Mathematics and Computer Science
2024, Vol. 9, No. 4, pp. 74-87
https://doi.org/10.11648/j.mcs.20240904.12

Research Article

Intelligent Design of Street Lamp in Rural Areas Based on an Improved Genetic Algorithm

Xianhao Deng[1], Qiancheng Tan[1], Hao Liu[1], Yubiao Long[1], Yonghui Qin[1, 2, 3, *]

[1]Guangxi Colleges and Universities Key Laboratory of Data Analysis and Computation, College of Mathematics and Computing Science, Guilin University of Electronic Technology, Guilin, P. R. China

[2]Center for Applied Mathematics of Guangxi (GUET), Guilin, P. R. China

[3]Guangxi Key Laboratory of Automatic Detecting Technology and Instruments (GUET), Guilin, P. R. China

Abstract

This study addresses the demand for more efficient streetlight designs in rural areas by introducing an improved genetic algorithm (GA) to optimize the geometry and placement of streetlight poles. Conventional GAs frequently suffer from premature convergence and becoming trapped in local optima, reducing their effectiveness. To mitigate these issues, this research integrates the genetic algorithm with Sequential Quadratic Programming (SQP), using the quasi-optimal solution generated by the GA as the initial input for the SQP, enhancing both accuracy and stability. The methodology includes developing a geometric model of streetlight poles utilizing point cloud data and extracting the centerline via the optimized GA-SQP approach. Additionally, the study examines the effects of random errors, gross errors, incomplete point cloud data, and centerline deviations on the algorithm's performance.

Keywords

Intelligent Street Lights, Centerline, Sequence Quadratic Programming Algorithm, Genetic Algorithm

1. Introduction

In recent years, street lights, as the mainstay of urban lighting, have undergone rapid development, and technological upgrades have made them more intelligent, convenient, and environmentally friendly. However, the traditional lighting industry still faces numerous issues, such as excessive and disorderly placement of light poles, occupying excessive land resources, low energy efficiency, and safety hazard [1, 2].

As a result, intelligent streetlights have emerged as a solution, and excellent intelligent streetlight design can bring some significant environmental benefits.

Genetic algorithm is a heuristic algorithm [3] that can simulate biological evolution and is widely used in optimization, scheduling, and transportation due to its practicality and robustness. However, it suffers from certain issues such as premature convergence and a local optimal solution [4-7].

To address these issues, researchers have proposed various improved genetic algorithms in recent years, combining ge-

*Corresponding author: yonghui1676@163.com (Yonghui Qin)

Received: 29 August 2024; **Accepted:** 18 September 2024; **Published:** 29 September 2024

netic algorithms with some sequential quadratic programming algorithms [8, 9]. These hybrid genetic algorithms have been widely used in optimization problems [10-12], such as footstep detection [13], online performance optimization [14], and computational chemistry [15, 16], and have achieved good results. The research results show that the improved genetic algorithm can significantly improve computational efficiency and obtain satisfactory optimization results.

The improved genetic algorithm is based on the strong search capability of the genetic algorithm in the solution space. It employs the "quasi-optimal solution" obtained by the genetic algorithm to serve as the initial solution for the sequential quadratic programming algorithm, iteratively searching for the global optimal solution or a close approximation.

For the above problem, this paper proposes an improved genetic algorithm to obtain a more energy-efficient, environmentally friendly, and convenient intelligent streetlamp design. Based on the point cloud characteristics of the lamp post, the model of the centerline is obtained, and the genetic algorithm is used to extract the centerline of the lamp post. The extraction of the centerline here is mainly based on the central coordinates of the upper and lower surfaces of the lamp post to generate the centerline, considering random errors, gross errors, and observed centerlines, and combining actual conditions to obtain a complete algorithmic process for designing rural streetlamps. Additionally, using the point cloud features of the lamp post, a centerline model is established through statistical data. An improved mixed genetic algorithm is proposed by using the sequential quadratic programming algorithm to improve the genetic algorithm. The genetic algorithm scheme is established by extracting the centerline of the lamp post, considering the relationship between random errors, gross errors, observed centerlines, and the algorithm, and through experimental results, the superiority of the improved genetic algorithm is demonstrated.

2. Model and Algorithm Implementation

2.1. Model and Algorithm for Street Light Centerline Extraction

Point cloud data is typically obtained with 3D imaging sensors such as stereo cameras, 3D scanners, and RGB-D cameras. Popular devices for capturing point cloud data include RGB-D cameras, Intel's RealSense, and the Structure Sensor. Point cloud data can be created by scanning the intrinsic parameters of the scanning camera or by scanning images obtained from an RGB-D camera. Additionally, LiDAR laser detection experiments can be used to obtain point cloud data, primarily through satellite, airborne, and ground-based methods.

In this paper, the point cloud data collection method involves extracting the required point cloud data by establishing a geometric model like that of a streetlamp. This approach can obtain the necessary point cloud data without relying on external sensors or equipment.

The coordinates of the center point of the lower surface of the specified streetlamp are (x_1, y_1, z_1), with a radius of r_1, and the coordinates of the center point of the upper surface are (x_2, y_2, z_2), with a radius of r_2. The designed parameter solution is $(x_1, y_1, z_1, x_2, y_2, z_2, r_1, r_2)$. z_1 and z_2 can be determined by the minimum and maximum z-coordinates in the surface point cloud data.

The fitness function is represented by the central axis equation of the truncated cone and streetlamp model determined by the parameter solution $(x_1, y_1, z_1, x_2, y_2, z_2, r_1, r_2)$, as well as the sum of squared fitting residuals of the observed point cloud data.

$$Q = \sum_{p=1}^{n} \left(r_p - d_p\right)^2 = min \tag{1}$$

$$M: \frac{x-x_1}{x_2-x_1} = \frac{y-x_1}{y_2-y_1} = \frac{z-z_1}{z_2-z_1} \tag{2}$$

In equation (1), r represents the radius of the truncated cone corresponding to point p, in millimeters. d represents the distance between point p and the central axis of the truncated cone, in millimeters. Q represents the sum of squared fitting residuals. M represents the central axis equation of the truncated cone model.

The experimental procedure is as follows:

1) Solve for the parameter solution $(x_1, y_1, z_1, x_2, y_2, z_2, r_1, r_2)$ of the two-point spatial line equation (central axis equation of the truncated cone model)
2) Compute the distance d between point p on the surface of the truncated cone and its central axis.
3) Interpolate the radius of the truncated cone corresponding to point p using the upper and lower radii (r_2 and r_1) of the truncated cone.
4) For all surface point cloud data, repeat steps 2-3 and accumulate the fitting residuals using equation (1).
5) Use a genetic algorithm to iteratively search for a parameter solution ($x_1^k, y_1^k, z_1^k, x_2^k, y_2^k, z_2^k, r_1^k, r_2^k$) that minimizes the sum of squared fitting residuals Q.
6) Use the result from step 5 as the initial value for a second-order sequence optimization to further optimize the parameter solution.

By utilizing the algorithm, we obtain the optimal centers of the upper and lower surfaces of the truncated cone, and then complete the calculation of the spatial line equation using the two-point formula. To assess the effectiveness of the improved genetic algorithm, we introduce a numerical indicator, the positional mean error, to measure point positioning accuracy. In the present context, the positional mean error can be calculated using the following formula:

$$M = \pm\sqrt{\frac{\sum_{i=1}^{n} \Delta_i^2}{2n}} \tag{3}$$

where $\sum_{i=1}^{n} \Delta_i^2$ represents the objective function, and $2n$ represents point cloud quantity.

