applied sciences

MDPI

Article

B-Spline Curve Fitting of Hungry Predation Optimization on Ship Line Design

Changle Sun *, Mingzhi Liu and Shihao Ge

Department of Mechanical Engineering, Dalian Maritime University, Dalian 116026, China
* Correspondence: sunchangle@dlmu.edu.cn

Abstract: The ship line often describes by the offset table of discrete data points, which leads to the problems that three view coordinates may not correspond, the fitting error is large and the fitted curve cannot be easily modified. This will seriously affect the subsequent ship performance evaluation and op-timization. To solve this problem, this paper develops a B-spline curve fitting of hunger predation optimization on ship line design (HPA), which contains knot guidance technology, hungry preda-tion optimization technology and adaptive adjustment of algorithm input parameters. HPA transforms the discrete ship line into a continuous B-spline curve description, which improves the accuracy and modifiability of the ship line design. Through the real-time feedback of the results of each round of iteration, the knot vector is adaptively adjusted towards a better fitness, and then the optimal control point set that satisfies the error threshold can be obtained. The effectiveness and superiority of HPA are verified by comparing with related research and engineering software.

Keywords: knot placement; b-spline curve fitting; ship line design; optimization algorithm; reverse engineering

check for updates

Citation: Sun, C.; Liu, M.; Ge, S. B-Spline Curve Fitting of Hungry Predation Optimization on Ship Line Design. *Appl. Sci.* **2022**, *12*, 9465. https://doi.org/10.3390/app12199465

Academic Editor: Luís L. Ferrás

Received: 7 August 2022
Accepted: 16 September 2022
Published: 21 September 2022

1. Introduction

B-spline curve approximation has been a hot research topic in the field of computer-aided geometric design in recent years. With the development of computer-aided design (CAD) and the rise of advanced manufacturing technology, B-spline is widely used in the design and manufacturing industry of free curves and surfaces [1–5]. In the field of shipbuilding, the design of ship lines is the basis of shipbuilding [6]. The ship lines are closely related to the dynamic characteristics and resistance characteristics of the ship. Improving the design level of ship lines and then improving the design quality of ships has become a hot research topic in the field of shipbuilding industry [7,8]. However, due to the error in the design of the ship lines or the modification in the local smoothing process, the coordinates of the corresponding points of the three views may not correspond, resulting in the reduction of the accuracy of the ship line diagram [9]. In addition, the data points in the ship line diagram are discrete, and two adjacent discrete points are connected by straight line segments, resulting in poor smoothness, which brings trouble to the subsequent ship line setting out and processing and affects the quality and efficiency of shipbuilding.

Due to the locality of B-spline, it can be more convenient to modify the curve locally. The parameter continuity of B-spline can ensure that the fitting curve has good smoothness. If the continuous B-spline method is used in the ship line design to replace the discrete offset table method, the accuracy of ship line design will be greatly improved, and the subsequent multi-disciplinary collaborative optimization and manufacturing will be facilitated.

Compared with interpolation, B-spline curve approximation can better reflect the shape of curve, but the placement of knot vector is still an urgent problem to be solved. Unreasonable knot vector arrangement may lead to unacceptable shapes [10]. Researchers put higher fitting accuracy at the same knot number [11,12], fewer knot number under the same accuracy requirements [13,14] and faster operating efficiency [15,16] as the main pursuit goals of B-spline curve fitting.

This paper develops a B-spline curve fitting of hunger predation optimization on ship line design (HPA). In the previous dynamic knot method [17–22], the initial knots are randomly selected from the interval [0, 1], so there are many unreasonable knot vectors. Therefore, large population size and iteration number are required, which leads to low efficiency of the algorithm. The knot guidance technology is designed to add knots in the area with complex model shape at the initial knot selection stage. The premature loss of population diversity in optimization algorithm leads to slow convergence and is easy to fall into local optimality. A hunger search strategy is developed to make the hungry individuals in the population approach the optimal solution more quickly, and the influence of neighbors on the position adjustment is further considered. Aiming at the problem that the previous dynamic knot method requires manual adjustment of key input parameters, such as population size and iteration number, which is troublesome and time consuming. An adaptive adjustment of key input parameters in the HPA algorithm is proposed, which can quickly adapt to the replacement of model and fitting accuracy. HPA achieves the goal of B-spline curve fitting with higher fitting accuracy at the same control points, less control points under the same accuracy requirements, faster operation efficiency and better universality; it can better solve practical engineering problems.

The rest of the paper is organized as follows. Section 2 introduces the basic B-spline theory used in HPA algorithm and research on existing B-spline curve fitting. Section 3 proposes the HPA method, including a new knot guidance technology, a hungry predation optimization technology, fitness function selection and the dimension calculation rule. Section 4 analyzes the influence of population size, iteration number and error threshold of HPA on curve fitting results, and an adaptive adjustment method of initial input parameters is proposed. Section 5 verifies the performance of HPA algorithm; we verify the superior performance of HPA algorithm in fitting accuracy and convergence speed by comparing with typical static knot method, dynamic knot method and engineering commercial software. Finally, we summarize this paper and look forward to future research directions.

2. Related Works

2.1. B-Spline Theory Knowledge

This section refers to the bibliography [23,24]. The knot vector U consists of the non-decreasing parameter u_i, U: $u_0 \leq u_1 \leq \cdots \leq u_{n+\rho+1}$, where the dimensions and values of the initial knot vector are random under certain constraints. After that, they will be adjusted with each iteration. It will be introduced in detail in Section 3. So far, B-spline basis function $N_{i,\rho}(x)$ can be solved, as shown in Formula (1) and (2),

$$\begin{array}{l} N_{i,0}(x) = 1, \;\; u_i \leq x \leq u_{i+1} \; (i = 0, \ldots, n) \\ \qquad\qquad 0, \;\; \text{other} \end{array} \tag{1}$$

$$N_{i,\rho}(x) = \frac{x - u_i}{u_{i+\rho} - u_i} N_{i,\rho-1}(x) + \frac{u_{i+\rho+1} - x}{u_{i+\rho+1} - u_{i+1}} N_{i+1,\rho-1}(x) \tag{2}$$

where the first subscript i represents the sequence number, and the second subscript ρ represents degree (order ρ + 1) of curve.

Inappropriate degree may lead to the fitting curve cannot meet the accuracy requirement. Designers prefer to use piecewise low degree curves to describe complex curves, and they do not want a curve to be divided into too many segments. Because the cubic curve is not only a plane curve with inflection point, but also the lowest degree of spatial curves, it is widely used [24].

A polynomial B-spline curve of degree ρ (or order ρ + 1) is a piecewise polynomial curve given by

$$p(x) = \sum_{i=0}^{n} d_i N_{i,\rho}(x) \quad x \in [0, 1] \tag{3}$$

where d_i is the control point of the curve and $N_{i,\rho}(x)$ is the ρ-degree-normalized B-spline basis function.

According to Formula (3), it is necessary to obtain the control point d_i to realize the curve fitting, and the solution of d_i adopts the standard least square method and endpoint constraints. Interpolate the first and last data points $P_0 = p(0)$, $P_m = p(1)$, and other data points P_i ($i = 1, 2, \cdots, m-1$) are approximated by the least-squares minimization method.

The given data points are parameterized by the normalized accumulation chord length, as shown in Formula (4),

$$\begin{gathered} x_0 = x_0^* = 0 \\ x_i^* = x_{i-1}^* + |\Delta P_{i-1}| \quad i = 1,\ 2,\ \cdots,\ m \\ x_i = \frac{x_i^*}{x_m^*} \end{gathered} \tag{4}$$

where x_i is the parameter value corresponding to the current data point, and $\Delta P_{i\,-\,1}$ is the forward difference vector, $\Delta P_{i-1} = P_i - P_{i-1}$.

The objective function is the square difference between each data point and the corresponding point on the curve fitted according to the new knot vectors obtained by HPA in the previous iteration, as shown in Formula (5). Then we can get Formula (7) through the transformation of Formula (6).

$$f = \sum_{i=1}^{m-1} [\, P_i - p(x_i)\,]^2 \tag{5}$$

Let

$$r_i = P_i - P_0 N_{0,\rho}(x_i) - P_m N_{n,\rho}(x_i) \quad i = 1,\ 2, \cdots,\ m-1 \tag{6}$$

Then,

$$f = \sum_{i=1}^{m-1} [r_i - \sum_{j=1}^{n-1} d_j N_{j,\rho}(x_i)]^2 \tag{7}$$

The L-th derivative of it is shown in Formula (8), in order to minimize the objective function f, the derivative of $n-1$ control points d_j needs to be 0, and then the final least-square fitting in Formula (9) can be obtained to solve the control point d_j in the current number of iterations.

$$\frac{\partial f}{\partial d_L} = \sum_{i=1}^{m-1} [-2r_i N_{L,\rho}(x_i) + 2N_{L,\rho}(x_i) \sum_{j=1}^{n-1} d_j N_{j,\rho}(x_i)] \tag{8}$$

$$\sum_{j=1}^{n-1} (\sum_{i=1}^{m-1} N_{L,\rho}(x_i) N_{j,\rho}(x_i)) d_j = \sum_{i=1}^{m-1} r_i N_{L,\rho}(x_i)] \tag{9}$$

According to the obtained basis functions $N_{i,\rho}(x)$ and control points d_j, they are brought into Formula (3) to complete the curve fitting.

