the convergence speed of the algorithm. Let's briefly introduce these improvement strategies.

(a) Tournament selection

In the traditional genetic algorithm (GA), the roulette selection method is usually used, while in NSGA-II, the tournament selection method is used in order to select the outstanding individuals faster. The main ideas are: Randomly select m ($m < N$) individuals from a population P_i with a population size of N , select the individual with the best fitness value to enter the next generation population P_{i+1} and repeat this step until the number of individuals in the new population P_{i+1} reaches N to stop the selection.

(b) Fast non-dominated sorting

For the i th individual in the population P , record the number of individuals dominated by the i th individual ($n(i)$) and the set of other individuals dominated by the i th individual ($S(i)$). The steps of the fast non-dominated sorting algorithm are as follows:

Step 1. Let $k = 1$

Step 2. Store all individuals with $n(i) = 0$ in the population P into the set F_k ;

Step3. Find the set $S(i)$ of individuals dominated by the i th individual in the set F_k , iterate through all j individuals in the set $S(i)$, and perform the operation $n(j) = n(j) - 1$, if $n(j) = 0$, store the individual j into the set G ;

Step4. Take F_k as the k th non-dominated individual solution set, and assign the individual non-dominated ranking rank in the current set F_k : $i = r_{rank}$;

Step5. Take the set G as the current population P , $k = k + 1$, and repeat Step2~Step4 until all individuals in the population are graded.

(c) Crowdedness

Crowdedness implies the density of non-dominated solutions in the neighborhood of the solution. All solutions in the set of non-dominated solutions are sorted in ascending order according to the fitness value of each objective function. According to Figure 1, for candidate solution i , its crowding degree i_d is expressed as the average side length of the rectangle formed by neighboring candidate solutions $i-1$ and $i+1$. In addition, the crowdedness of two candidate solutions on the boundary is assumed to be infinite. A larger i_d means more sparsity and better diversity of solutions near candidate solution i . Crowdedness is calculated as follows:

$$i_d = i_d + (f(i+1) - f(i-1)), i = 2, 3, \dots, N-1. \quad (20)$$

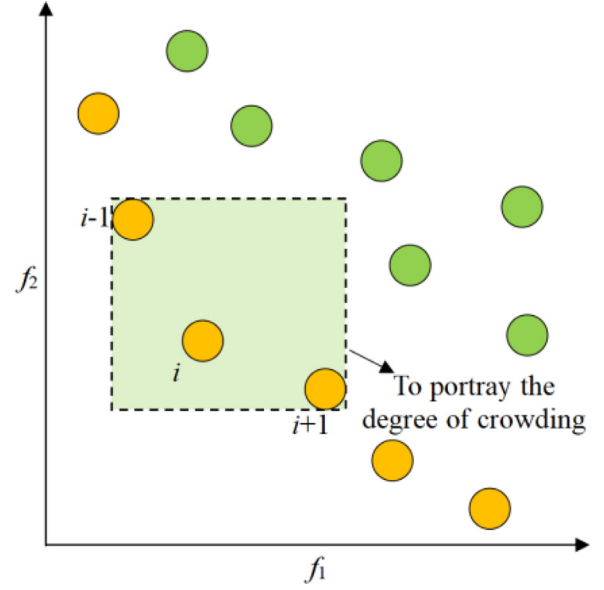


Fig. 1. Crowdedness of solutions with two objectives.

(d) Elite retention strategy

The elite retention strategy is mainly used in the algorithm to generate a new parent population. The main ideas are: first, merge the parent population P_i of the same current generation with the offspring population S_i to obtain a temporary population R_i , and the population size is $2N$. Then, N is selected in population R_i . Two factors are considered for the selection of individuals: non-dominant ranking level and crowding degree. The selection gives preference to the individuals with the top non-dominant ranking level. However, when faced with two individuals with the same ranking and only one slot exists, the individual with a high degree of crowding is selected for the new parental population. Finally, the population is traversed, and N superior individuals are selected as the new parent population P_{i+1} according to the pros and cons relationship between every two individuals.

4.3 Algorithm steps

The basic principles of NSGA-II can be summarized as the following steps:

Step 1. Set the maximum number of iterations of the algorithm to $\max Gen$ and $Gen = 1$. Then randomly generate an initial population P_i with a population size of N , that is, the initial parent population;

Step 2. Perform non-dominated sorting on the individuals in the population P_i , then select individuals through the tournament to perform crossover and mutation operations to obtain the offspring population S_i , and the population size is also N ;

Step 3. $Gen = Gen + 1$;

Step 4. For the current parent population P_i and offspring population S_i , use the elite retention strategy to select the new parent population P_{i+1} ;

