

## Article

# Path Planning for Conformal Antenna Surface Detection Based on Improved Genetic Algorithm

Yifan Ding <sup>1</sup>, Xiaodong Du <sup>2</sup>, Changrui Wang <sup>1</sup>, Wei Tian <sup>1,\*</sup> , Chao Deng <sup>2</sup>, Ke Li <sup>1</sup> and Zihang Wang <sup>1</sup>

<sup>1</sup> College of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

<sup>2</sup> No. 29 Research Institute of China Electronics Technology Group Corporation, Chengdu 610036, China

\* Correspondence: tw\_nj@nuaa.edu.cn; Tel.: +86-25-84891836

**Abstract:** The conformal antenna is a precision device installed on the wing of an aircraft, and its components are distributed on a curved surface. Quality detection is required after assembly. In solving the path planning problem for conformal antenna surface detection, the traditional genetic algorithm faces problems such as slow convergence and easily falling into a local optimal solution. To solve this problem, an improved genetic algorithm combining the historical optimal population (CHOP-IGA) is proposed. First, the algorithm uses the probability-based four-nearest-neighbor method to construct an initial population. Subsequently, the probabilities of the crossover and mutation operators are dynamically adjusted. Next, the algorithm applies the crossover and mutation operators to the population and performs mutation operations on each individual of the historical optimal population. Then, the fitness value is calculated and the next generation of individuals is selected from the parent, offspring, and historical optimal populations according to the elite selection strategy. Finally, the current best fitness is checked to determine whether updating the historical optimal population is necessary. When the termination condition is satisfied, the algorithm outputs the optimal result. Experiments showed that the algorithm satisfactorily solved the path planning problem for conformal antenna surface detection, with a 48.44% improvement in detection efficiency.

**Keywords:** traveling salesman problem; improved genetic algorithm; probability-based four-nearest-neighbor method; historical optimal population; conformal antenna



**Citation:** Ding, Y.; Du, X.; Wang, C.; Tian, W.; Deng, C.; Li, K.; Wang, Z. Path Planning for Conformal Antenna Surface Detection Based on Improved Genetic Algorithm. *Appl. Sci.* **2023**, *13*, 10490. <https://doi.org/10.3390/app131810490>

Academic Editors: Stylianos Pappas, Sokratis Katsikas and Valentino Santucci

Received: 18 July 2023

Revised: 2 September 2023

Accepted: 16 September 2023

Published: 20 September 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The airborne electronic countermeasures (ECM) system is a key piece of electronic information equipment on military aircraft which significantly impacts combat results [1]. Compared with the traditional planar ECM antenna, the conformal antenna can fit well with the aircraft's wings, fuselage, and other areas, retain excellent aerodynamic performance, and reduce the radar-scattering cross-sectional area. Thus, it is one of the main directions of the future development of antenna technology [2,3]. Figure 1 shows the schematic diagram of a certain type of conformal antenna, which consists of a conformal substrate and a radiating surface. The radiating surface is distributed with many feature points, mainly for the welding pads and feeding holes distributed along the curved surface, which need to be processed one by one, and finally, each point is inspected for quality. Currently, this type of conformal antenna is usually assembled manually. Due to the small size and large number of feature points and their distribution on curved surfaces, this leads to extremely low production efficiency and makes it impossible to improve the quality consistency of the product and the qualification rate [4].

To address the above problems, an intelligent assembly system as shown in Figure 2 was designed. It uses a 3D optical scanning measurement system to reconstruct the workpiece. Then, the conformal antenna is coated, mounted, and welded. The 3D optical scanning measurement system can only detect the normal direction and the approximate

position of the feature points. Therefore, after reconstruction, it is necessary to use a high-resolution industrial camera to detect the precise position of the feature points to provide positional guidance for coating, mounting, welding, and auto optical inspection (AOI). Therefore, the planning of detection paths, including position and quality detection, is one of the key factors affecting production efficiency, and it also provides guidance for the planning of paths for plaster coating and welding. The above problems can be grouped into constrained three-dimensional traveling salesman problems (TSP), and there is no essential difference between two-dimensional TSP problems and three-dimensional TSP problems. Thus, the algorithms for solving two-dimensional TSP problems can be improved to solve three-dimensional TSP problems. Therefore, to realize the efficient and high-quality production of conformal antennas, suitable TSP-solving algorithms are needed to plan the detection paths correctly.