2.2. Research Status

At present, many companies have developed relatively mature ship CAD software systems such as AVEVA's TRIBON Solution, DASSAULT's Catia, NAPA's NAPA system, etc. In addition, In-Il Kim [25], Hyeon-deok Lee [26] and Dongkon Lee et al. [27] also developed a special ship design system. The dedicated ship CAD system effectively improves the efficiency and quality of ship design. However, it has not achieved a better solution to the problem of discrete point processing in the offset table. For example, in the research on the reconstruction of the segmented outer plate of the ship surface, Yu Yong et al. [28] carried out the rapid inverse calculation of the non-uniform B-spline curve, but the inverse calculation process is to interpolate the discrete points, which cannot avoid errors. Nowadays, researchers prefer B-spline approximation. Moreover, the knot placement problem is a popular content that has been studied by many scholars in recent years [29]. The existing research methods can be divided into the static knot method and dynamic knot method according to whether the knot can be adjusted after selection. We first introduce the relevant static knot methods. Piegl L and Tiller W [23] first proposed an averaging technique and knot placement technique (KTP) in 1978, and knots are selected to reflect the distribution of data points. On this basis, Piegl L and Tiller W put forward NKTP technology [30]. Lyche T proposed a knot removal method (KRM) in 1988 [31,32]. However, this method cannot capture the internal characteristics of data points, and the amount of calculation is huge. In 1999, Razdan A [33] proposed to use the shape information of data points for knot placement. In the research method of Park H and Lee J [34], the idea of knot adaptation was proposed. Xu Jin, Ke Yinglin et al. [35] further proposed a feature point including the crease point, inflection point and curvature maximum point. The influence of curvature on fitting has also become a more concerned factor [36–39]. The static knot method is usually simple to calculate and has higher efficiency; the knot calculation comes from artificial assumptions and cannot be moved after placement. Compared with the dynamic knot method, the static knot method requires more knots under the same precision requirement. The dynamic knot method is mainly a combination of B-spline curve fitting theory and optimization algorithm, and the overall algorithm performance largely depends on the optimization algorithm. There are hundreds of existing optimization algorithms, and the nature-inspired optimization algorithm [40–47] is one of the most practical branches. According to the algorithm principle, it can be divided into three categories, as shown in Figure 1. Yoshimoto F et al. [17] and Sarfraz M [18] proposed the application of genetic algorithm. Özkan İNİK et al. [19] combined the grey wolf algorithm. He Bingpeng et al. [20] combined the differential evolution algorithm. Kübra Uyar et al. [21] combined the invasive weed optimization (IWO). Akemi Gálvez et al. [22] proposed the combination of immune algorithm and B-spline curve fitting and improved the parameter optimization and algorithm complexity. The dynamic knot method realizes the adaptive adjustment of knot dimension and position in each iteration; it does not rely on artificial assumptions and carries out forward calculation through multiple groups of initial random knots. The fitting effect of dynamic knot method is often better than the static knot method, but its operation efficiency is low.

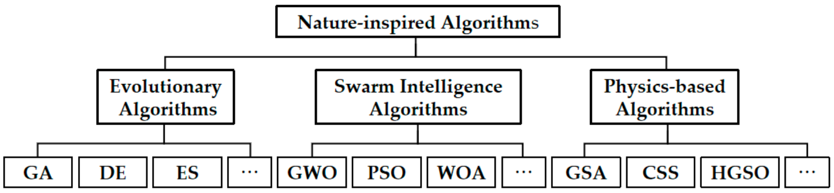


Figure 1. Classification of nature-inspired optimization algorithms.

3. A B-Spline Curve Fitting of Hungry Predation Optimization Algorithm with Knot Guidance

3.1. Knot Guidance Technology

In B-spline curve fitting, zero inner knots represent a straight line. As the shape complexity increases, so does the inner knot number. The knot guidance technology proposed in this paper mainly captures the feature points of the curve through the preprocessing of the curve data points. The feature points include jump points (position discontinuity points), sharp points (tangent discontinuity points), curvature discontinuity points, curvature extreme points and inflection points. This paper mainly focuses on the engineering application of ship line, which includes three types of feature points, as shown in the Figure 2.

According to the Formula (10) [38],

$$K_i = \frac{2\Delta P_{i-1}P_iP_{i+1}}{l_il_{i+1}l_i^*} = \operatorname{sgn}(\Delta P_{i-1}P_iP_{i+1})\frac{\operatorname{sinff}_i}{l_i^*}\ (i = 1,2,\ldots,m\ -\ 1) \tag{10}$$

where K is the curvature of the data point; P is the data point; subscript i is the serial number of the corresponding data point; $\Delta P_{i-1}P_iP_{i+1}$ is the triangular area composed of data point P_{i-1}, P_i and P_{i+1}; sgn is the symbolic function; $\Delta P_{i-1}P_iP_{i+1}$ is positive when the order is counterclockwise; l is the chord length between two points; and α_i is the angle of the chord, as shown in the Figure 3.

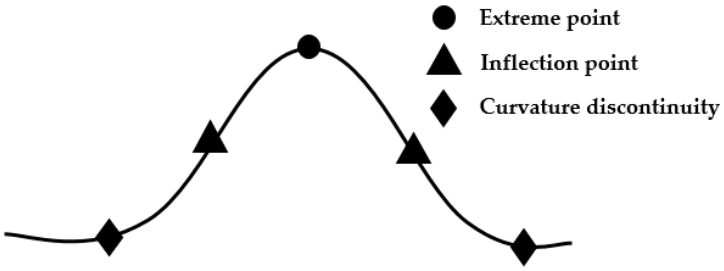


Figure 2. Feature points of curve.

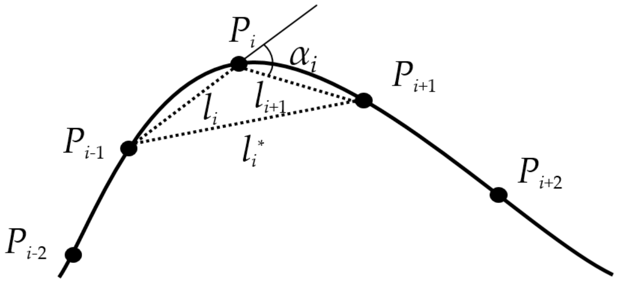


Figure 3. Schematic diagram of Formula (9).

After obtaining the curvature of each point, find the position of the feature point and the corresponding parameter value u_i. In order to ensure the diversity of the population size, half of population still take random values in the interval [0, 1]. The other half takes values in each interval [u_i − 1, u_i + 0.1]; each knot has a 50% chance to mutate, and the value interval after mutation is [0, 1]. The method of finding feature points is described in reference [35].

3.2. Hungry Predation Optimization Algorithm

The adaptive adjustment of knots in HPA simulates the predator–prey behavior of animals under starvation. Distinguish individual levels in the population based on fitness function. Those with higher fitness are α, and the remaining individuals are collectively referred to as ω.

The three groups of knot vectors with the best fitness and their coordinates are:

$$U_p^{\alpha_1}= (u_p^1, u_p^2, \cdots, u_p^{D\alpha_1})$$

$$U_p^{\alpha_2}= (u_p^1, u_p^2, \cdots, u_p^{D\alpha_2})$$

$$U_p^{\alpha_3}= (u_p^1, u_p^2, \cdots, u_p^{D\alpha_3})$$

Traverse the population in turn, and the corresponding position coordinate is:

$$U_i= (u_i^1, u_i^2, \cdots, u_i^D)$$

If the individual is α, then its position remains unchanged. If the individual is ω, the next position U_{i+1} is described as Formulas (11)–(13),

$$a = 2 - Iter/Max\_iter \tag{11}$$

$$U_{i+1}^{\alpha_k}= U_p^{\alpha_k} - A|CU_p^{\alpha_k} - U_i|\ k = 1,2,3 \tag{12}$$

$$U_{i+1} \ = \sum_{k=1}^{3} U_{i+1}^{\alpha_k}/3 \tag{13}$$

where A is a random number between $[-a, a]$, C is a random number between [0, 2] and *Iter* is the current number of iterations.

At this time, the initial adjustment of the population position has been achieved, and each individual has a 50% chance to become a hungry individual. The hungry individual will further hunger search and move faster to the prey position.

First, use the Euclidean distance between the current position U_i and the position U_α of the α_1 to calculate the radius R_i of the neighbor range, as shown in Formula (14).

$$R_i = ||U_i - U_{\alpha_1}|| \tag{14}$$

Then, traverse the population size to find neighbors, as shown in Formula (15).

$$N_i = \{U_j | D_i(U_i, U_j) \ \le \ R_i\} \tag{15}$$

where N_i is the set of neighbors and U_j is the location of the neighbor.

Finally, move the position, as shown in Formula (16).

$$U_{i+1} = U_r + (U_r - U_i) \times S \ \times \ \cos(\frac{Iter}{Max\_iter} - 0.5) \tag{16}$$

where U_r has a 50% probability of being the location of α, and 50% is the neighbor in the population size. S is a random number between (0, 1).

3.3. Fitness Function Selection

As a key component of the algorithm, the selection of fitness function directly affects whether it can find the optimal solution and the convergence speed of HPA. In the whole process of population size search, it does not rely on external information, only based on the fitness function. Individuals in a population adjust their positions based on fitness.

This paper proposes three fitness functions that can be applied to HPA.

(a) Bayesian Information Criterion (BIC) [22]: its fitness function is described as Formula (17).

$$Fitness = -2\mathrm{Ln}(L_1) + \mathrm{Ln}(n_1) \ \times \ N_p \tag{17}$$

where L_1 is the likelihood function, n_1 is the sample size and N_p is the number of parameters.

(b) Maximum error of a single point of the curve is described as Formula (18).

$$Fitness = Max(P_{\mathrm{ir}} - P_{if}\Big) \tag{18}$$

where P_{ir} is the actual given data point, and P_{if} is the point on the corresponding fitting curve.

(c) Overall standard deviation of curve is described as Formula (19).