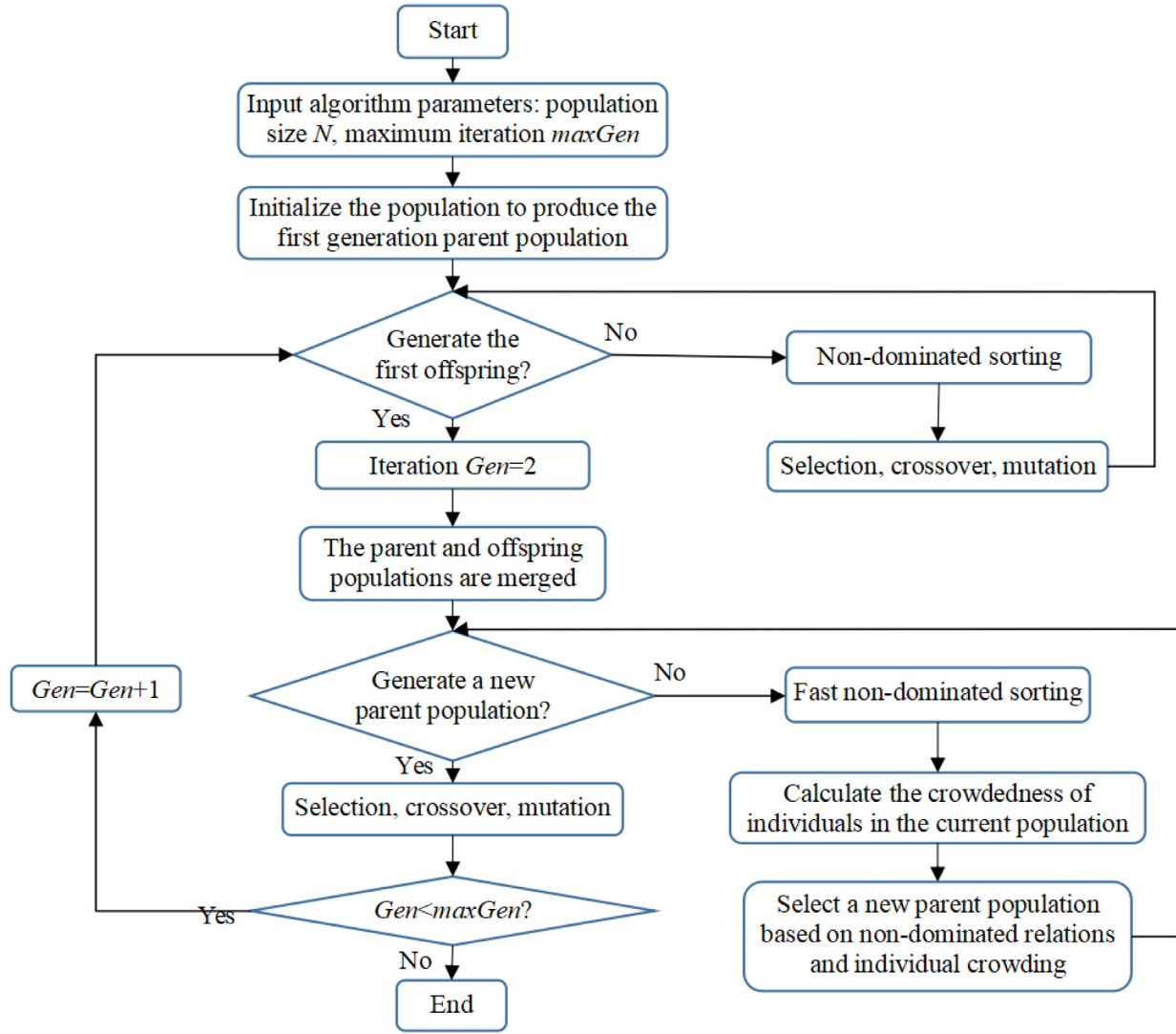


Fig. 2. Flowchart of NSGA-II algorithm.

Step 5. Let $P_i = P_{i+1}$, and perform a fast non-dominant sorting of the individuals in P_i . Then select, cross, and mutate to obtain a new offspring population S_{i+1} ;

Step 6. $S_i = S_{i+1}$ and repeat steps 3 through 5 until $\max Gen = Gen$ and the algorithm ends.

The crossover and mutation operations in the algorithm operation process are the same as the traditional genetic algorithm [34]. In order to express the specific process of the algorithm more clearly, a detailed explanation is given in the flowchart (Fig. 2).

5 Multi-objective numeric experiments for the case of developable surfaces

For the four optimization models K1–K4 proposed in the Section 2, in this section, NSGA-II, MSSA and MOGOA are used to solve the models, and the optimization results obtained are compared. Finally, this paper demonstrates

the effectiveness of NSGA-II in solving the GCBLDS problem by an example. In this experiment, the population size and the number of calculation iterations are both set to 200.

5.1 Evaluation index

- Generation Distance (GD) is used to describe the degree of approximation between the obtained Pareto front and the real Pareto front. The calculation formula is

$$GD = \frac{1}{|P|} \sqrt{\sum_{\lambda \in P} d^2(\lambda, P^*)}, \quad (21)$$

- Inverted Generational Distance (IGD) is a comprehensive performance evaluation index. It is used to evaluate the convergence performance and distribution performance of the algorithm. The formula is expressed as

Table 1. Optimization results of control parameters and objective functions for models $K1$ – $K4$.