2.2. Methodology and Algorithms

Individual encoding: The initial population is generated randomly based on the parameter range. According to extensive parameter optimization experience, the chromosome gene dimension is set to 6. In this case, the chromosome genes are represented as $(x_1^k, y_1^k, x_2^k, y_2^k, r_1^k, r_2^k)$. The genetic algorithm encoding method is real-number encoding, which has the advantages of wide applicability and high precision. The maximum number of genetic generations is denoted as s, the crossover probability as p, and the mutation probability as q.

Selection of the initial population: Genetic algorithms involve a series of evolutionary operations on populations, so it is necessary to prepare a certain amount of initial population data, denoted as N. Individuals in the population can be generated randomly.

crossover and mutation: crossover is performed according to an elitist scheme, and mutation is carried out using a roulette wheel method, after the initial population is generated and individual fitness is calculated. Fitness is represented by the sum of squared fitting residuals Q observed from the point cloud data. The mutated offspring population is merged with the parent population, sorted based on their fitness values, and the top N individuals are selected as the new population for the next generation of genetic operations.

In the selection and crossover operations, we utilize an elitist ranking scheme known as the "monarch" method, which is characterized by its faster convergence speed. Although genetic algorithms cannot guarantee that every individual in the offspring is better than every individual in the parent population, they can guarantee that the best individual in the offspring is better than all individuals in the parent population. The roulette wheel method is used to select individuals for mutation based on their fitness values relative to the total fitness of the population [17]. The combination of the roulette wheel method and the monarch scheme plays an important role in improving the global convergence of genetic algorithms.

Sequential Quadratic Programming (SQP) is an algorithm that transforms complex nonlinear optimization problems into simpler quadratic programming problems. The basic idea is to simplify the objective function of the nonlinear optimization problem at the iteration point into a quadratic function using a Taylor expansion, while simplifying the constraint function that can be expressed as linear equations or inequalities into a linear function.

Considering the nonlinear constrained optimization problem:

$$\min f(x) \ \text{(s. t. } h_i(x) = 0, i \in E = \{1,2,3,\dots,l\}.) \qquad (4)$$

where $f: R^n \to R, h_i: R^n \to R (i \in E)$ are twice continuously differentiable real functions. Then get the Lagrangian function of problem (4)

$$L(x,\mu) = f(x) - \sum_{i=1}^{l} \mu_i h_i(x) = f(x) - \mu^T h(x), \quad (5)$$

where $\mu = (\mu_1, \dots, \mu_l)^T, h(x) = \left(h_1(x), \dots, h_l(x)\right)^T$, and the gradient of the constraint function $h(x)$:

$$\nabla h(x) = [\nabla h_1(x), \dots, \nabla h_l(x)],$$

then the Jacobian matrix of $h(x)$ is $A(x) = \nabla h(x)^T$.

Combining with the Karush-Kuhn-Tucker (KKT) conditions [18] of problem (4), we can obtain:

$$\nabla L(x,\mu) = \begin{bmatrix} \nabla_x L(x,\mu) \\ \nabla_x L(x,\mu) \end{bmatrix} = \begin{bmatrix} \nabla f(x) - A(x)^T \mu \\ -h(x) \end{bmatrix} = 0. \quad (6)$$

Using the Newton method to solve the nonlinear equation system formula (6), it can be transformed into a strictly convex quadratic programming problem (7):

$$min\ q_k(d) = \frac{1}{2} d^T B(x_k, \mu_k) d + \nabla f(x_k)^T d \ \text{(s. t. } h(x_k) + A(x_k) d = 0,) \qquad (7)$$

where $B(x_k, \mu_k)$ is a positive definite $n \times n$ matrix. $A(x_k)$ is a full-rank $l \times n$ matrix, and d_k is the global minimum point. Meanwhile, the penalty function is defined:

$$p(x,\mu) = \|\nabla L(x,\mu)\|^2 = \|\nabla f(x) - A(x)^T \mu\|^2 + \|h(x)\|^2 \quad (8)$$

$$\nabla p(x_k, \mu_k)^T p_k = -2p(x_k, \mu_k) \le 0 \qquad (9)$$

The specific algorithm steps are as follows:

Step 1: Choose $x_0 \in R^n, \mu_0 \in R^l, \rho, \gamma \in (0,1)$. Set $k = 0$ and $0 \le \varepsilon << 1$.

Step 2: Calculate the value of $p(x_k, \mu_k)$. If $p(x_k, \mu_k) \le \varepsilon$, stop. Otherwise, proceed to Step3.

Step 3: Solve the quadratic programming subproblem to obtain d_k and $\bar{\mu}_k$, and let v_k=$\bar{\mu}_k$-μ_k-$\frac{1}{2\tau}A(x_k)d_k$.

Step 4: If $p(x_k + d_k, \mu_k + v_k) \le (1-\gamma) p(x_k, \mu_k)$, set $\alpha_k = 1$ and go to Step 6. Otherwise, proceed to Step 5.

Step 5: Let m_k be the smallest non-negative integer that satisfies the inequality:

$$p(x_k + \rho^m d_k, \mu_k + \rho^m v_k) \le (1 - \gamma\rho^m) p(x_k, \mu_k) \ \text{set} \ \alpha_k = \rho^{m_k} \qquad (10)$$

Step 6: Let $x_{k+1} = x_k + \alpha_k d_k, \mu_{k+1} = \mu_k + \alpha_k v_k$, k=k+1, and go back to Step 1.

2.3. Introduction to Data Perturbations

Point cloud data with random errors

It is not difficult to find that many things are accompanied by random errors. In the measurement process, errors with mutually offsetting characteristics are often formed due to a series of related small factors. For this experiment, the same situation exists, so it is necessary to process the point cloud data collected. Currently, there are several methods to generate random numbers in MATLAB, including:

1) *randi*(): to produce uniformly distributed random integers;
2) *rand*(): to produce uniformly distributed random numbers;
3) *unifrnd*(): to produce continuously uniform random numbers;
4) *unidrnd* (): to produce discrete uniform random numbers.

Considering the specific situation of this experiment, *rand*() can produce pseudo-random numbers with a uniform distribution, which is more suitable for the current data needs. The point cloud data generated by using the pseudo-random numbers generated by the rand function are perturbed with random errors of 1mm, 3mm, and 5mm, respectively, in the X and Y coordinates. The stability and accuracy of the results are observed by running the experiment multiple times, and the impact of random errors is analyzed.

Outlier point cloud data

During data collection, observers may inadvertently introduce gross errors, which are erroneous results or out-of-range errors caused by observer negligence, such as sighting errors, reading errors, and recording errors. The presence of gross errors can greatly affect the reliability of adjustment results, and even lead to completely wrong results. If incorrect data is collected, it will have a significant impact on the experimental results.

Half-space point cloud data

In practical situations, it is common to encounter scenarios where only a portion of the data is collected due to environmental factors or personal reasons, such as when using instruments to collect data on terrain and landforms. In such cases, it is necessary to understand the impact of the incomplete data on the experiment.