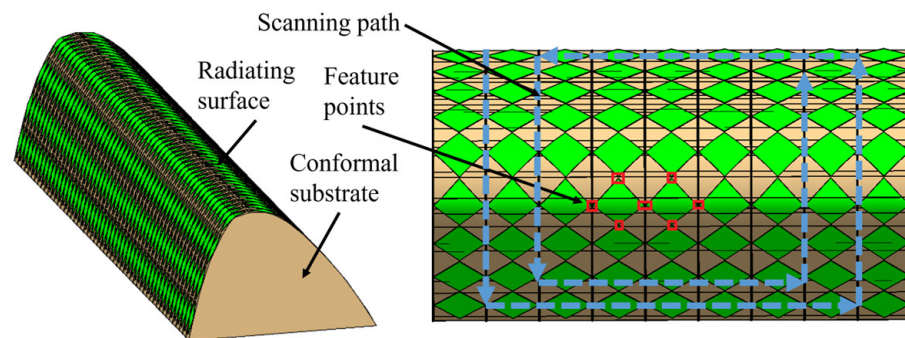


Figure 1. A certain type of conformal antenna.

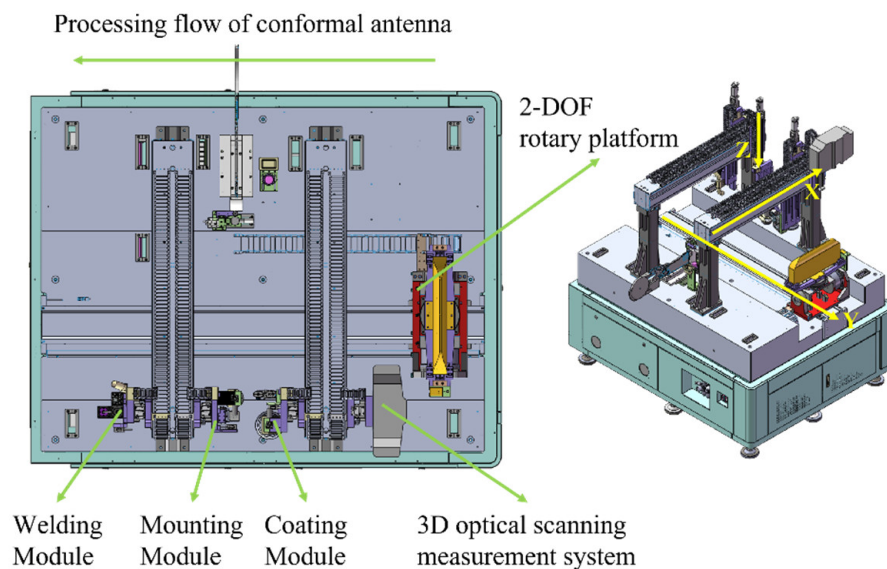


Figure 2. Intelligent assembly system.

TSP-solving algorithms can be divided into two categories: deterministic algorithms and intelligent algorithms. Deterministic algorithms include the enumeration method, dynamic programming method [5], branch and bound method [6], etc. Deterministic algorithms seek the exact solution to a TSP problem at the cost of a huge amount of time. The dynamic programming method, for example, has an exponential time complexity of  $O(N^2 2^N)$  and a space complexity of  $O(N 2^N)$ , which makes it almost impossible to deal with problems with more than 30 cities. Intelligent algorithms include the particle swarm optimization algorithm [7], ant colony algorithm [8], simulated annealing algorithm [9], sparrow search algorithm [10], hybrid algorithm [11], etc. The intelligent algorithms reach the solution by performing multiple iterations and utilizing the most useful information

from each generation. Although the exact solution cannot be guaranteed, this achieves the best balance between time and accuracy and is one of the best solutions for solving TSPs.

The genetic algorithm, an intelligent algorithm proposed by Holland in 1967, can also be used to solve TSPs. As one of the earliest of the intelligent algorithms, the genetic algorithm has evolved many variants, including the binary coded genetic algorithm, real number coded genetic algorithm, parallel genetic algorithm, hybrid genetic algorithm, etc. It has been successfully utilized in problems such as facility layout, job scheduling, image segmentation, path planning, and so on [12]. Xu et al. [13] proposed a bioinformatics heuristic genetic algorithm (BHGA) for small-scale TSPs. They constructed a novel crossover operator using equivalence matrices with reference to the kernel sequence comparison technique in biology to improve the performance of the algorithm. Arram et al. [14] proposed a multi-parent order operator (MPOX) modeled after the order operator (OX) and experimentally verified the reliability and computational efficiency of the operator in solving the TSP. Zhang et al. [15] addressed the problems of the traditional genetic algorithm (TGA) in solving the TSP. They proposed a genetic algorithm with jumping genes and heuristic operators, which achieved better results in terms of accuracy and convergence speed. Nevertheless, in the research on the improvement of genetic algorithms, current attention is mainly focused on the improvement of crossover and mutation operators and, generally, only one of the multiple steps of the genetic algorithm is improved. In addition, there is no literature on the application of genetic algorithms to constrained three-dimensional path planning problems, such as the path planning problem for conformal antenna surface detection.