Optimization model $K1$ – $K4$	Optimization parameters	MSSA	NSGA-II	MOGOA
$K1 = \begin{cases} \min_{\lambda \in \Omega} M_1(\lambda) = \{m_1(\lambda), m_2(\lambda)\}, \\ s.t. \lambda_1, \lambda_3 \in [-3, 1], \lambda_2 \in [-3, 3]. \end{cases}$	Control parameters	λ_1 0.055877511	0.055877511	0.055877511
		λ_2 -0.009776853	-0.009776853	-0.009776853
		λ_3 0.107055186	0.025448503	0.102001368
	Objective function value	m_1 -383564.7256	-383565.9373	-383564.8042
$K2 = \begin{cases} \min_{\lambda \in \Omega} M_2(\lambda) = \{m_1(\lambda), m_3(\lambda)\}, \\ s.t. \lambda_1, \lambda_3 \in [-3, 1], \lambda_2 \in [-3, 3]. \end{cases}$	Control parameters	λ_1 0.20971211	-0.22576139	0.20971211
		λ_2 0.01549505	-0.16777051	0.01549505
		λ_3 0.11131884	0.11889576	0.31186633
	Objective function value	m_1 -383562.1877	-383567.4744	-383558.6957
$K3 = \begin{cases} \min_{\lambda \in \Omega} M_3(\lambda) = \{m_2(\lambda), m_3(\lambda)\}, \\ s.t. \lambda_1, \lambda_3 \in [-3, 1], \lambda_2 \in [-3, 3]. \end{cases}$	Control parameters	λ_1 0.56055333	0.54620926	0.54620926
		λ_2 0.00917097	0.00917097	0.00917097
		λ_3 0.55613241	0.53995858	0.53272784
	Objective function value	m_1 3653.557347	3650.632477	3650.030595
$K4 = \begin{cases} \min_{\lambda \in \Omega} M_4(\lambda) = \{m_1(\lambda), m_2(\lambda), m_3(\lambda)\}, \\ s.t. \lambda_1, \lambda_3 \in [-3, 1], \lambda_2 \in [-3, 3]. \end{cases}$	Control parameters	m_2 13839.55037	13944.68548	13970.56956
		λ_1 0.19115202	0.12094328	0.19115202
		λ_2 -0.00918123	0.00846505	-0.00918123
	Objective function value	λ_3 0.14426514	0.14426514	-0.12614374
		m_1 -383561.9926	-383563.1471	-383565.7228
		m_2 3730.19846	3751.918888	3855.558338
		m_3 19191.59211	19929.21851	22796.68964

$$IGD = \frac{1}{|P^*|} \sum_{\lambda^* \in P^*} d(\lambda^*, P), \quad (22)$$

$$\begin{aligned} \mathbf{Q}_2 &= (0, 20 \cos 60^\circ, 20 \sin 60^\circ, 400), \\ \mathbf{Q}_3 &= (0, 20 \sin 60^\circ, 20 \cos 60^\circ, 400). \end{aligned}$$

where P is the solution set on the obtained Pareto front, and λ is any individual in P ; Similarly, P^* is the solution set on the real Pareto optimal front, λ^* is any individual in P^* ; $|P|$ and $|P^*|$ represent the size of solution set P and P^* respectively; $d(\lambda, P^*)$ is the Euclidean distance between individual λ and the closest individual in the solution set P^* , $d(\lambda, P)$ is the Euclidean distance between individual λ and the closest individual in the solution set P ; The smaller the value of GD and IGD , the better the convergence and distribution of the algorithm. When the value is 0, it means that the obtained Pareto frontier is completely consistent with the real Pareto frontier, and the better the solution obtained.

– Computing time $T(s)$: Record the total time from the beginning to the end of the calculation, which can measure the calculation speed of the algorithm.

5.2 Numerical experimental analysis of generalized cubic Bezier-like surface

Given a generalized cubic Bezier-like curve, the control plane coordinates are as follows:

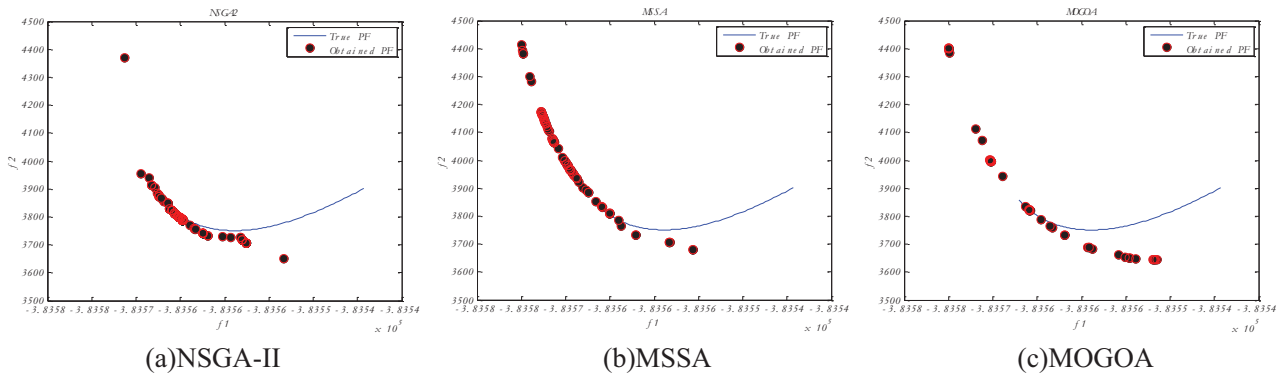
$$\begin{aligned} \mathbf{Q}_0 &= (0, -20 \sin 60^\circ, 20 \cos 60^\circ, 400), \\ \mathbf{Q}_1 &= (0, -20 \cos 60^\circ, 20 \sin 60^\circ, 400), \end{aligned}$$

Table 1 records the optimized control parameters and objective function values of the three algorithms under four optimization models $K1$ – $K4$. Considering that the final optimal solution set obtained from the multi-objective optimization is not unique, the optimization results provided by the three algorithms in the table are all optimal solution sets. The provided optimization control parameters and objective functions are used for the model building of generalized cubic Bessel curves to verify the advanced nature of the optimization results.