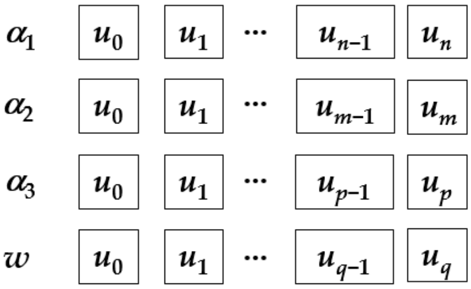
$$Fitness = \sqrt{\frac{\sum_{i=1}^{n} (P_{ir} - P_{if}\Big)^2}{N_k}} \tag{19}$$

where N_k is the number of knot vertices.

The BIC method considers the complexity of the calculation model and avoids the over fitting problem by adding the penalty term of model complexity. When the number of knots and control points is small, the BIC value will decrease with increasing the number of knots until it reaches the minimum value. Then, as the number of knots continues to increase, the accuracy of the curve improves slowly, and the BIC value will increase. In practical engineering application, Formulas (18) and (19) are more intuitive and convenient.

3.4. Dimension Calculation Rule

In the iterative calculation of population position, different individuals have different internal knot rates λ, which causes the dimensions to be different for each individual, as shown in Figure 4. The dimension corresponds to the number of internal knots. In order to find fewer control points to complete the curve fitting, we must ensure the diversity of individuals. When ω_i moves towards or away from α, its dimension remains unchanged. We deal with α whose dimension is different from ω_i. Then we can get $U_p^{\alpha 1}$, $U_p^{\alpha 2}$ *and* $U_p^{\alpha 3}$, and all of them have the same dimensions with ω_i.



α_1	u_0	u_1	$\cdots$	u_{n-1}	u_n
α_2	u_0	u_1	$\cdots$	u_{m-1}	u_m
α_3	u_0	u_1	$\cdots$	u_{p-1}	u_p
w	u_0	u_1	$\cdots$	u_{q-1}	u_q

Figure 4. Dimensions of different individuals.

First, the dimensions of α are judged. If the dimension is higher than that of the currently selected ω_i, calculate the extra number of dimension $N1$ and the difference M_i between every two adjacent inner knots of the high-dimensional individual.

The probability p_i of each point to be deleted is shown in Formula (20):

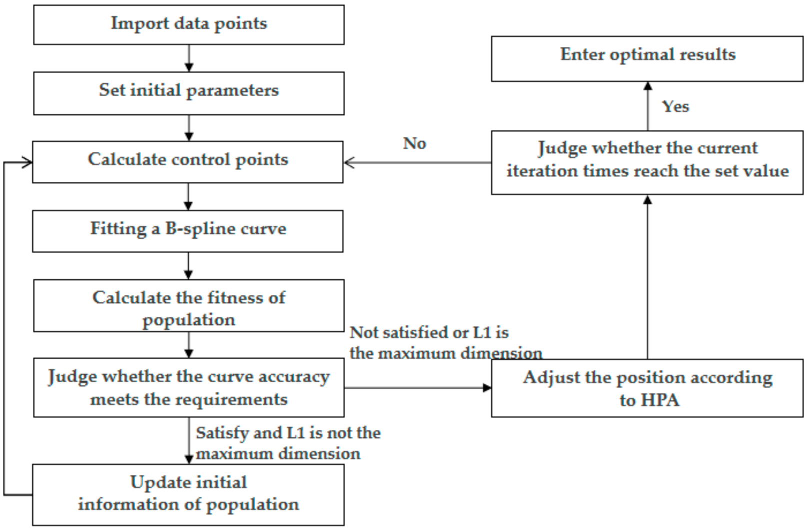
$$p_i = \frac{M_i}{\sum_{i=0}^{m} M_i}(i = 0,\ 1,\ \cdots,\ m) \tag{20}$$

Randomly delete a knot according to the probability p_i, and this process is repeated $N1$ times until two individuals have the same dimension.

If the dimension is lower than that of the currently selected individual, calculate the extra number of dimension $N1$ and the difference M_i between every two adjacent inner knots of the high-dimensional individual. Randomly add a new knot in the interval with the largest inner knot difference M_i and repeat this process to $N1$ times, so that the dimensions of the two individuals are the same.

3.5. Algorithm Flow

The overall algorithm flow of HPA is shown in Figure 5.



Import data points
Set initial parameters
Calculate control points
Fitting a B-spline curve
Calculate the fitness of population
Judge whether the curve accuracy meets the requirements
Satisfy and L1 is not the maximum dimension
Update initial information of population
Not satisfied or L1 is the maximum dimension
Adjust the position according to HPA
Judge whether the current iteration times reach the set value
No
Yes
Enter optimal results

Figure 5. Algorithm flow of HPA.

Step 1: Import the data points of the curve, order the data points and carry out the parameterization of the standard accumulation chord length to obtain the corresponding parameter values.

Step 2: Set the error threshold, population size and iteration number, which are adjusted adaptively according to the feature points and error threshold, which is introduced in Section 4. The corresponding internal knots are obtained through the knot guidance technology.

Step 3: Based on the existing knot vector, curve degrees and data point, the respective control points are obtained by using the least-squares minimization method, in which the curve degrees and data point parameters are fixed values, and the value and dimension of each individual's knot vectors may change after each iteration.

Step 4: According to the existing knot vectors and the control points obtained in Step 3, a B-spline curve can be fitted by using Formula (3).

Step 5: Give a fitness function, calculate the fitness of each individual.

Step 6: Judge whether the curve fitted by each individual meets E. If the individual meets E, and its dimension is not the largest among the population, then go to step 7. If it does not meet E or the individual's dimension is the largest among the population, go to step 8.

Step 7: Determine the minimum dimension D of population that satisfies the condition, remove individuals whose dimension is greater than D, supplement the population with the same number of individuals with dimension of D or $D-1$ and return to step 3.

Step 8: Use the hungry predation optimization technology to adjust the position of each individual.

Step 9: Judge whether the current iteration number reaches the set value *Max_iter*. If it reaches, output the current optimal result. If it does not reach *Max_iter*, return to step 3.

4. Key Parameter Settings

There are four key parameters of the HPA:

(1) Population size W, which determines how many individuals find the optimal solution in the population size at the same time.

(2) Internal knot rate λ: The range of λ depends on the number of given data points, usually select interval [0, 0.5], which can be reduced with the increase of data points. The λ is different for each individual.

(3) Error threshold E, which ensures that the error value of the final fitted curve is less than E.

(4) Iteration number *Max_iter*, which determines the number of iterations of the whole optimization process, iterates to the *Max_iter* time and outputs the current optimal solution.

In previous studies [17–22], the population size and iteration number have been based on different fitting models and fitting error thresholds, relying on intuitive experience or multiple attempts to give appropriate values, which is troublesome and time consuming. This section takes two continuous semicircle models with radius 1 as an example, as shown in Figure 6. We further explore the impact of population size, iteration number and error threshold on curve fitting effect, and give the setting formula of population size and iteration number.

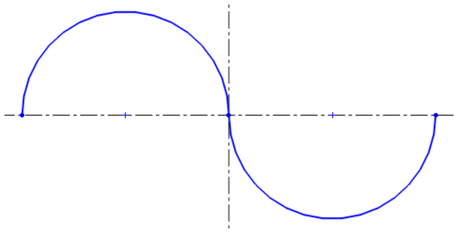


Figure 6. Two continuous semicircles. Radius: 1.

The population size determines the number of parallel solution paths. When the iteration number *Max_iter* is 20 and the error threshold *E* is 0.1, the population size *W* is set to 10, 20 and 50, respectively; the fitting result is recorded as shown in Figure 7.

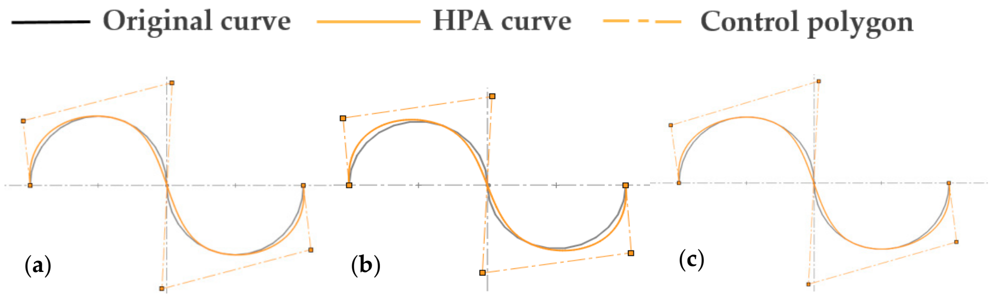


Figure 7. Influence of population size *W* on curve fitting, (**a**) *W* = 10, (**b**) *W* = 20 and (**c**) *W* = 50.

The inner knots of Figure 7a are 0.4328 and 0.5771, and the fitting accuracy is 0.0186. The inner knots of Figure 7b are 0.4612 and 0.5376, and the fitting accuracy is 0.0207. The inner knots of Figure 7c are 0.4076 and 0.5924, and the fitting accuracy is 0.0169. It can be seen from Figure 7 that the fitting result with *W* = 20 is worse than *W* = 10 in this experiment.

The setting of the iteration number affects the step size of each iteration. The more the iteration number, the more carefully the algorithm searches. However, when the iteration number is large enough, increasing the iteration number will have little effect on the curve fitting accuracy. Besides, a large iteration number will improve the operation time of the algorithm. When the error threshold *E* is 0.1, multiple experiments are carried out by changing the population size *W* and iteration number *Max_iter*. The results are shown in Figure 8.