To study the impact of incomplete data, one approach is to use the half-space point cloud. For example, in the case of a streetlamp, the Y-axis of half of the point cloud data can be used to calculate the center. The resulting error can be analyzed to evaluate the impact of incomplete data on the algorithm, as well as to assess the effectiveness of the algorithm's error-correction capabilities.

Centerline offset

In practical situations, when collecting data on objects such as poles, the centerline of the pole may not always be aligned with the observer's line of sight. This can occur due to environmental factors, such as uneven terrain or obstructions, or human factors, such as the observer's position or orientation. As a result, the collected point cloud data of the frustum may be affected by the offset centerline. This can lead to inaccuracies in the reconstruction or modeling of the object and can affect the performance of algorithms used to analyze or manipulate the point cloud data.

When the centerline is offset, the observed centerline is not aligned with the observer's line of sight, meaning that the observed centerline and the centerline of the pole are not aligned. To study the impact of this situation on the experiment, we can make the following assumption: by changing the center coordinates, we can offset the X-axis of the centerline of the pole by 150 units. We can then use an improved genetic algorithm to determine whether a good fit can be obtained with the resulting centerline. The impact of the offset centerline on the algorithm can then be evaluated.

To evaluate the impact of the offset centerline, the width of the field of view can be set to ten times that of the pole, and the centerline observed at a non-central position can be used to collect point cloud data. Genetic algorithms and improved genetic algorithms can be compared to evaluate the impact of the offset centerline on the performance of the algorithm.

3. Genetic Algorithm and Improved Genetic Algorithm Simulation

3.1. Definition of Symbols

Table 1. *Symbol explanation for simulation experiment.*

X1:	The coordinate on the X-axis of the bottom surface of a frustum
Y1:	The coordinate on the Y-axis of the bottom surface of a frustum
R1:	The radius of the bottom surface of a frustum
X2:	The coordinate on the X-axis of the top surface of a frustum
Y2:	The coordinate on the Y-axis of the top surface of a frustum
R2:	The radius of the top surface of a frustum
K:	The number of iterations of an improved genetic algorithm
VAL:	The accuracy of an improved genetic algorithm
ANS:	Positional mean error
Q:	Random error introduced in experiments
W:	Proportion of gross errors introduced in experiments

3.2. Construction of Geometric Model

The main task in extracting the centerline of streetlamps is to extract the centerline equation based on point cloud data. Point cloud refers to a collection of points, and compared to images, point cloud has an irreplaceable advantage - deep 3D point cloud can directly provide 3D spatial data, while images require perspective geometry to infer 3D data. Point clouds can be divided into two types based on their composition characteristics: ordered point clouds and unordered point clouds.

To extract the centerline of streetlamps, a set of center coordinates and radii for the upper and lower surfaces of a frustum need to be constructed to closely match the actual surface point cloud frustum.

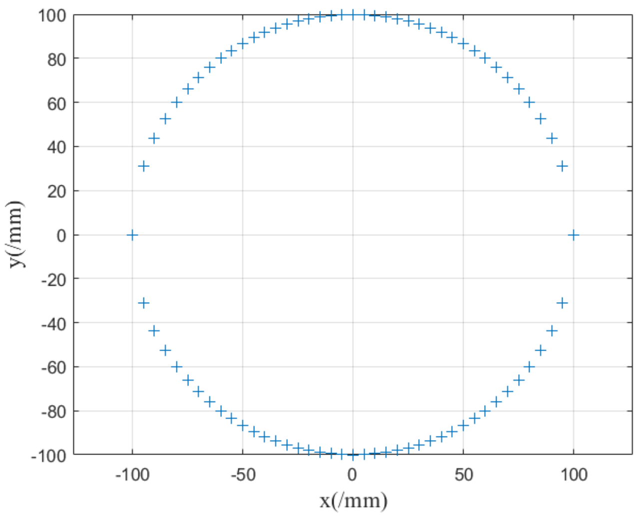

Figure 1. *Bottom cross-sectional point cloud diagram.*

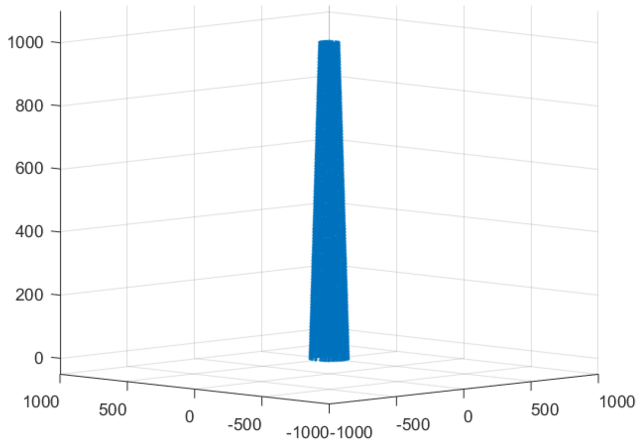

Figure 2. *Surface point cloud illustration of frustum.*

Specifically, the center coordinate of the bottom surface is (0, 0, 0), with a radius of 100mm; the center coordinate of the top surface is (0, 0, 1000), with a radius of 50mm. In the Z direction, a section is set every 10mm, and a total of 101 sections are set. In each section, two coordinate points are generated every 5mm in the X direction, as shown in Figure 1. The surface point cloud of the frustum is generated using the spatial equation of the frustum combined with MATLAB, as shown in Figure 2. Based on these point cloud data, a specific algorithm or method can be used to extract the centerline equation of the frustum for the purpose of extracting the centerline of streetlamps.

3.3. Experimental Results

Simulation results of the algorithm's effectiveness and stability:

In this example, the maximum genetic generation number s is set to 100, crossover probability p is set to 0.8, mutation probability q is set to 0.1, and the initial population size N is set to 100. The XY coordinates are confined to a square area with sides of length 2000 units, centered at the origin of the two-dimensional plane, while the radius R is within the range of (0, 200). A total of 100 initial population individuals are randomly generated.

Due to the high computational complexity of the objective function in sequential quadratic programming when used in the genetic algorithm, a "roulette wheel" selection method is used to randomly select 100 points from the point cloud data to form the objective function for the sequential quadratic programming algorithm. Additionally, to facilitate the program's calculations, an inequality constraint R1 > R2 is added.

From Table 2, the optimal solution $(x_1, y_1, z_1, x_2, y_2, z_2, r_1, r_2)$ obtained by the genetic algorithm differs from the input value in the experiment (0, 0, 0, 0, 0, 1000, 100, 50), and the selected optimal solution is not unique, with results exhibiting fluctuations. The reason for this is that in the experiment, the initial population of the genetic algorithm is established using randomly generated point cloud data, which has random characteristics within a certain range. This leads to different results in the optimal solution obtained through the genetic algorithm, reflecting the limitations of the genetic algorithm.

Table 2. *Simulation results of genetic algorithm experiment.*

	X1	Y1	R1	X2	Y2	R2
Input values	0.0000	0.0000	100.0000	0.0000	0.0000	50.0000
	5.2147	0.0366	99.0285	-4.8141	1.8253	50.5692
Simulation results	0.0887	-4.0183	100.0362	4.1909	-3.9940	50.4307
	0.7458	4.4898	100.1389	-2.3314	-3.2689	49.7842

As shown in Table 3, the improved genetic algorithm using sequential quadratic programming obtains an optimal solution that is consistent with the input value in the experiment. Compared with the genetic algorithm, the improved genetic algorithm has superior performance and achieves better fitting results.