The results in the table show that for the optimization results of the $K1$ model, the optimal control parameters λ_1 and λ_2 for MSSA, NSGA-II and MOGOA are 0.055877511 and -0.009776853. only the third control parameter is different. Also, for the results of the optimal objective function, MSSA, NSGA-II and MOGOA are Pareto optimal solutions. For the $K2$ model, the optimal control parameters λ_1 and λ_2 are the same for MSSA and MOGOA, which are 0.055877511 and -0.009776853. the optimal control parameters of NSGA-II are different from all other algorithms, and the differences are large. In addition, for the results of the optimized objective function, MSSA, NSGA-II and MOGOA are all Pareto optimal solutions. For the $K3$ model, the optimal control parameters λ_1 are

Table 2. Comparison results of MSSA, NSGA-II, and MOGOA for models $K1$ – $K4$ in GD , IGD , and $T(s)$.

Algorithms		MSSA			NSGA-II			MOGOA		
Model		GD	IGD	$T(s)$	GD	IGD	$T(s)$	GD	IGD	$T(s)$
$K1$	MIN	1.16	9.65	101.64	0.36	8.73	25.76	1.79	13.01	237.69
	MAX	1.32	11.57	103.65	0.80	20.07	26.71	1.96	14.52	241.05
	AVG	1.23	10.45	102.00	0.59	15.85	26.45	2.15	16.06	239.78
	STD	0.15	0.78	1.03	0.21	4.41	0.85	0.17	1.32	2.33
$K2$	MIN	11.12	86.84	87.79	3.00	62.70	28.55	31.65	242.44	234.72
	MAX	34.18	109.29	91.02	10.40	151.60	29.52	69.98	488.62	251.68
	AVG	17.16	133.57	95.62	5.27	126.77	29.01	40.54	341.78	238.72
	STD	10.69	28.33	3.32	3.78	35.98	1.01	12.48	110.35	10.25
$K3$	MIN	513.84	4900.00	99.86	145.14	4710.00	24.94	511.24	4880.00	246.75
	MAX	580.51	5590.00	100.34	193.49	5410.00	25.21	1132.30	5690.00	291.82
	AVG	535.44	5120.00	102.94	165.04	4940.00	26.54	724.25	5110.00	257.29
	STD	35.98	287.36	0.64	21.87	178.96	0.32	236.36	352.69	15.26
$K4$	MIN	23.65	192.46	110.44	5.13	130.07	34.43	78.86	488.85	243.71
	MAX	35.43	302.26	120.05	6.41	162.09	36.35	111.79	966.08	261.86
	AVG	30.71	229.19	114.82	21.73	146.66	35.10	87.64	569.45	249.22
	STD	5.69	45.21	3.98	6.35	15.38	2.87	10.56	178.69	7.58

**Fig. 3.** Pareto optimal front of model $K1$.

the same for NSGA-II and MOGOA, while the chemical control parameters λ_2 are the same for all three algorithms. In addition, for the results of the optimized objective function, MSSA, NSGA-II, and MOGOA are Pareto optimal solutions. For the $K4$ model, the optimal control parameters λ_1 and λ_2 of MSSA and MOGOA are the same, which are 0.19115202 and -0.00918123 . In addition, the optimal control parameters λ_3 of MSSA and NSGA-II are the same. In addition, for the results of the optimized objective function, MSSA, NSGA-II and MOGOA are Pareto optimal solutions.

The comparative results of the $K1$ – $K4$ models MSSA, NSGA-II, and MOGOA in GD , IGD , and $T(s)$ are given in Table 2. Analysis of the data in Table 2 reveals that NSGA-II obtains the minimum mean, minimum and maximum values of GD for all four optimization models. Also, NSGA-II obtained the mean, minimum and maximum values of the smallest IGD under all four optimization models. Finally, NSGA-II not only obtains the best optimization

results but also its computational speed is much faster than the other two algorithms, which indicates the advantage of the method for solving optimization problems of this type of exploitable surfaces.

Figures 3–6 show the real Pareto front and the obtained Pareto front when the optimization models are $K1$ – $K4$ respectively, the blue curve is real Pareto front and the red is the obtained Pareto front. The higher the agreement between the two curves, the better the calculation result. Combining the data in Table 2, it can be found that when the same number of iterations and population size are set, the optimization results of using NSGA-II to solve the four optimization models are better than MSSA and MOGOA, and NSGA-II is significantly faster in calculation.