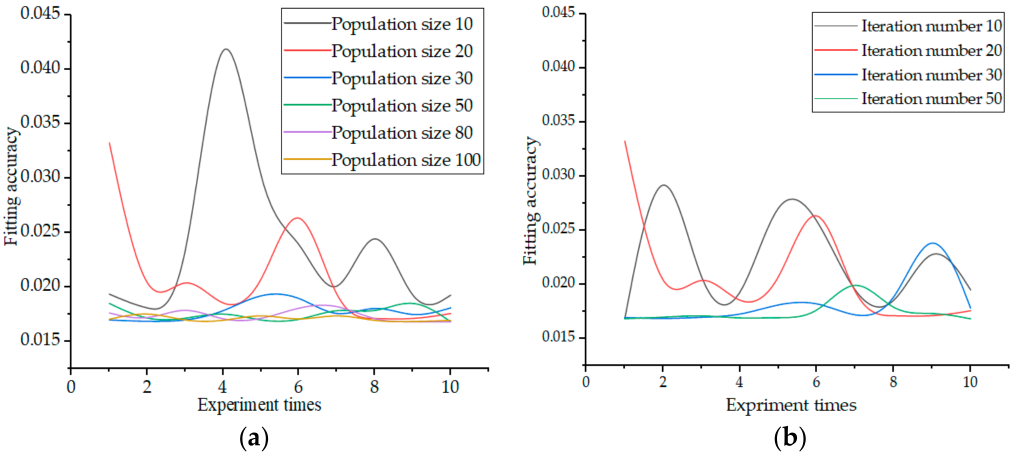


Figure 8. Fluctuation of curve fitting accuracy when (**a**) *Max_iter* is 20 and *W* varies from 10 to 100, and when (**b**) *W* is 20 and *Max_iter* varies from 10 to 50.

As shown in Figure 8a, *Max_iter* is 20. When *W* is 10 and 20, the result fluctuates greatly, and the fluctuation is relatively small when *W* is 30. In Figure 8b, *W* is 20, and the fluctuation is small after 30 iterations.

The setting of the error threshold determines the fitting accuracy of the B-spline curve. The HPA algorithm can better implement the B-spline curve with the least number of control points when the error threshold is satisfied. Set *W* to 30, *Max_iter* to 20 and set *E* to 0.1, 0.01 and 0.001, respectively; the fitting results are shown in Figure 9.

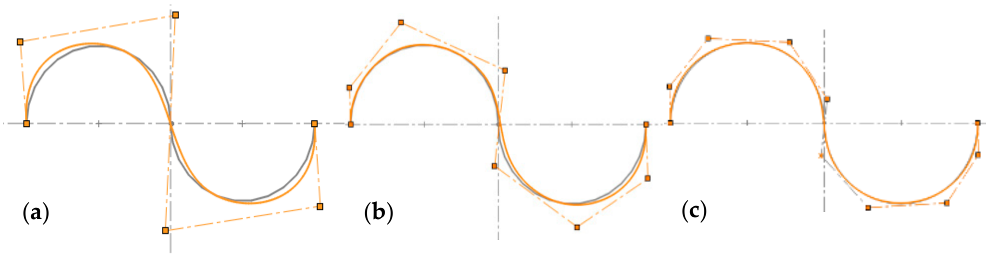


Figure 9. When *E* is (**a**) 0.1, (**b**) 0.01 and (**c**) 0.001, the influence of their changes on curve fitting.

The least-squares fitting accuracies of Figure 9a–c are 0.0207, 0.0016, and 0.0004, respectively. It can be seen from the results that as *E* decreases, the final curve fitting result is getting better, and more control points are required.

Adjust *E* from 1 to 0.0001, and the change of inner knot number required for the fitted curve, which is shown in Figure 10.

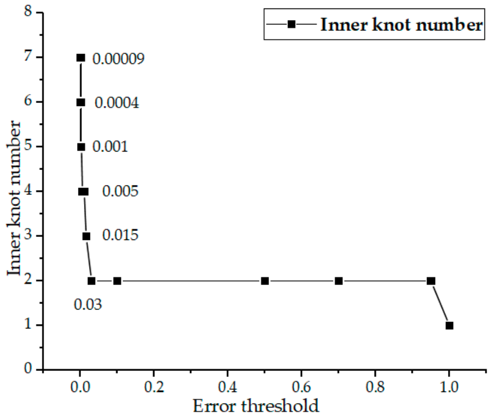


Figure 10. Influence of error threshold on the inner knot number.

The optimization algorithm in the field of B-spline curve fitting is to solve the extremum problem, and the solution is a number of interrelated decimals arranged in ascending order from 0 to 1. The setting of population size and iteration number is mainly related to the number and distribution of solutions. It is affected by factors such as error threshold and model complexity. A small numerical setting will make it difficult to find the optimal solution, and a large numerical setting will lead to the algorithm low efficiency. Therefore, before the algorithm runs, we need to set a reasonable population size *W* and iteration number *Max_iter*. This paper gives the relational Formulas (21) and (22).

$$W = \lfloor -\mathrm{In}\left(\frac{E}{L_{MAX}}\right) \times 1.2^{\mathrm{Log}_{10}\left(\frac{L_{MAX}}{E}\right)} + 1.5 \times Cp \rfloor \tag{21}$$

$$Max\_iter = 2 \times W \tag{22}$$

where L_{MAX} is the maximum side length of the model, *Cp* is the number of model feature points and $\lfloor\rfloor$ is rounded down.

The design of Formulas (21) and (22) makes the HPA algorithm only need to input the model data points and the required error threshold, and the HPA algorithm can adaptively adjust the population size and iteration number. The whole algorithm has better universality, and it is ensured that the algorithm can obtain a better fitting effect at a higher operating efficiency.

5. HPA Algorithm Test

The experiment is divided into two parts. Section 5.1 compares the static knot method and dynamic knot method, and Section 5.2 compares with the existing commercial Software Solidworks2022 (Dassault, US) and Catia2017 (Dassault, France).

5.1. Comparison with Existing Research

When we deal with the problem of knot placement, except for the method of combining optimization algorithm, B-spline curve fitting based on adaptive curve refinement using dominant points (DOM), KTP, NKTP, KRM etc. are the best representative methods, as shown in Table 1. Figure 11 is a comparison example of HPA in this paper and DOM method. Park H compared related algorithms, as shown in Figure 12. Furthermore, we add the fitting results of GWO and HPA to Figure 12. Compared with related methods, HPA uses less control points under the same accuracy.

Table 1. Comparison of related fitting methods.

	Incremental Method Using KTP or NKTP	**Knot Removal Method KRM**	**Incremental Method Using DOM**	**Adaptive Adjustment Using HPA**
Preprocessing	1 Parameterization	1 Parameterization 2 Interpolation of all points	1 Parameterization 2 Selection of seed points	1 Parameterization 2 Find feature points
Iteration process	1 Knot placement 2 Least-squares minimization 3 Deviation check	1 Selection of a candidate knot 2 Deviation check 3 Knot removal	1 Knot placement 2 Least-squares minimization 3 Deviation check 4 Selection of a new dominant point	1 Knot placement 2 Least-squares minimization 3 Comparative fitness 4 Knot adaptive adjustment
Ref.	[23,30]	[31]	[34]	Our method

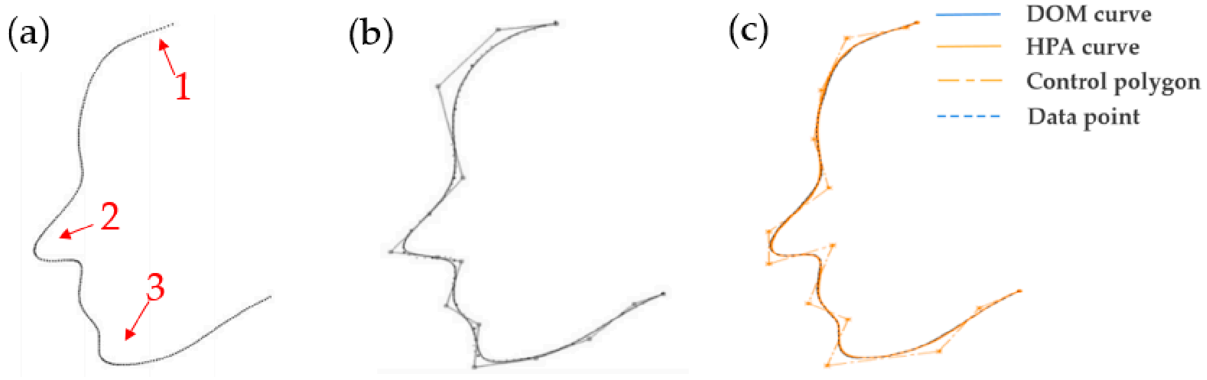


Figure 11. Comparing (**a**) the face data points and the fitting results of (**b**) DOM and (**c**) HPA especially at 1–3 marked.

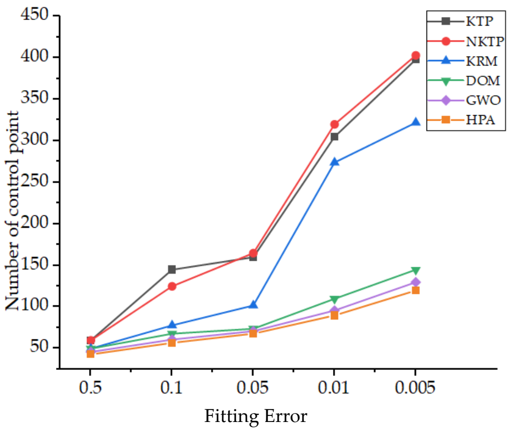


Figure 12. Comparison of control point number used at the same fitting error with DOM, KTP, NKTP, KRM, GWO and HPA.

As shown in Figure 11, the same face contour is fitted with the same 15 control points. From the fitting results, it can be seen that the HPA proposed in this paper can better represent the curve shape of face contour, especially at specially marked 1, 2 and 3 points, which are fuller at the tip of nose and more prominent at the chin compared with DOM.

Operational efficiency is an important criterion for judging the pros and cons of an algorithm. However, due to the different research periods of each algorithm, there are differences in both software and hardware. It is not objective to simply compare the running time with previous literature. The optimization algorithms of the natural heuristic class take a similar time for each iteration, and it is fairer to compare the convergence of the same number of iterations. This paper selects six representative optimization algorithms for face contour fitting: Genetic Algorithm [40], Differential Evolution Algorithm [41], Grey Wolf Algorithm [42,43], Whale Algorithm [44], Particle Swarm Algorithm [45] and Gravity Algorithm [46]. The search ability and convergence efficiency of HPA and other algorithms are compared when using 15 control points, as shown in Figure 13.