Table 3. *Simulation result of improved genetic algorithm experiment.*

	X1	Y1	R1	X2	Y2	R2
Input values	0.0000	0.0000	100.0000	0.0000	0.0000	50.0000
Simulation result	-0.0000	-0.0000	100.0000	0.0000	0.0000	50.0000

Table 4. *Accuracy of improved genetic algorithm.*

K	VAL	ANS
11	1.7983e-16	8.6875e-10

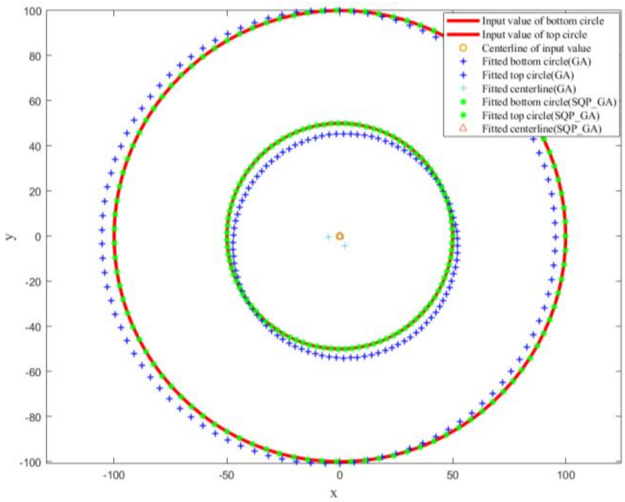

Figure 3. *Centerline fitting of top and bottom circles of two algorithms.*

From Table 3 and Table 4, the improved genetic algorithm obtains a highly accurate final solution, and the point mean error in the experiment also shows that the improved genetic algorithm has superior performance and high fitting accuracy. In conclusion, if only genetic algorithm is used, the results obtained will exhibit fluctuations and will not be unique. However, if the genetic algorithm is improved by adding sequential quadratic programming, the resulting algorithm will have stability and can suppress the fluctuations of the genetic algorithm. This leads to more stable results and higher precision. Figure 3 and Figure 4 clearly demonstrate the effectiveness of the improved algorithm.

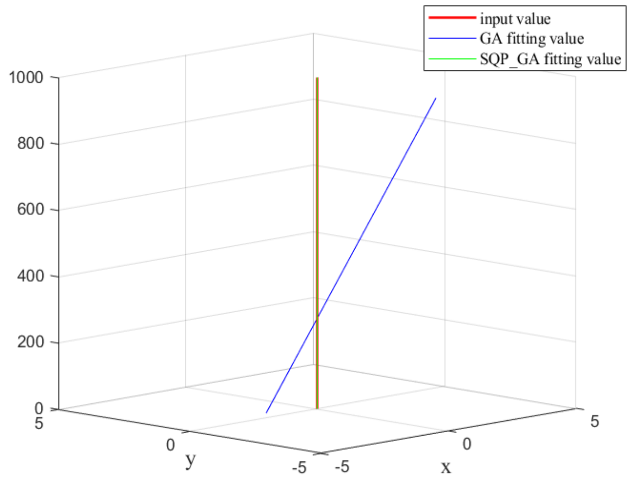

Figure 4. *Centerline fitting results of two algorithms.*

Simulation results of the algorithm under random error

As shown in Table 5 and Table 6, it can be observed that for both the genetic algorithm and the improved genetic algorithm using sequential quadratic programming, the fitting error of the obtained results increases as the random errors added to X and Y increase, and the fitting performance becomes worse. However, it is also observed that the fitting error of the improved genetic algorithm is much smaller than that of the genetic algorithm, and the fitting performance is significantly improved. Moreover, from the results of multiple fittings, it is found that as the random errors introduced in the experiment increase, the change in the X and Y coordinates of the upper and lower surface coordinates of the frustum fitted by the improved genetic algorithm is much smaller than that of the genetic algorithm.

Table 5*. Random errors of genetic algorithm.*

Q	X1	Y1	R1	X2	Y2	R2	ANS
1mm	-1.5609	1.7677	99.9794	-1.3074	2.3328	50.2272	1.3722
3mm	1.3887	0.0519	99.3735	-14.6245	-1.7772	51.5124	3.3427
5mm	-1.3799	-8.5345	100.6289	2.6409	3.7319	49.4890	3.4162

Table 6*. Random errors of improved genetic algorithm.*

Q	X1	Y1	R1	X2	Y2	R2	ANS
1mm	-0.1489	0.1644	99.9508	0.0294	0.0645	49.9980	0.4199
3mm	0.1994	-0.0431	100.0348	0.1332	0.2440	50.2905	1.2302
5mm	0.6580	-1.1303	99.7183	-1.7397	-0.2109	49.5453	2.1156

From Table 7, the improved genetic algorithm significantly outperforms the genetic algorithm in terms of fitting accuracy. Although the deviation caused by random errors is unavoidable, the error of the improved genetic algorithm is less than 3 when the random error is no greater than 5mm. Therefore, given the application of genetic algorithms, this improved algorithm can still be used effectively. The specific effectiveness is demonstrated in Figure 5 and Figure 6.