Figures 7–10 show the generalized cubic Bezier-like developable surfaces obtained by optimizing the optimization models $K1$ – $K4$ with the optimal shape parameters of a non-dominated solution set obtained by the three algorithms MSSA, NSGA-II and MOGOA. First, it can

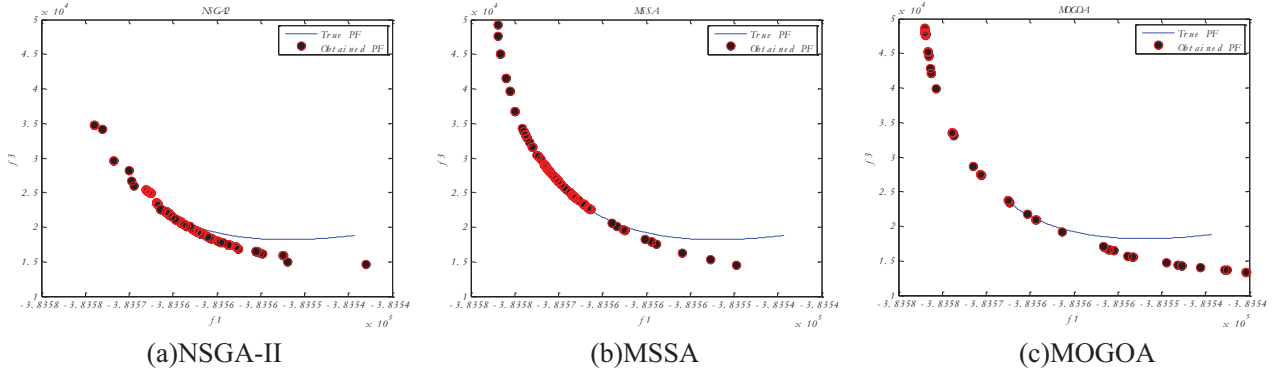


Fig. 4. Pareto optimal front of model K2.

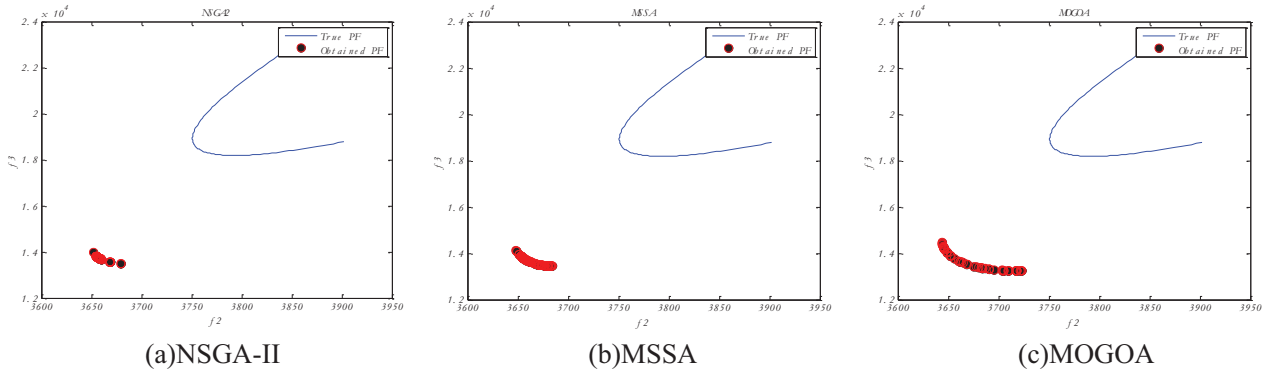


Fig. 5. Pareto optimal front of model K3.

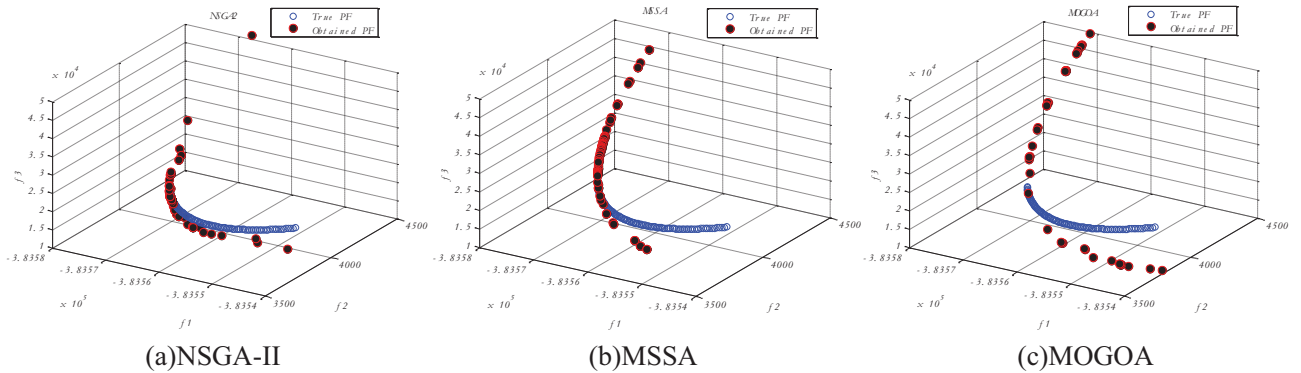


Fig. 6. Pareto optimal front of model K4.

be noticed from Figure 7 that the transitions of the generalized cubic Bessel-like developable surfaces constructed by the NSGA-II optimized algorithm are smoother. In addition, according to the experimental results in Figure 8, the generalized cubic Bessel-like developable surface constructed by the NSGA-II optimization algorithm is steeper on both sides and has a more pronounced slope. This phenomenon is mainly caused by

the large difference between the parameters optimized by NSGA-II in the K2 model and those of MSSA and MOGOA. For the results of model K3 in Figure 9, the generalized cubic Bessel-like developable surfaces constructed by the NSGA-II optimized algorithm are gentler around the top of the slope. In addition, the overall tilt of the generalized cubic Bessel-like developable surface constructed by the algorithm optimized by MSSA is more