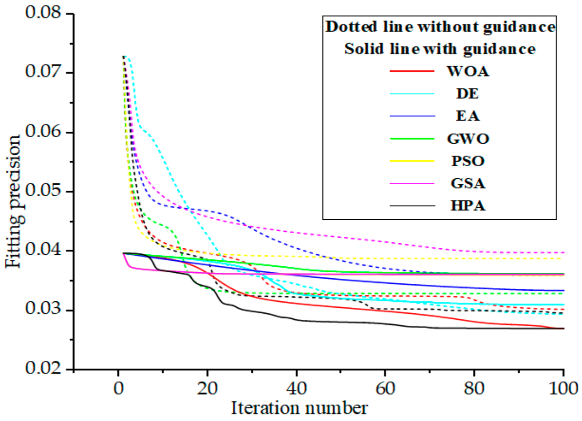


Figure 13. Comparison of the effect of various optimization algorithms on face contour fitting.

It can be seen from the above experiments that after knot guidance technology, the algorithm can find a better solution in the first generation. Compared with other algorithms, HPA tends to have higher fitting accuracy and achieve faster convergence at the same number of iterations. Besides, DE, WOA and HPA demonstrate stronger search capabilities. The DE algorithm uses a greedy algorithm, which makes the entire calculation time approach twice that of other algorithms.

The paper further selects the six models mentioned in the literature [19] for testing, as shown in Table 2, and compares them with the GWO method mentioned in the literature. The results of fitting the model are shown in Figure 14 and Table 3.

Table 2. Test six models mentioned in the literature. (Adapted with permission from Ref. [19]).

Fuction	Description	Variable Range
1	Data point [48]	
2	$f_2(x) = \frac{10x}{(1+100x^2)}$	$[-2, 2]$
3	$f_3(x)= 0.2\mathrm{e}^{-0.5x}\sin 5x + 4$	$[0, 4\pi]$
4	$f_4(x) = \frac{100}{\mathrm{e}^{\lvert 10x-5\rvert}} + \frac{(10x - 5)^5}{500}$	$[0, 1]$
5	$f_5(x)= \sin x + 2\mathrm{e}^{-30x^2}$	$[-2, 2]$
6	$f_6(x)= \sin 2x + 2\mathrm{e}^{-36x^2}+2$	$[-2, 2]$

It can be seen that the fitting results of the HPA method proposed in this paper for the six models are much better than GWO method at the same population size, iteration number, data points and number of knots.

The test function $f_4(x)$ is a more challenging function in the field of B-spline fitting [22], and there is a sharp point at x = 0.5, which is a commonly used example in the B-spline curve fitting literature. Existing fitting methods based on optimization algorithm and their fitting results are shown in the Table 4; PESA, MOGA, FFA and some other methods cannot fit the $f_4(x)$ function well. BIC is a criterion for many researchers to judge the pros and cons of an algorithm. However, each scholar has a slightly different understanding of the parameters in the BIC formula. Therefore, the BIC judgment formula in the B-spline curve fitting application is different.

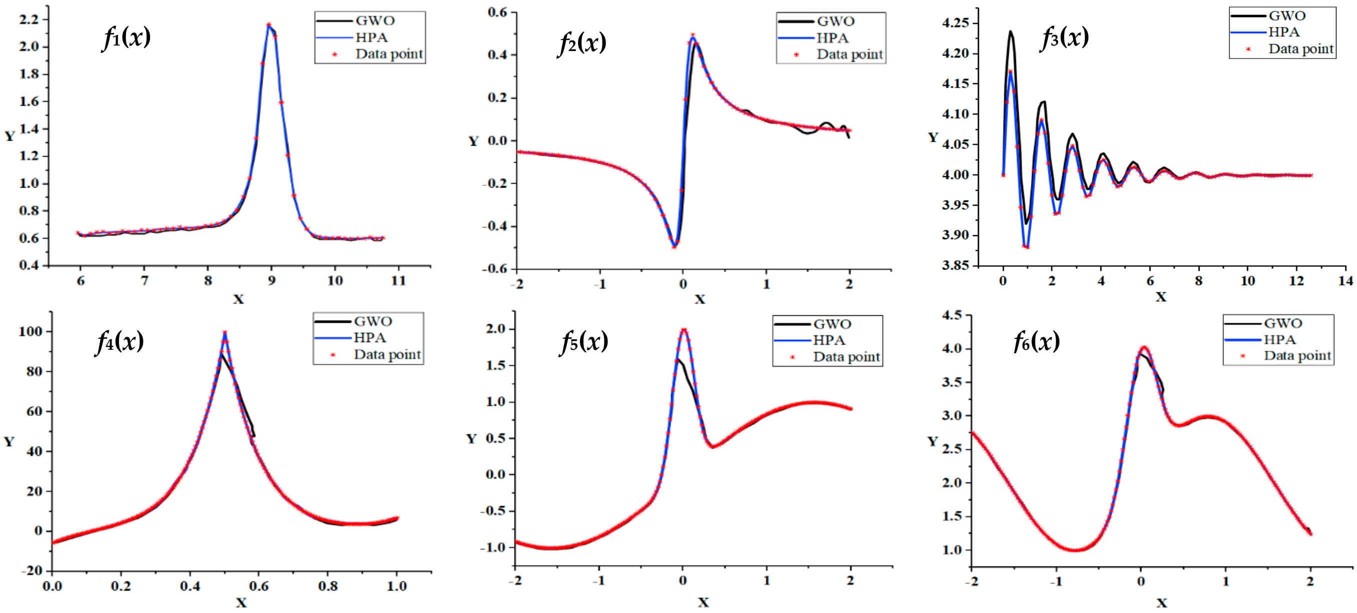


Figure 14. Comparison of GWO and HPA fitting results for six test models.

Table 3. The experimental results for all functions.

Fuction	Degree of Curve	Population Size	Number of Iteration	Number of Point	Number of Knot	Number of Control Point	MSE (GWO)	MSE (HPA)
1	3	50	100	49	16	20	0.024	3.21×10^{-3}
2	3	50	100	90	53	57	0.010	6.32×10^{-7}
3	3	50	100	200	77	81	0.008	2.31×10^{-6}
4	3	50	100	201	40	44	1.395	3.18×10^{-3}
5	3	50	100	201	46	50	0.032	2.43×10^{-4}
6	3	50	100	201	37	41	0.026	7.58×10^{-4}

Table 4. Existing fitting methods based on optimization algorithm and their fitting results. ('×' means that the method can not fit the function f_4 well).

Authors, Year and Reference	Method	Iteration Number	Knot Number	BIC
Yoshimoto et al. [17]	Genetic algorithms (GA)	200	5	1188
Sarfraz and Raza [18]	GA and Detection Algorithms	120	×	×
Özkan İNİK et al. [19]	Gray Wolf Optimization (GWO)	100	40	704
Kübra Uyar et al. [21]	Invasive Weed Optimization (IWO)	15	6	430
Gálvez et al. [22]	Artifficial Immune Systems (AIS)	100	5	1121
Ulker and Arslan [49]	Artifficial Immune Systems	500	×	×
Ulker [50]	Pareto Envelope-Based Selection Algorithm (PESA)	500	×	×
Valenzula et al. [51]	Multi-Objective Genetic Algorithms (MOGA)	120	×	×
Gálvez and Iglesias [52]	Fireflfly Algorithm (FFA)	500	×	×
Yuan et al. [53]	Adaptive Multiresolution Basis Set with Lasso Selection Method	Variable	×	×

In the AIS [22] method, the BIC calculation method is shown in Formula (23).

$$\begin{gathered} \mathrm{BIC} = N \times \mathrm{LN}(Q_1) + (\mathrm{LN}(N))(2Nod - 3\rho^{*} + 2) \\ Q_1 = \sum_{i=1}^{N} (X_{1i} - X_{2i})^2 + (Y_{1i} - Y_{2i})^2 \end{gathered} \tag{23}$$

where N is the number of data points, *Nod* is the number of knots and ρ^* is the order of the curve; X_{1i} and X_{2i} are the X coordinate values of the i-th point on the actual curve and the fitted curve, respectively; Y_{1i} and Y_{2i} are the Y coordinate values of the i-th point on the actual curve and the fitted curve, respectively.

In the testing process of the AIS method, 201 data points are selected at equal distances. The degree of curve ρ is 3 (ρ^* = 4). In order to further test the ability of the algorithm to withstand interference, a normal fluctuation with a mean of 0 and a variance of 1 was applied to the data points. The minimum BIC value is 1121.09 when k = 5. Under the same BIC calculation formula, the HPA algorithm obtains the smallest BIC value of 1040.57 when k = 9, as shown in Table 5.

Table 5. BIC values of AIS and HPA methods at the same knot.