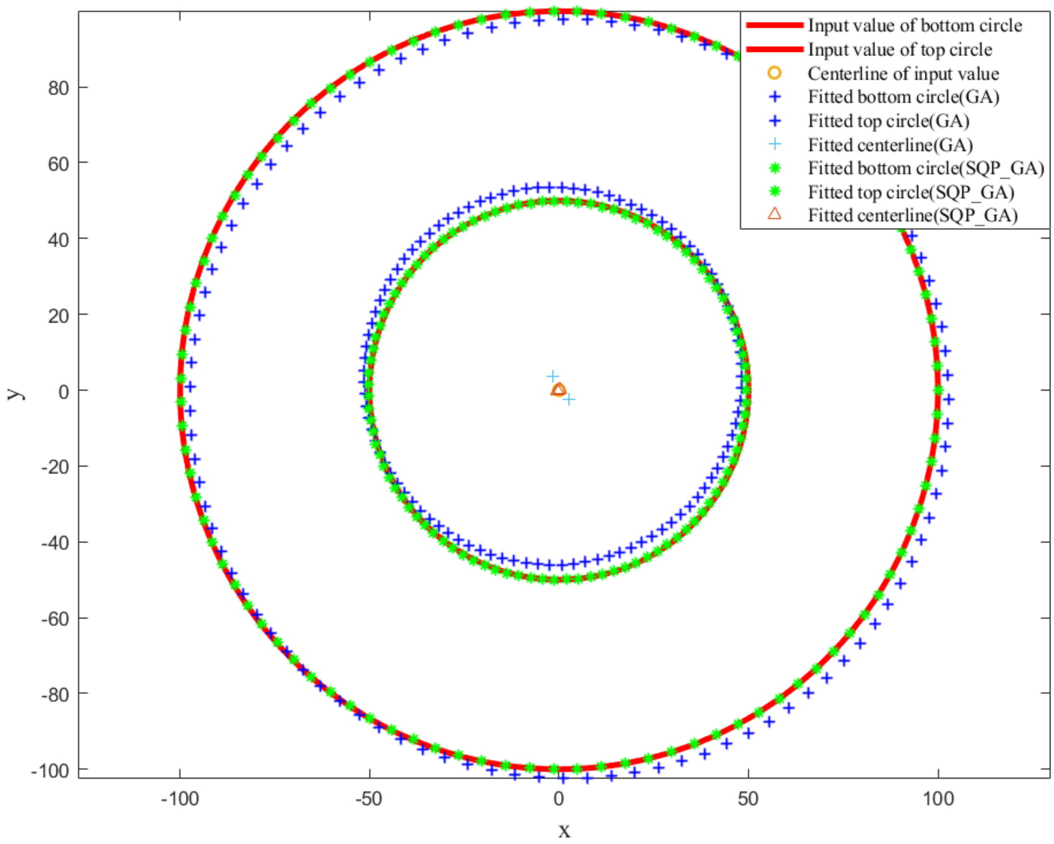

Figure 5*. Centerline fitting of top and bottom circles under random error.*

Table 7. Accuracy of improved genetic algorithm under random error.

Q	K	VAL	ANS
1mm	11	31.4123	0.4199
3mm	11	328.5819	1.2302
5mm	11	729.7239	2.1156

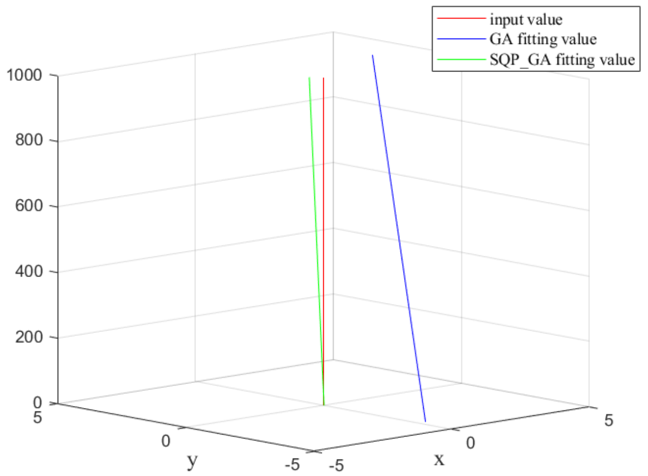

Figure 6. Centerline fitting under random error.

Simulation results of the algorithm under gross error:

From Table 8 and Table 9, it can be observed that when gross errors of 2%, 4%, and 6% of the total point cloud are added, the genetic algorithm's fitting accuracy of the upper and lower surface coordinates and the fitting performance decreases with the increase of the gross error proportion. The fitting performance drops significantly at around 6%, and the difference between the coordinates of the upper and lower surface and the input value becomes significant. Therefore, the genetic algorithm can only be used when the proportion of gross errors added is not more than about 3%.

For the improved genetic algorithm using sequential quadratic programming, the coordinates of the upper and lower surface and the fitting error remain relatively stable, even when the proportion of gross errors added increases. Although both and improved genetic algorithms cannot completely avoid an increase in error with increasing gross errors, the improved genetic algorithm produces a better fitting performance and can tolerate a higher proportion of gross errors compared to the genetic algorithm. It has some resistance to gross errors in the point cloud data, making it more robust in practical applications.

Table 8. Gross errors in genetic algorithm.

W	X1	Y1	R1	X2	Y2	R2	ANS
2%	-1.1139	-3.1750	100.1714	2.7430	-1.3544	49.9823	1.7395
4%	0.2567	2.6333	99.6372	-10.7467	-8.8703	50.7028	3.5515
6%	39.5872	-1.2449	97.1781	-72.2938	-1.4992	69.8942	12.9626

Table 9. Gross errors in improved genetic algorithm.

W	X1	Y1	R1	X2	Y2	R2	ANS
2%	0.5477	0.1255	100.1738	-0.0216	0.0211	50.0059	0.9159
4%	0.1679	0.2075	100.2243	0.2074	0.2959	49.2169	1.3214
6%	0.4990	-0.7683	99.1017	0.0161	1.2245	50.9122	1.5069

As shown in Table 10, the addition of gross errors has a certain impact on the accuracy and error of the improved genetic algorithm using sequential quadratic programming. However, its fitting error is much smaller than that of the genetic algorithm, even after the addition of gross errors. Therefore, the improved genetic algorithm is still effective and has a certain resistance to interference from gross errors.

The effectiveness of the improved algorithm is further illustrated in Figure 7 and Figure 8, which show the fitting results of the frustum using the genetic algorithm and the improved genetic algorithm under the influence of gross errors, respectively. The figures demonstrate that the improved genetic algorithm still produces a much better fit than the genetic algorithm, with lower error and improved stability, even in the presence of gross errors.

Table 10. Accuracy of improved genetic algorithm under gross error.

W	K	VAL	ANS
2%	10	132.0368	0.9159
4%	11	368.7035	1.3214
6%	11	549.4276	1.5069

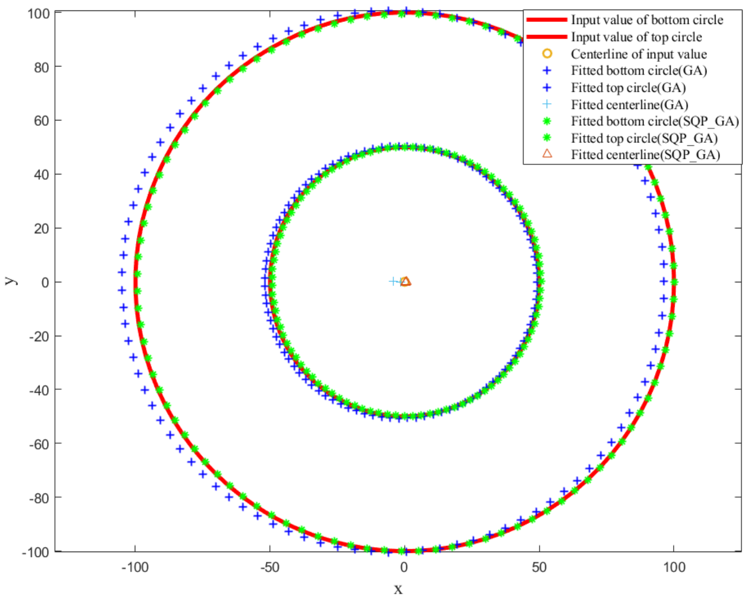

Figure 7. Centerline fitting of top and bottom circles under gross error.