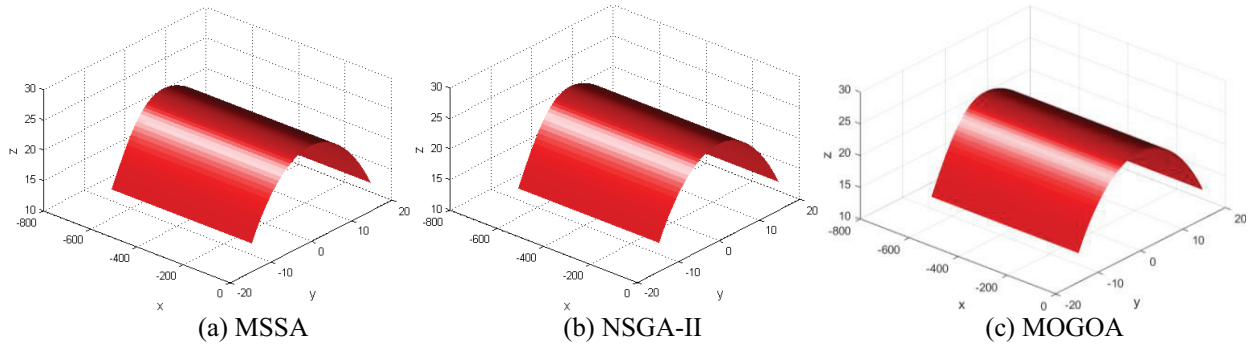


Fig. 7. GCBLDS optimized based on model K1.

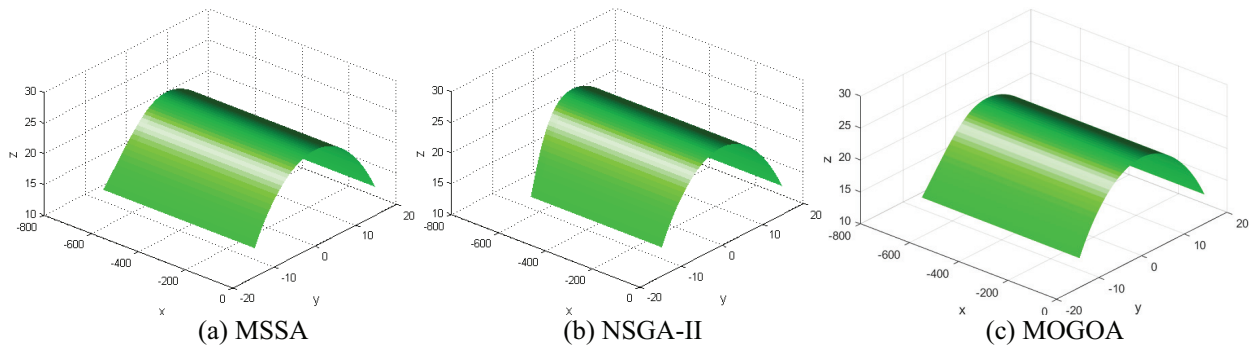


Fig. 8. GCBLDS optimized based on model K2.

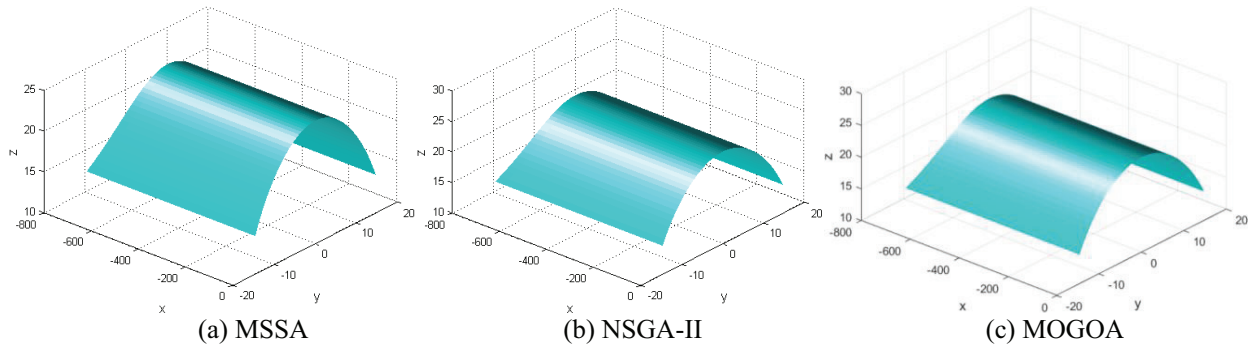


Fig. 9. GCBLDS optimized based on model K3.