Inner Knot Number	Method	Q_1	RMSE	BIC
κ = 3	HPA	500.889	1.578	1313.12
	AIS	593.977	1.719	1336.79
κ = 4	HPA	329.117	1.279	1239.32
	AIS	381.871	1.378	1258.60
κ = 5	HPA	169.758	0.919	1116.86
	AIS	182.761	0.953	1121.09
κ = 6	HPA	142.331	0.842	1092.05
	AIS	181.051	0.949	1129.81
κ = 7	HPA	122.786	0.782	1072.96
	AIS	178.013	0.941	1137.01
κ = 8	HPA	102.723	0.715	1047.71
	AIS	176.391	0.936	1145.78
κ = 9	HPA	94.045	0.684	1040.57
	AIS	174.515	0.931	1154.24
κ = 10	HPA	92.984	0.680	1043.60
	AIS	172.218	0.919	1159.61

In the Gray Wolf Optimizer (GWO) for knot placement in B-spline curve fitting [19], the BIC calculation formula is shown in Formula (24). The HPA proposed in this paper is much better than GWO in fitting the function $f_4(x)$.

$$\begin{gathered}\mathrm{BIC} = N \times \mathrm{LN}(\mathrm{MSE}) + (\mathrm{LN}(N))2 \times (2 \times Nod + \rho^*) \\ \mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}\sqrt{(X_{1i} - X_{2i})^2 + (Y_{1i} - Y_{2i})}^2\end{gathered} \tag{24}$$

where N is the number of data points, *Nod* is the number of knots and ρ^* is the order of the curve; X_{1i} and X_{2i} are the X coordinate values of the i-th point on the actual curve and the fitted curve, respectively; $Y_{1\mathrm{i}}$ and Y_{2i} are the Y coordinate values of the i-th point on the actual curve and the fitted curve, respectively.

The design of the BIC value in this article refers to the mean square error. Compared with the variance, the value will be smaller under the same fitting situation and may even be negative. When the optimal solution fitted by the GWO method is 40 knots, the BIC value is 704. However, the HPA algorithm obtains a minimum BIC value of 230.31 at k = 4, as shown in Table 6. Figure 15 records the fitting results of different numbers of knots of test function $f_4(x)$.

When the number of knots is small, the increase of knots can greatly improve the curve fitting accuracy, and the BIC value will decrease accordingly. When the inner knot is 4, the shape of the curve can be better fitted. At this time, the BIC value reaches the minimum value. As the number of knots continues to increase, the improvement of fitting accuracy begins to decrease, and the BIC value begins to increase.

In the GWO method, when 40 knots are used, the minimum BIC value [19] is obtained, and the fitting result of the test function $f_4(x)$ shows that the final fitting effect is not ideal, as shown in Figure 14. It can be seen that the HPA algorithm proposed in this paper achieved an accurate fitting of data points when using 40 knots, while the GWO method still cannot solve the cusp fitting when there are 40 knots. We guess that the use of heavy knots may be ignored in GWO.

Table 6. BIC value and knot distribution of HPA with different knot number.

Inner Knot Number	MSE	BIC	Knot Distribution
$\kappa = 2$	2.737	446.37	0.311, 0.372
$\kappa = 3$	1.015	268.32	0.282, 0.528, 0.604
$\kappa = 4$	0.756	230.31	0.192, 0.505, 0.505, 0.505
$\kappa = 5$	0.735	245.76	0.152, 0.472, 0.472, 0.472, 0.717
$\kappa = 6$	0.707	259.12	0.151, 0.476, 0.476, 0.476, 0.697, 0.819
$\kappa = 7$	0.643	261.50	0.230, 0.309, 0.470, 0.470, 0.530, 0.673, 0.785

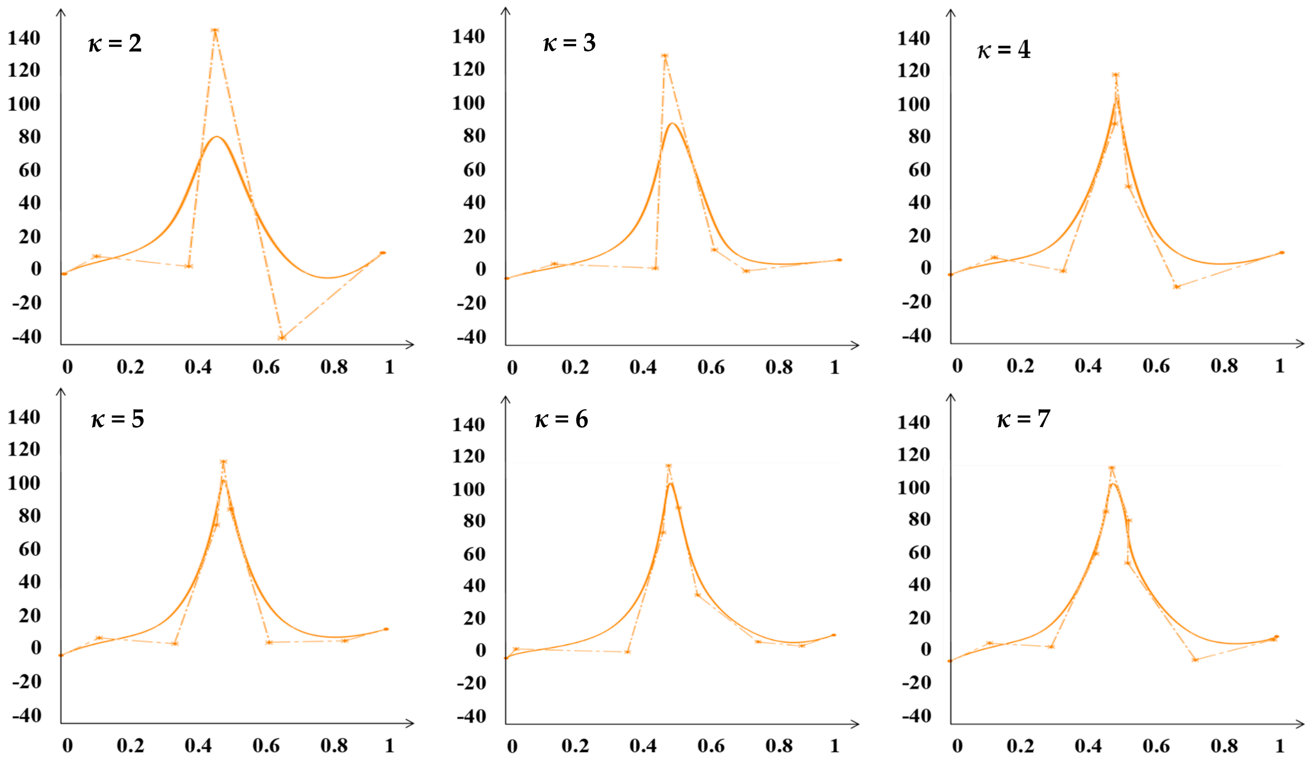


Figure 15. Fitting results of different number of knots.

5.2. Fitting of Ship Line

Figure 16 shows the offset table and ship line diagram of a ship. The ship has a total length of 189.98 m, a shape width of 32.26 m and a shape depth of 16 m.

Solidworks and Catia are popular design softwares at present. In particular, Catia has more prominent capabilities in complex surface design and reverse engineering. They are widely used in shipbuilding, aviation and other industries. Take the complex ship line on the cross section, longitudinal section and waterline surface and fit it through the existing design software Solidworks, Catia and HPA. Figure 17 compares the maximum error of Solidworks, Catia and HPA fitting curves under the condition of the same number of control points.

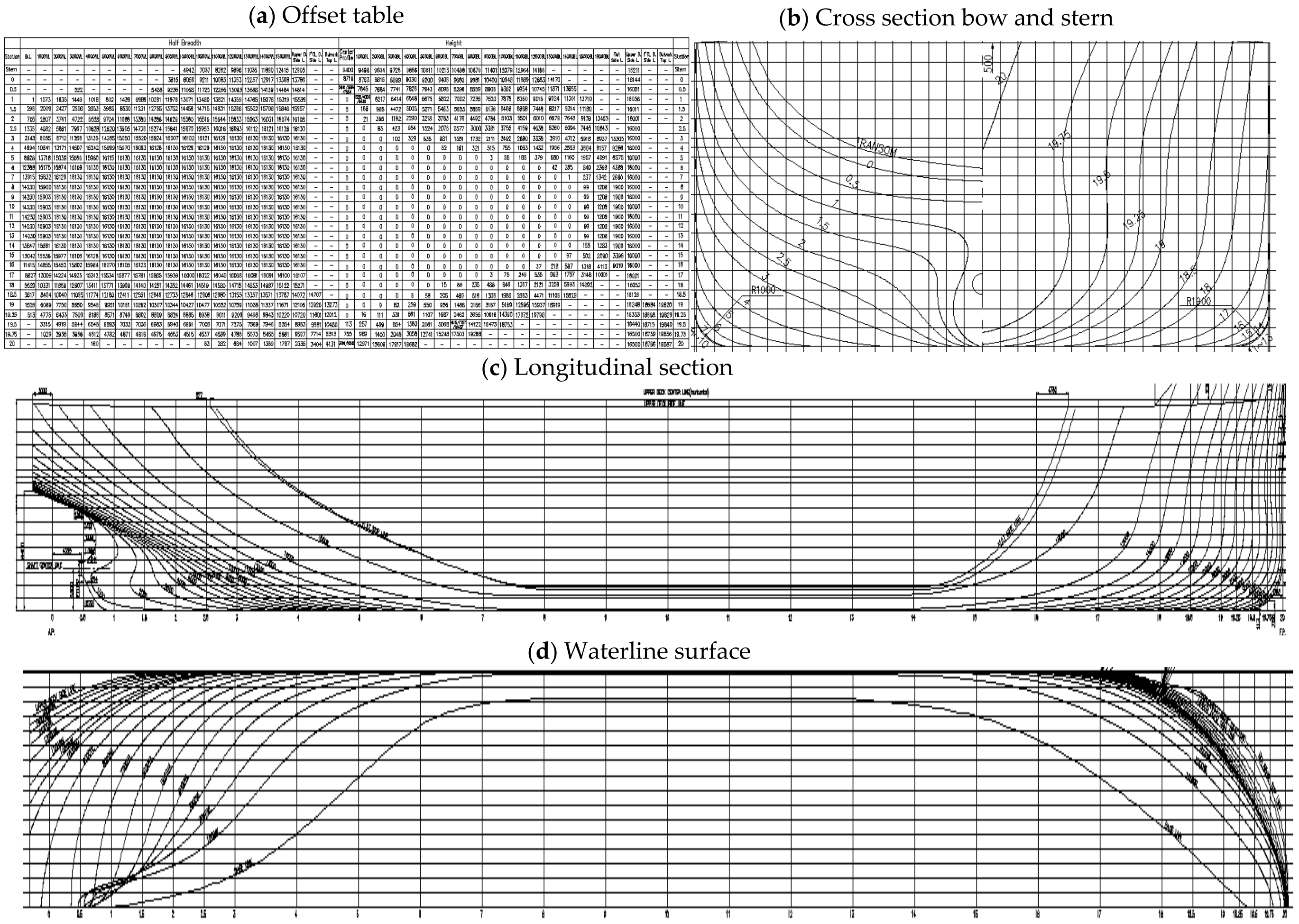

(**a**) Offset table
Half Breadth
Height
(**b**) Cross section bow and stern
TRANSOM
R1000
R1900
(**c**) Longitudinal section
(**d**) Waterline surface

Figure 16. (**a**) Offset table and ship line diagram, which include (**b**) cross section bow, (**c**) longitudinal section and (**d**) waterline surface.