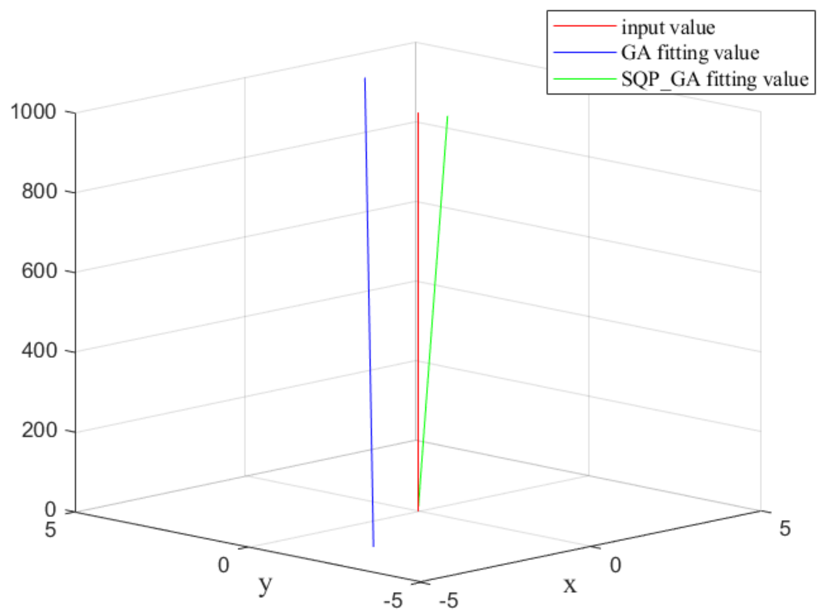

Figure 8. Centerline fitting under gross error.

Simulation results of the algorithm under data loss:

In this experiment, only the positive Y-axis will be selected as the initial point cloud data population, as shown in Figure 9 below:

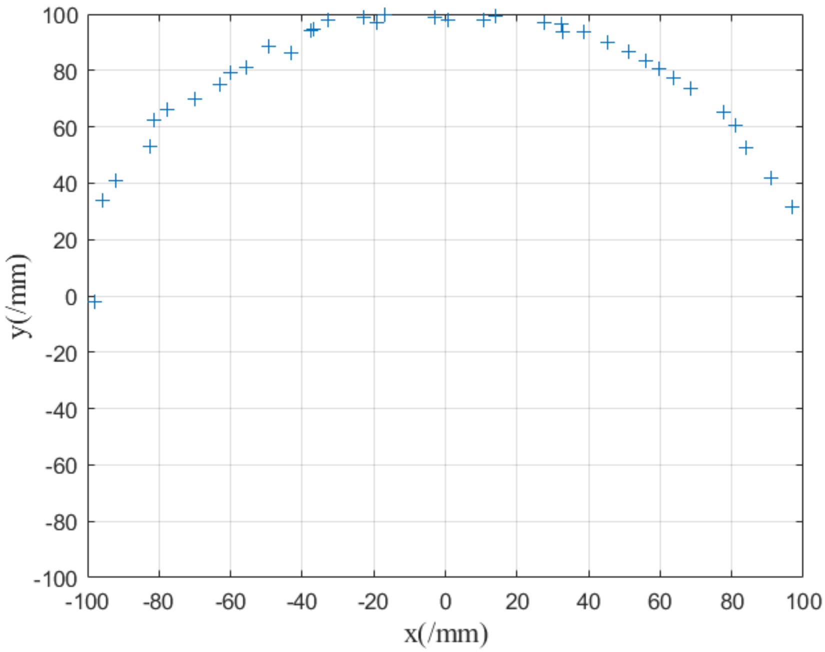

Figure 9. *Schematic diagram of bottom section point cloud.*

When half of the point cloud data is removed and random errors are introduced, the resulting data will be as follows:

Table 11. *Simulation results of improved genetic algorithm under random error in comprehensive point clouds.*

Q	X1	Y1	R1	X2	Y2	R2	ANS
1mm	-0.1489	0.1644	99.9508	0.0294	0.0645	49.9980	0.4199
3mm	0.1994	-0.0431	100.0348	0.1332	0.2440	50.2905	1.2302
5mm	0.6580	-1.1303	99.7183	-1.7397	-0.2109	49.5453	2.1156

Table 12. *Simulation results of improved genetic algorithm under random error in half-space point clouds.*

Q	X1	Y1	R1	X2	Y2	R2	ANS
1mm	0.1151	0.1845	99.8847	0.0314	-0.0214	50.2065	0.4185
3mm	0.4911	-0.8037	101.2339	-0.4183	0.8884	48.5907	1.2615
5mm	0.1104	1.9355	99.1000	0.3221	-6.1120	54.8055	2.1105

Table 11 and Table 12 show that using the improved genetic algorithm to experiment with comprehensive and half point clouds results is almost no change in the point error. Based on the data collected and the generated fitting images, when the entire surface of the object cannot be observed, the impact of using the genetic algorithm with the sequence quadratic programming algorithm added is very small. The influence of incomplete observation can be ignored for both traditional and improved genetic algorithms. Both algorithms can adapt well to the situation where the point cloud distribution on the surface is not comprehensive. The specific fitting effect is shown in Figure 10 and Figure 11.