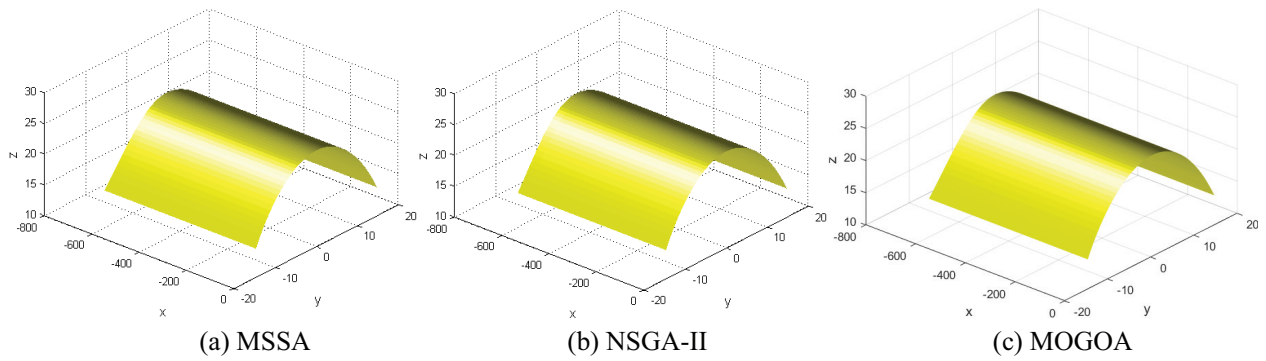


Fig. 10. GCBLDS optimized based on model K4.

pronounced. The main reason for this is that the optimization results of λ_1 and λ_3 are significantly different from the NSGA-II and MOGOA algorithms. Finally, for the results of model K4 in Figure 10, the differences existing in the generalized cubic Bezier-like developable surfaces constructed by the optimization of the three algorithms are insignificant. It is mainly due to the fact that the three objective functions make the optimization process more rational and comprehensive. The optimization results of the three algorithms do not differ significantly in shape.

6 Conclusions

In this paper, NSGA-II is used to optimize the multi-objective shape of GCDBLS. First, according to the dual principle of point and plane in 3D projective space, three optimization criteria for shape optimization of developable surfaces are proposed, and a multi-objective optimization model is established by freely combining two or three of the three optimization criteria. Secondly, it introduces the corresponding basic knowledge in the multi-objective optimization problem and the thought theory of the NSGA-II algorithm. Finally, use NSGA-II to solve the proposed model to obtain the optimal shape parameters. Compared with the other two multi-objective optimization algorithms, it is found that NSGA-II has obvious advantages for solving this type of multi-objective shape optimization problem. However, the use of multiple optimization criteria is experimentally found to increase the time complexity.

For future research directions, attempts can be made to improve NSGA-II by combining other algorithms and introducing effective search operators so as to improve the multi-objective optimization performance of NSGA-II. Finally, many complex single-objective geometric optimization problems are also extended to multi-objective problems, such as shape optimization of the CQGS-Ball surfaces in [40].

Funding

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

Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Data availability statement

All data generated or analyzed during this study are included in this published article (and its supplementary information files).

References

- [1] W.H. Frey, D. Bindschadler, Computer Aided Design of a Class of Developable Bézier Surfaces, General Motors R & D Publication 8057 Springer Verlag, New York, USA, 1993

- [2] W. Gunter, P. Peter, Computer-aided treatment of developable surfaces, *Comput. Graph.* **12**, 39–51 (1988)
- [3] T. Maekawa, J. Chalfant, Design and tessellation of B-spline developable surfaces, *J. Mech. Des.* **120**, 453–461 (1998)
- [4] G. Aumann, A simple algorithm for designing developable Bézier surfaces, *Comput. Aided Geom. Des.* **20**, 601–619 (2003)
- [5] X.W. Zhang, G.J. Wang, A new algorithm for designing developable Bézier surfaces, *J. Zhejiang Univ. Sci. A*, **7**, 2050–2056 (2006)
- [6] C.H. Chu, C. Wang, C.R. Tsai, Computer aided geometric design of strip using developable Bézier patches, *Comput. Ind.* **59**, 601–611 (2008)
- [7] H.D. Hwang, S.H. Yoon, Constructing developable surfaces by wrapping cones and cylinders, *Comput. Aided Des.* **58**, 230–235 (2015)
- [8] R. Bodduluri, B. Ravani, Design of developable surfaces using duality between plane and point geometries, *Comput. Aided Des.* **25**, 621–632 (1993)
- [9] R.M.C. Bodduluri, B. Ravani, Geometric design and fabrication of developable Bézier and B-spline surfaces, *J. Mech. Des.* **116**, 1042–1048 (1994)
- [10] M. Zhou, G.H. Peng, Z.L. Ye, Design of developable surfaces by using duality between plane and point geometries, *J. Comput. Aided Geom. Des.* **16**, 1401–1406 (2004)
- [11] M. Zhou, J.Q. Yang, H.C. Zheng et al., Design and shape adjustment of developable surfaces, *Appl. Math. Model.* **37**, 3789–3801 (2013)
- [12] G. Hu, H.X. Cao, X.Q. Qin et al., Geometric design and continuity conditions of developable λ -Bézier surfaces, *Adv. Eng. Softw.* **114**, 235–245 (2017)
- [13] G. Hu, H.X. Cao, X.Q. Qin, Construction of generalized developable Bézier surfaces with shape parameters, *Math. Methods Appl. Sci.* **41**, 7804–7829 (2018)
- [14] G. Hu, H.X. Cao, S.X. Zhang, Developable Bézier-like surfaces with multiple shape parameters and its continuity conditions, *Appl. Math. Model.* **45**, 728–747 (2017)
- [15] G. Hu, J.L. Wu, X.Q. Qin, A new approach in designing of local controlled developable H-Bézier surfaces, *Adv. Eng. Softw.* **125**, 27–54 (2018)
- [16] G. Hu, J.L. Wu, Generalized quartic H-Bézier curves: construction and application to developable surfaces, *Adv. Eng. Softw.* **138**, 102723 (2019)
- [17] G. Aumann, Interpolation with developable Bézier patches, *Comput. Aided Geom. Des.* **8**, 409–420 (1991)
- [18] K. Tang, M. Chen, Quasi-developable mesh surface interpolation via mesh deformation, *IEEE Trans. Vis. Comput. Graph.* **15** (3), 518–528 (2009)
- [19] M. Peternell, Developable surface fitting to point clouds, *Comput. Aided Geom. Des.* **21**, 785–803 (2004)
- [20] Y.J. Liu, Y.K. Lai, S.M. Hu, Stripification of free-form surfaces with global error bounds for developable approximation, *IEEE Trans. Autom. Sci. Eng.* **6**, 700–709 (2009)
- [21] L. Zeng, Y.J. Liu, M. Chen et al., Least squares quasi-developable mesh approximation, *Comput. Aided Geom. Des.* **29**, 565–578 (2012)
- [22] C.H. Chu, J.T. Chen, Geometric design of developable composite Bézier surfaces, *Comput. Aided Des. Appl.* **1**, 531–539 (2004)
- [23] G. Hu, H.X. Cao, X. Wang et al., G2 continuity conditions for generalized Bézier-like surfaces with multiple shape parameters, *J. Inequal. Appl.* **248**, 1–17 (2017)
- [24] C.Y. Li, C.G. Zhu, G1 continuity of four pieces of developable surfaces with Bézier boundaries, *J. Comput. Appl. Math.* (2018)