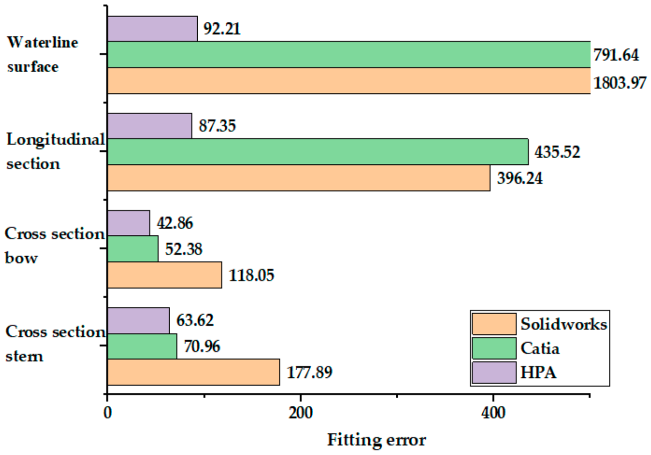


Figure 17. Maximum error of fitting using Solidworks, Catia and HPA at the same number of control points.

Input the same data points into Solidworks and Catia and use 10 control points to fit the ship line of the cross section stern, 6 control points to fit the cross section bow, 14 control points to fit the longitudinal section and 12 control points to fit the waterline surface. From the fitting results, it can be seen that under the same number of control points, Catia's fitting accuracy is higher than Solidworks, but both of their overall fitting results are less effective, especially in the part with continuous curvature change. Given the same data points and the same number of control points, the HPA proposed in this paper has higher precision and can better represent the shape of the curve than Solidworks and Catia.

Figure 18 records the fitting of the ship line of cross section bow, cross section stern, longitudinal section and waterline surface by Solidworks, Catia and HPA. We test the number of control points required by different methods when the fitting accuracy is similar. Figure 19 shows the fitting results with the same number of control points described in Figure 17.

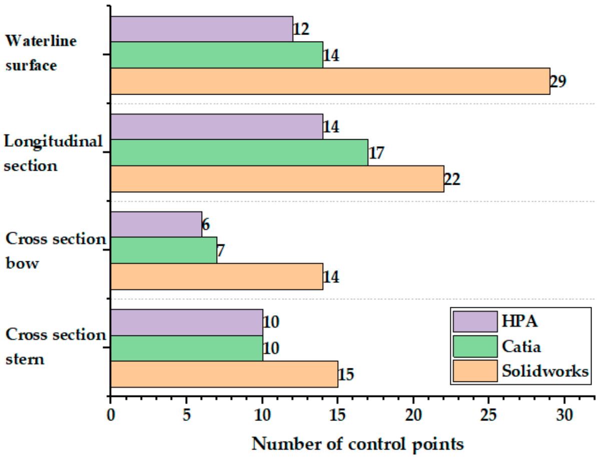


Figure 18. Number of control points when the maximum single point error is within 0.1 m.

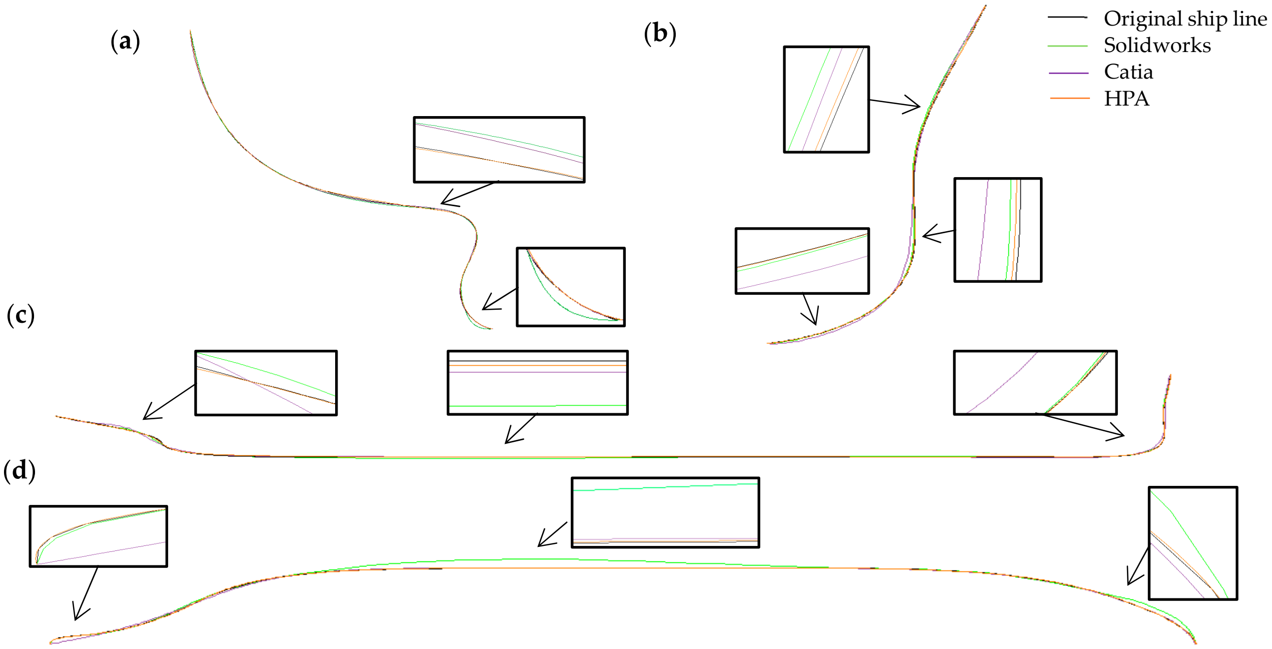


Figure 19. Fitting results of ship line of (**a**) cross section stern, (**b**) cross section bow, (**c**) longitudinal section and (**d**) waterline surface with Solidworks, Catia and HPA.

Through the fitting results, it can be seen that under the input of the same data points and the approximate fitting accuracy, compared with the existing design software Solidworks and Catia, HPA can complete the fitting with the same effect with fewer control points, which better proves the superiority of the method proposed in this paper.

6. Discussion and Conclusions

This paper develops a B-spline curve fitting of hunger predation optimization on ship line design (HPA). In the previous dynamic knot method, the initial knot is randomly selected from the interval [0, 1], which causes many unreasonable knot vectors. Therefore, a large population size and iteration number are required, which leads to a decrease in the efficiency of the algorithm. The knot guidance technology is designed to add knots in the area with complex model shape at the initial knot selection stage. This aims at the problem that the population loses diversity prematurely in the optimization algorithm, which leads to slow convergence and easy to fall into local optimality. A hunger algorithm search strategy is developed to make the hungry individuals in the population size appear near the optimal solution more quickly, and the influence of the neighbors in the population size on their position adjustment is further considered. This aims at the problem that the previous dynamic knot method requires manual adjustment of key input parameters such as population size and iteration number, which is troublesome and time consuming. An adaptive adjustment of key input parameters in HPA algorithm is proposed, which can quickly adapt to the replacement of model and fitting accuracy. We compared with the typical static knot method, dynamic knot method and the existing commercial Software Solidworks and Catia, and the feasibility and superiority of HPA algorithm, are verified. HPA achieves the goal of B-spline curve fitting with higher fitting accuracy at the same control points, less control points under the same accuracy requirements, faster operation efficiency and better universality. HPA can better solve practical engineering problems.

Future work includes extending HPA to B-spline surface fitting, the application of the hunger predation algorithm in other fields and further improving the fitting accuracy and fitting efficiency of B-spline curve fitting.

Author Contributions: Conceptualization, C.S. and M.L.; methodology, C.S. and M.L.; software, M.L.; validation, M.L. and S.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Unveiling and Commanding Science and Technology Project of Liaoning Province (No.2021JH1/10400093); National Natural Science Foundation of China (51305052); Key Laboratory for Precision & Non-traditional Machining of Ministry of Education of Dalian University of Technology, China (JMTZ202001).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data will be made available on reasonable request.

Acknowledgments: This work was supported by Unveiling and Commanding Science and Technology Project of Liaoning Province (No.2021JH1/10400093); National Natural Science Foundation of China (51305052); Key Laboratory for Precision and Non-traditional Machining of Ministry of Education of Dalian University of Technology, China (JMTZ202001).

Conflicts of Interest: The authors declare no conflict of interest, and there is no copyright issue in all of figures.