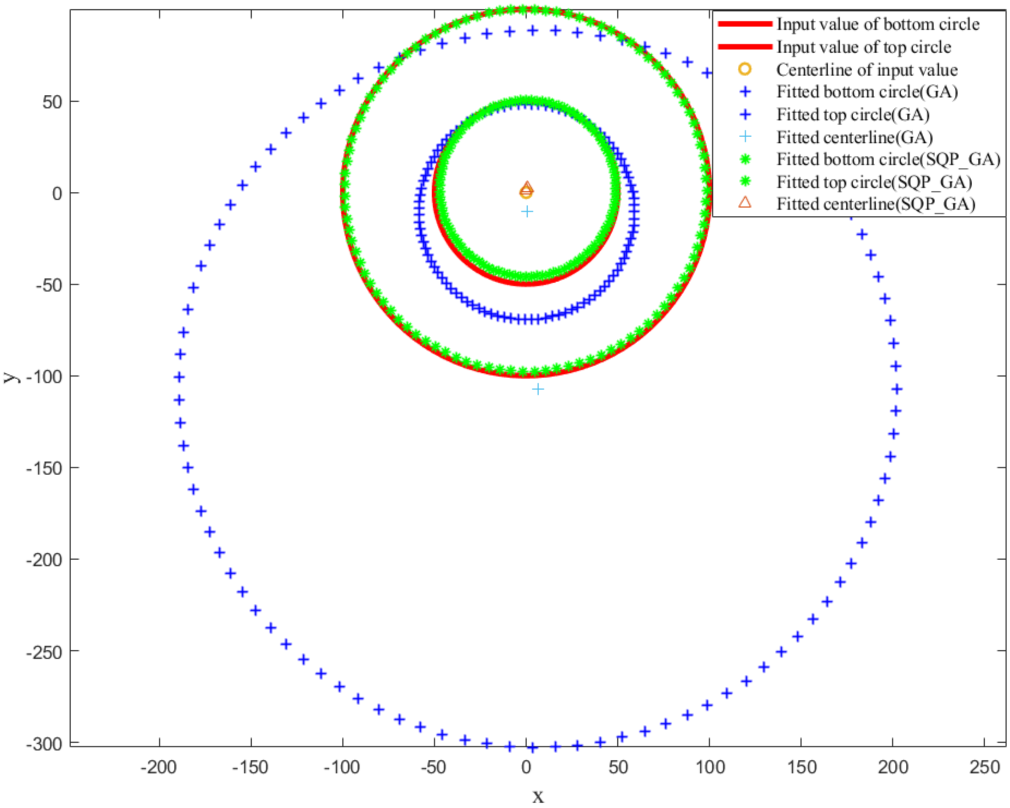

Figure 10. *Centerline fitting of top and bottom circles under data loss.*

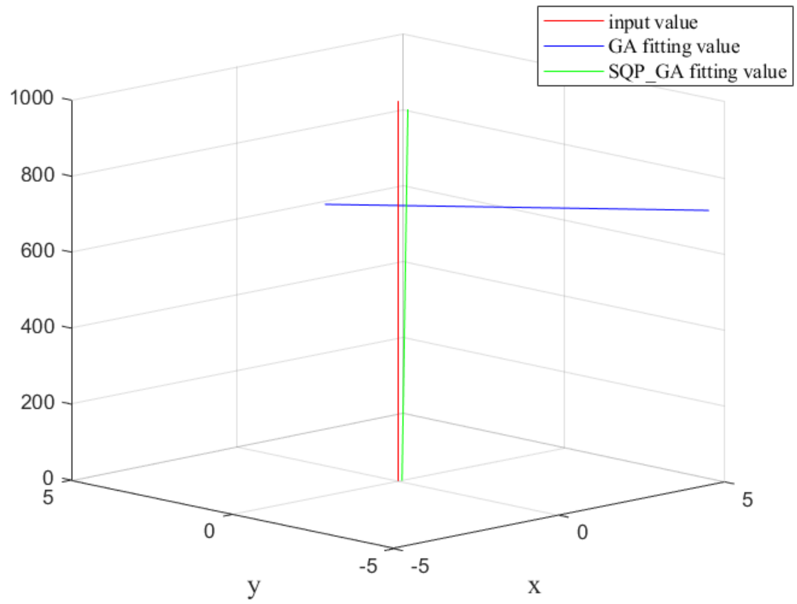

Figure 11. *Centerline fitting under data loss.*

Simulation results of the algorithm under centerline offset

In this experiment, the X-axis coordinate of the centerline of the rod is offset by 150. That is, the coordinates of the upper surface of the true frustum are (150, 0, 1000), and the centerline coordinate of the lower surface is (150, 0, 0).

Then, it is judged whether the improved genetic algorithm can obtain a good fitting effect with the centerline obtained from the experiment after the centerline is offset, and the influence of the centerline offset on the algorithm is evaluated.

Table 13. *Algorithm simulation results under centerline offset.*

	the value of input	genetic algorithm	improved genetic algorithm
X1	150.0000	150.7459	150.0000
Y1	0.0000	-2.2817	-0.0000
R1	100.0000	99.8501	100.0000
X2	150.0000	150.2318	150.0000
Y2	0.0000	-4.0029	0.0000
R2	50.0000	51.4718	50.0000

Table 14. *Accuracy of improved genetic algorithm under centerline offset.*

K	VAL	ANS
11	5.4170e-15	4.1635e-09

Table 13 and Table 14 show that when the centerline is offset, the improved genetic algorithm can still locate the position of the centerline, while the results obtained by using the genetic algorithm will have some deviation.

When the improved genetic algorithm is used to handle the situation of centerline offset, it can still perfectly fit the position of the centerline and suppress the fluctuation of the genetic algorithm, making the results more stable. Additionally, the precision of the results obtained with the improved genetic algorithm is high enough, and the point error is much smaller than that of the genetic algorithm, which demonstrates the superiority of the improved genetic algorithm. The specific effect is shown in Figure 12 and Figure 13.

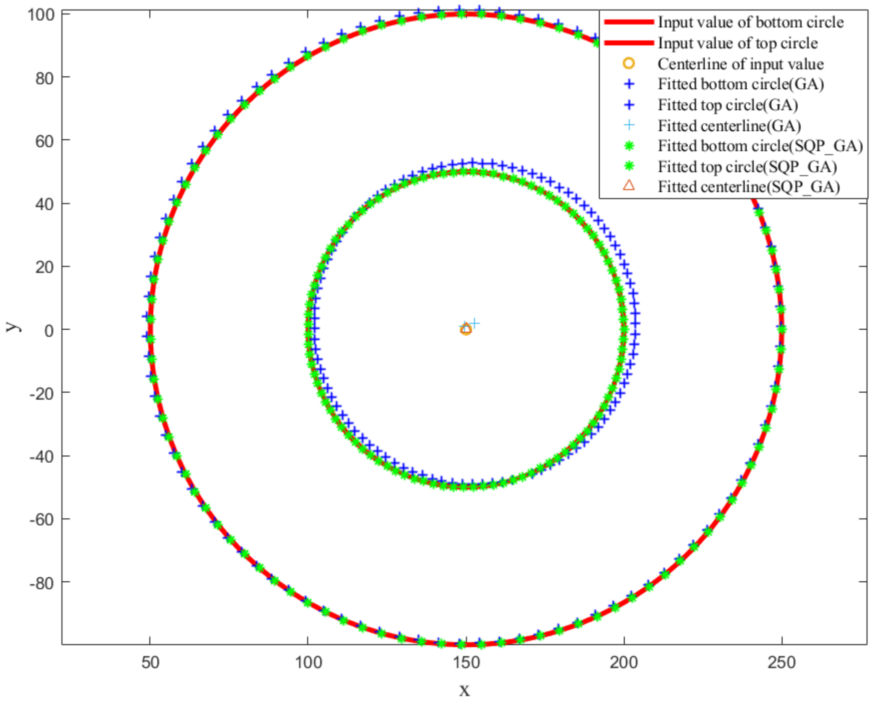

Figure 12. *Centerline fitting of top and bottom circles under centerline offset.*

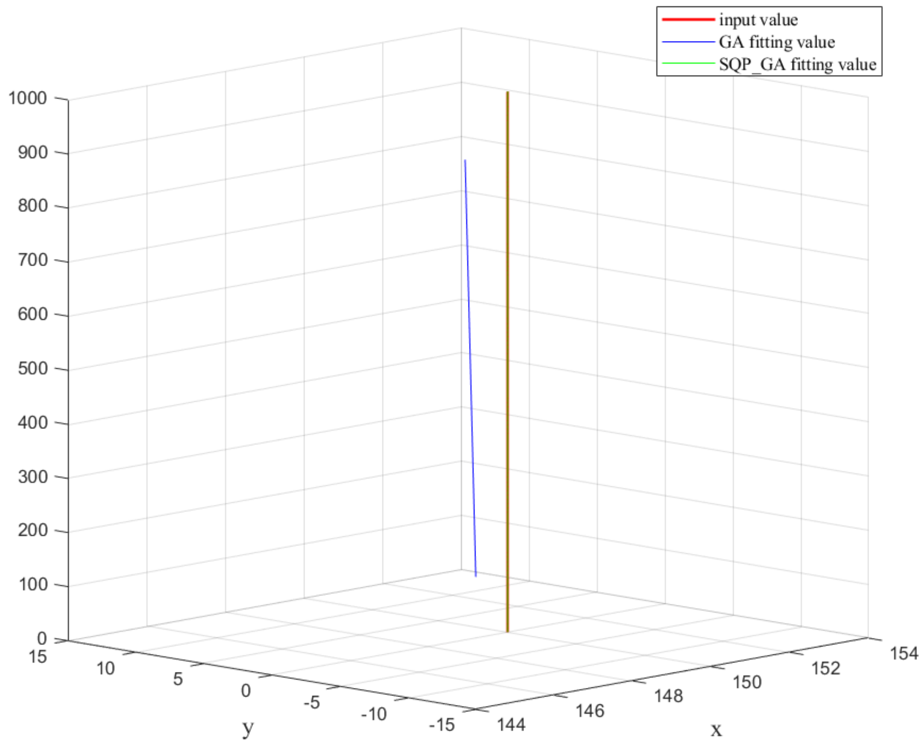

Figure 13. *Centerline fitting under centerline offset.*

4. Conclusion and Discussion

This paper grasps the basic theory of genetic algorithms and incorporates Sequential Quadratic Programming (SQP) into genetic algorithms. The objective function of SQP cannot directly adopt the objective function of the genetic algorithm. Therefore, a sparse screening of the genetic algorithm's objective function is needed to serve as the SQP algorithm's objective function, with the genetic algorithm's results used as the initial values for solving SQP.