- [25] G. Hu, J.L. Wu, H.N. Li et al., Shape optimization of generalized developable H-Bézier surfaces using adaptive cuckoo search algorithm, *Adv. Eng. Softw.* **149**, 102889 (2020)
- [26] J.D. Schaffer, Multiple objective optimizations with vector evaluated genetic algorithms, *Proceedings of the 1st International Conference on Genetic Algorithms*, Pittsburgh, PA, USA, July 1985. Lawrence Erlbaum Associates Publishers, Hillsdale, 1985
- [27] N. Srinivas, K. Deb, Multiobjective optimization using nondominated sorting in genetic algorithms, *Evol. Comput.* **2**, 221–248 (1994)
- [28] K. Deb, S. Agrawal, A. Pratap et al., A fast and elitist multiobjective genetic algorithm: NSGA- II, *IEEE Trans. Evol. Comput.* **6**, 182–197 (2002)
- [29] C.A.C. Coello, G.T. Pulido, M.S. Lechuga, Handling multiple objectives with particle swarm optimization, *IEEE Trans. Evol. Comput.* **8**, 256–279 (2004)
- [30] C.A.C. Coello, N.C. Cortes, Solving multiobjective optimization problems using an artificial immune system, *Genet. Program. Evolvable Mach.* **6**, 163–190 (2005)
- [31] Q. Zhang, H. Li, MOEA/D: a multiobjective evolutionary algorithm based on decomposition, *IEEE Trans. Evol. Comput.* **11**, 712–731 (2007)
- [32] S. Mirjalili, A.H. Gandomi, S.Z. Mirjalili et al., Salp Swarm algorithm: a bio-inspired optimizer for engineering design problems, *Adv. Eng. Softw.* **114**, 163–191 (2017)
- [33] S.Z. Mirjalili, S. Mirjalili, S. Saremi et al., Grasshopper optimization algorithm for multi-objective optimization problems, *Appl. Intell.* **48**, 805–820 (2018)
- [34] D.E. Goldberg, D.M. Goldberg, *Genetic algorithm is searching optimization and machine learning*, Addison Wesley **xiii**, 2104–2116 (1989)
- [35] G. Hu, X. Zhu, G. Wei et al., An improved marine predator's algorithm for shape optimization of developable Ball surfaces, *Eng. Appl. Artif. Intell.* **105**, 104417 (2021)
- [36] J. Lu, X. Su, J. Zhong et al., Enhanced Salp Swarm algorithm for optimizing the shape of developable Bézier-like surfaces, *Math. Probl. Eng.* **2022**, 1796642 (2022)
- [37] G. Hu, X. Zhu, X. Wang et al., Multi-strategy boosted marine predators' algorithm for optimizing approximate developable surface, *Knowl. -Based Syst.* **254**, 109615 (2022)
- [38] G. Hu, M. Li, J. Zhong, Combined cubic generalized ball surfaces: construction and shape optimization using an enhanced JS algorithm, *Adv. Eng. Softw.* **176**, 103404 (2023)
- [39] S. BiBi, M.Y. Misro et al., Shape optimization of GHT-Bézier developable surfaces using particle swarm optimization algorithm, *Optim. Eng.* **24**, 1321–1341 (2023)
- [40] J. Zheng, X. Ji, Z. Ma, G. Hu, Construction of local-shape-controlled quartic generalized said-ball model, *Mathematics* **11**, 2369 (2023)

Cite this article as: J. Lu, X. Su, J. Zhong, G. Hu, Multi-objective shape optimization of developable Bézier-like surfaces using non-dominated sorting genetic algorithm, *Mechanics & Industry* **24**, **38** (2023)