References

1. Duan, J.; Wang, B.; Cui, K.; Dai, Z. Path Planning Based on NURBS for Hyper-Redundant Manipulator Used in Narrow Space. *Appl. Sci.* **2022**, *12*, 1314. [CrossRef]
2. Gao, X.; Zhang, S.; Qiu, L.; Liu, X.; Wang, Z.; Wang, Y. Double B-Spline Curve-Fitting and Synchronization-Integrated Feedrate Scheduling Method for Five-Axis Linear-Segment Toolpath. *Appl. Sci.* **2020**, *10*, 3158. [CrossRef]
3. Liu, Y.; Wan, Z.; Yang, C.; Wang, X. NURBS-Enhanced Meshfree Method with an Integration Subtraction Technique for Complex Topology. *Appl. Sci.* **2020**, *10*, 2587. [CrossRef]
4. Li, X.; He, L. Shape Optimization Design for a Centrifuge Structure with Multi Topological Configurations Based on the B-Spline FCM and GCMMA. *Appl. Sci.* **2020**, *10*, 620. [CrossRef]
5. Ni, Y.; Liu, F.; Wu, Y.; Wang, X. Continuous-Time Fast Motion of Explosion Fragments Estimated by Bundle Adjustment and Spline Representation Using HFR Cameras. *Appl. Sci.* **2021**, *11*, 2676. [CrossRef]

6. Son, M.J.; Kim, T.W. Implementation of an executable business process management model for the ship hull production design process. *J. Mar. Sci. Technol.* **2014**, *19*, 170–184. [CrossRef]
7. Yi, L. Application of Web technology and non-uniform B-splines in hull line design system. *Ship Sci. Technol.* **2020**, *42*, 4–6.
8. Ma, J.; Xie, K.; Tong, X. Research and application of hull shape design software based on non-uniform B-splines. *Comput. Eng. Appl.* **2004**, *23*, 223–225.
9. Wang, P. Discussion on the smoothing method of hull line shape. *China Water Transp.* **2016**, *16*, 3–5+15.
10. Ma, W.; Kruth, J.P. Parameterization of randomly measured points for least squares fitting of b-spline curves and surfaces. *Comput. Aided Des.* **1995**, *27*, 663–675. [CrossRef]
11. Lin, H.; Chen, W.; Wang, G. Curve reconstruction based on an interval B-spline curve. *Vis. Comput.* **2005**, *21*, 418–427. [CrossRef]
12. Yang, Z.; Deng, J.; Chen, F. Fitting unorganized point clouds with active implicit B-spline curves. *Vis. Comput.* **2005**, *21*, 831–839. [CrossRef]
13. Jose, E.; Carlos, H.; Maria, D.; Juan, G.; Raul, E.; Horacio, R. Parallel Hierarchical Genetic Algorithm for Scattered Data Fitting through B-Splines. *Appl. Sci.* **2019**, *9*, 2336. [CrossRef]
14. Zhou, H.; Wang, Y.; Liu, Z.; Lu, B. Non-Uniform Rational B-Splines Curve Fitting Based on the Least Control Points. *J. Xi'an Jiaotong Univ.* **2008**, *42*, 73–77.
15. Zhao, D.; Guo, H. A Trajectory Planning Method for Polishing Optical Elements Based on a Non-Uniform Rational B-Spline Curve. *Appl. Sci.* **2018**, *8*, 1355. [CrossRef]
16. Hu, M.; Feng, J.; Zheng, J. An additional branch free algebraic B-spline curve fitting method. *Vis. Comput.* **2010**, *26*, 801–811. [CrossRef]
17. Yoshimoto, F.; Harada, T.; Yoshimoto, Y. Data fitting with a spline using a real-coded genetic algorithm. *Comput. Aided Des.* **2003**, *35*, 751–760. [CrossRef]
18. Sarfraz, M. Representing shapes by fitting data using an evolutionary approach. *Comput. Aided Des. A* **2013**, *1*, 179–185. [CrossRef]
19. Özkan, İ.; İsmail, K. Gray Wolf Optimizer for Knot Placement in B-Spline Curve Fitting. *Gaziosmanpasa J. Sci. Res.* **2017**, *6*, 97–109.
20. He, B.; Feng, R.; Yv, S. B-Spline Curve and Surface Fitting Using Differential Evolution Algorithm. *J. Graph.* **2016**, *37*, 178–183. [CrossRef]
21. Kübra, U.; Erkan, Ü. B-spline curve fitting with invasive weed optimization. *Appl. Math. Model.* **2017**, *52*, 320–340. [CrossRef]
22. Gálvez, A.; Iglesias, A.; Avila, A.; Otero, C.; Arias, R.; Manchado, C. Elitist clonal selection algorithm for optimal choice of free knots in B-spline data fitting. *Appl. Soft Comput.* **2015**, *26*, 90–106. [CrossRef]
23. Piegl, L.; Tiller, W. *Tiller, the NURBS Book*; Springer: Berlin/Heidelberg, Germany, 1997.
24. Shi, F. *CAGD&NURBS*; Higher Education Press: Beijing, China, 2013; pp. 57–59.
25. Kim, I. A development of data structure and mesh generation algorithm for whole ship analysis modeling system. *Adv. Eng. Softw.* **2006**, *37*, 85–96. [CrossRef]
26. Lee, H.-D.; Son, M.-J.; Oh, M.-J.; Lee, H.-W.; Ki, T.-W. Development of a simple model for batch and boundary information updation for a similar ship's block model. *Sci. China Inf. Sci.* **2012**, *55*, 1019–1031. [CrossRef]
27. Lee, D.; Lee, K.-H. An approach to case-based system for conceptual ship design assistant. *Expert Syst. Appl.* **1999**, *16*, 97–104. [CrossRef]
28. Yv, Y. *Research on Reconstruction and Deployment Technology of Ship Curved Surface Segmented Outer Plate*; Jiangsu University of Science and Technology: Jiangsu, China, 2016.
29. Várady, T.; Martin, R.R.; Cox, J. Reverse engineering of geometric models—An introduction. *Comput. Aided Des.* **1997**, *29*, 255–268. [CrossRef]
30. Piegl, L.; Tiller, W. Least–Square B–Spline Curve Approximation with Arbitrary End Derivatives. *Eng. Comput.* **2000**, *16*, 109–116. [CrossRef]
31. Lyche, T.; Morken, K. Knot removal for parametric B-Spline curves and surface. *Comput. Aided Geom. Des.* **1987**, *4*, 217–230. [CrossRef]
32. Lyche, T.; Morken, K. A date reduction strategy for splines. *IMA J. Numer. Anal.* **1988**, *8*, 185–208. [CrossRef]
33. Razdan, A. *Knot Placement for B-Spline Curve Approximation*; ASU: Phoenix, AZ, USA, 1999.
34. Park, H.; Lee, J.-H. B-spline curve fitting based on adaptive curve refinement using dominant points. *Comput. Aided Des.* **2007**, *39*, 439–451. [CrossRef]
35. Xu, J.; Ke, Y.; Qu, W. B-spline curve approximation technology based on automatic identification of feature points. *Chin. J. Mech. Eng.* **2009**, *45*, 212–217. [CrossRef]
36. Park, H.; Kim, K.; Lee, S.C. A method for approximate NURBS curve compatibility based on multiple curve refitting. *Comput. Aided Des.* **1995**, *32*, 237–252. [CrossRef]
37. Hamann, B.; Chen, J. Data point selection for piecewise linear curve approximation. *Comput. Aided Geom. Des.* **1994**, *11*, 289–301. [CrossRef]
38. Liu, G.; Wong, Y.; Zhang, Y.; Loh, H. Adaptive fairing of digitized data with discrete curvature. *Comput. Aided Des.* **2002**, *34*, 309–320. [CrossRef]
39. Li, W.; Xu, S.; Zhao, G.; Goh, L. Adaptive knot placement in b-spline curve approximation. *Comput. Aided Des.* **2005**, *37*, 791–797. [CrossRef]
40. Zhou, M.; Sun, S. *Genetic Algorithm: Theory and Applications*; National Defence Industry Press: Beijing, China, 1999. (In Chinese)

41. Storn, R.; Price, K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [CrossRef]
42. Seyedali, M.; Seyed, M.; Andrew, L. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
43. Zhang, X.; Wang, X. Review of Research on Grey Wolf Optimization Algorithm. *Comput. Sci.* **2019**, *46*, 30–38.
44. Seyedali, M.; Andrew, L. The Whale Optimization Algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [CrossRef]
45. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the IEEE International Conference, Perth, Australia, 27 November–1 December 1995; pp. 1942–1948.
46. Rashedi, E.; Nezamabadi-Pour, H.; Saryazdi, S. GSA: A gravitational search algorithm. *Inf. Sci.* **2009**, *179*, 2232–2248. [CrossRef]
47. Fatma, A.; Essam, H.; Mai, S.; Walid, A.; Seyedali, M. Henry gas solubility optimization: A novel physics-based algorithm. *Future Gener. Comput. Syst.* **2019**, *101*, 646–667. [CrossRef]
48. De Boor, C.; Rice, J.R. *Least Squares Cubic Spline Approximation II—Variable Knots*; Computer Science Technical Report; International Mathematical Statistical Libraries: Houston, TX, USA, 1968.
49. Ulker, E.; Arslan, A. Automatic knot adjustment using an artificial immune system for B-spline curve approximation. *Inf. Sci.* **2009**, *179*, 1483–1494. [CrossRef]
50. Ulker, E. B-Spline curve approximation using Pareto envelope-based selection algorithm-PESA. *Int. J. Comput. Commun.* **2013**, *2*, 60–63. [CrossRef]
51. Valenzuela, O.; Delgado-Marquez, B.; Pasadas, M. Evolutionary computation for optimal knots allocation in smoothing splines. *Appl. Math. Model.* **2013**, *37*, 5851–5863. [CrossRef]
52. Gálvez, A.; Iglesias, A. Firefly algorithm for explicit B-spline curve fitting to data points. *Math. Probl. Eng.* **2013**, *2013*, 528215. [CrossRef]
53. Yuan, Y.; Chen, N.; Zhou, S. Adaptive B-spline knot selection using multiresolution basis set. *IIE Trans.* **2013**, *45*, 1263–1277. [CrossRef]