By comparing the results data of genetic algorithm and improved genetic algorithm, it is found that the results obtained by genetic algorithms have fluctuations, oscillating around input values and failing to precisely reach the correct input values. In contrast, the results obtained by the improved genetic algorithm are stable and consistent with the central coordinates of the upper and lower surfaces of the input values. Consequently, it can be concluded that the SQP algorithm has a suppressive effect on the fluctuations of genetic algorithms, making the results fit better with the initial data and achieving higher precision.

By introducing point position errors, the data obtained can be used to analyze the impacts of random errors, gross errors, incomplete point cloud observations, and centerline deviations on genetic algorithms. The improved genetic algorithm has higher accuracy and a certain level of resistance to random errors, gross errors, incomplete point cloud observations, and centerline deviations. Its resistance is superior to that of genetic algorithms.

However, the hybrid optimization algorithm formed by the combination of genetic algorithms and SQP still has its shortcomings. Under normal point cloud conditions, the obtained results are consistent with the output values, but it still cannot completely avoid the influence of error factors such as random errors, gross errors, and incomplete observations on the algorithm. When the accumulated error factors reach a certain level, the results obtained cannot be used. At the same time, due to the limitations of the equipment, Inability to process and output large-scale data results in computational limitations.

Abbreviations

SQP	Sequential Quadratic Programming
GA	Genetic Algorithm

Author Contributions

Xianhao Deng: Conceptualization, Formal Analysis, Investigation, Methodology, Project administration, Resources, Supervision

Qiancheng Tan: Data curation, Investigation, Supervision, Writing – original draft, Writing – review & editing

Hao Liu: Formal Analysis, Resources, Software, Validation, Writing – review & editing

Yubiao Long: Data curation, Project administration, Software, Supervision, Visualization, Writing – review &

editing

Yonghui Qin: Conceptualization, Formal Analysis, Funding acquisition, Investigation, Methodology, Supervision, Validation, Visualization

Funding

This work is supported by the Guangxi Key Laboratory of Automatic Detecting Technology and Instruments (YQ22106) and the Program for University Student Innovation Training (No. S202310595170).

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] Yu C, Liu Z. Distributed Intelligent City Street Lamp Monitoring and Control System Based on Wireless Communication chip nRF401 [C]// International Conference on Networks Security. IEEE, 2009. https://doi.org/10.1109/NSWCTC.2009.69

[2] Liu Y, Zheng K, Liu Y, et al. Intelligent design of street lamp based on Arduino [J]. IOP Conference Series Earth and Environmental Science, 2020, 546: 052055. https://doi.org/10.1088/1755-1315/546/5/052055

[3] Goldberg D E. Genetic Algorithm in Search, Optimization, and Machine Learning [M]. Addison-Wesley Pub. Co, 1989. ISBN 10: 0201157675.

[4] Vitali D, Garbuglia F, D'Alessandro V, et al. The renewable energy in led standalone streetlight [J]. International Journal of Energy Production and Management, 2017: 167-172. https://doi.org/10.2495/EQ-V2-N1-118-129

[5] Yoon Y, Kim J. Design Hybrid Type Streetlight for Railway Station with Renewable Energy [J]. The transactions Korean Institute of Electrical Engineers, 2016, 65(12): 80-92. https://doi.org/10.5370/KIEE.2016.65.12.2103

[6] Wright A, Vose M, Jong K, et al. Foundations of Genetic Algorithms [M]. Springer-Berlin Heidelberg; Springer, Berlin, Heidelberg: 2005. https://doi.org/10.1007/b138412

[7] Grefenstette J. Genetic Algorithms and their Applications [M]. Taylor and Francis: 2013. ISBN: 9781134989737.

[8] Gill P E, Murray W, Saunders M A. SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization [J]. Society for Industrial and Applied Mathematics, 2002. https://doi.org/10.1137/S0036144504446096

[9] Dennis B H, Dulikravich G S, Han Z X. Optimization of Turbomachinery Airfoils with a Genetic/ Sequential-Quadratic-Programming Algorithm [J]. Journal of Propulsion and Power, 2001. https://doi.org/10.2514/2.5853

[10] Aditya Shastry K., H. A. Sanjay. A modified genetic algorithm and weighted principal component analysis based feature selection and extraction strategy in agriculture [J]. Knowledge-Based Systems, 2021, 232. https://doi.org/10.1016/j.knosys.2021.107460

[11] Wang Hao, Chen Shunhuai, Salerno Nunzio. An Approach to Ship Deck Arrangement Optimization Problem Using an Improved Multiobjective Hybrid Genetic Algorithm [J]. Mathematical Problems in Engineering, 2021. https://doi.org/10.1155/2021/8784923

[12] Aziza H, Krichen S. A hybrid genetic algorithm for scientific workflow scheduling in cloud environment [J]. Neural Computing and Applications, 2020, 32(18): 15263-15278. https://doi.org/10.1007/s00521-020-04878-8

[13] Isatou Hydara, Abu Bakar Md Sultan, Hazura Zulzalil, Novia Admodisastro. Cross-Site Scripting Detection Based on an Enhanced Genetic Algorithm [J]. Indian Journal of Science and Technology, 2015, 8(30). https://doi.org/10.17485/ijst/2015/v8i30/86055

[14] Lyantsev O D, Breikin T V, Kulikov G G, et al. OnLine Performance Optimisation of Aero-Engine Control System [J]. Automatica, 2003, 39(12): 2115-2121. https://doi.org/10.1016/S0005-1098(03)00224-3

[15] Mansoornejad B, Mostoufi N, Jalali-Farahani F. A hybrid GA–SQP optimization technique for determination of kinetic parameters of hydrogenation reactions [J]. Computers & Chemical Engineering, 2008, 32(7): 1447-1455. https://doi.org/10.1016/j.compchemeng.2007.06.018

[16] Balasubramanian P, Bettina S J, Pushpavanam S, et al. Kinetic Parameter Estimation in Hydrocracking Using a Combination of Genetic Algorithm and Sequential Quadratic Programming [J]. Industrial & Engineering Chemistry Research, 2003, 42(20): 4723-4731. https://doi.org/10.1021/ie021057s

[17] Xie L, Chen Y, Chang R. Scheduling optimization of prefabricated construction projects by genetic algorithm [J]. Applied Sciences, 2021, 11(12): 5531. https://doi.org/10.3390/app11125531

[18] Beck A, Eldar Y C. Sparsity Constrained Nonlinear Optimization: Optimality Conditions and Algorithms [J]. SIAM Journal on Optimization, 2012, 23(3): 1480-1509. https://doi.org/10.1137/120869778