Thus, this paper introduces the concept of historical optimal population and proposes an improved genetic algorithm combining the historical optimal population. First, the probability-based four-nearest-neighbor method is used to construct the initial population; then, adaptive crossover and mutation operators are used to fully utilize the crossover and mutation operations; subsequently, an elite selection strategy is used in the selection, where the next generation of individuals is selected without duplication and new individuals are introduced when they fall into a local optimal solution; finally, according to the change of the fitness value, the historical optimal population is recorded, which makes these individuals participate in the mutation operation. When certain conditions are met, the algorithm ends and the optimal value is output. The main contributions of this paper can be summarized as follows:

- Several key steps of the genetic algorithm are optimized, and the probability-based four-nearest-neighbor method and adaptive crossover and mutation operators are innovatively proposed;
- A new concept is introduced: the historical optimal population, which is independent of parents and offspring, and can better retain good genes when performing selection operations;
- Application of the CHOP-IGA algorithm to the detection path planning of conformal antennas, which improved the detection efficiency by 48.44%.

The rest of the paper is organized as follows. Section 2 briefly describes the standard TSP and the corresponding mathematical model, and gives the basic flow of the TGA algorithm. Section 3 describes the flow of the CHOP-IGA algorithm in detail and gives a flowchart of the algorithm. In Section 4, the effectiveness of the algorithm is verified via simulation and experiments. Finally, the conclusion is given in Section 5.

## 2. Techniques and Methods

### 2.1. Basic Mathematical Model of the TSP

The TSP can be described in words: a salesman wants to sell goods in several cities, and the geographical location of each city and the distance between any two cities are known. The salesman starts from any city, arrives in each city to make a sale, and finally returns to the starting point. A route needs to be found that offers the shortest total length [16]. This problem is a typical nondeterministic polynomial (NP) problem. More accurately,

it is an NP-complete problem [17]. That is, it is impossible to determine whether there is a polynomial time solution to this kind of problem, but it can be verified whether the answer is accurate in polynomial time. The mathematical model of the TSP problem can be expressed as follows:

$$\begin{aligned} \min & \sum_{i=1}^N \sum_{j=1, i \neq j}^N k_{i,j} d_{i,j} \\ \text{s.t.} & \begin{cases} \sum_{i=1}^N k_{i,j} = 1 \\ \sum_{j=1}^N k_{i,j} = 1 \\ k_{i,j} \in \{0, 1\} \end{cases} \end{aligned} \tag{1}$$

where  $N$  represents the problem size, that is, the number of cities;  $k_{i,j}$  indicates whether there is a path between the two cities. If a road exists, it is 1, otherwise it is 0;  $d_{i,j}$  represents the distance between two cities. This paper discusses the symmetric TSP, that is, for any  $i$  and  $j$ ,  $d_{i,j} = d_{j,i}$ . Furthermore, a new mathematical model was used to solve the path planning problem for conformal antenna surface detection, which is described in Section 4.2.

### 2.2. The TGA

The TGA simulates the theory of evolution in biology, and its main parameters include population size, crossover operator, mutation operator, selection operator, etc. Among them, the crossover operator represents the reproduction phenomenon, the mutation operator represents the mutation phenomenon with a small probability, and the selection operator represents the phenomenon of survival of the fittest. As the iteration continues, the population will become more and more “adapted to the environment”, so as to obtain an optimal or near-optimal solution. The flow of the TGA is shown in Figure 3. The TGA can be used for problems with multiple minima/maxima, multidimensional problems, discrete problems, etc., and has inherent parallelism, which makes it very suitable for solving TSPs. However, its convergence speed is slow, and it is easy for it to converge to a local optimal solution [18]. Therefore, the TGA needs to be improved to better solve the TSP.

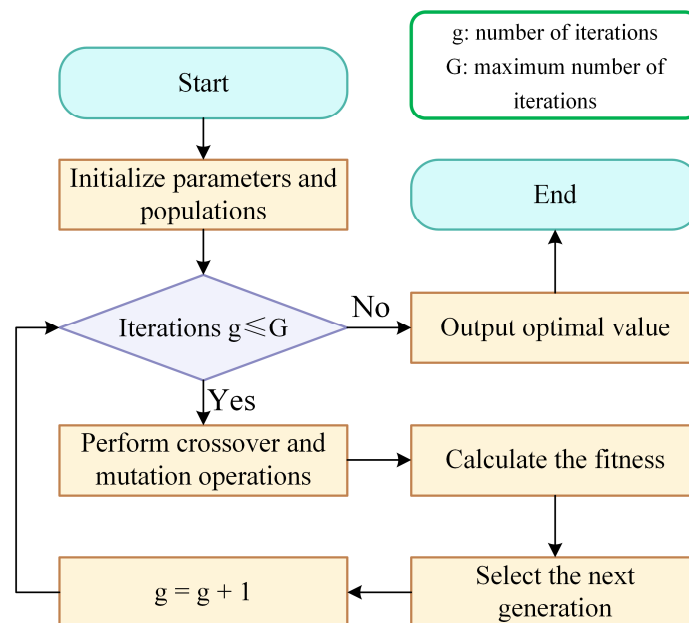


Figure 3. Flow chart of the TGA.

### 3. Improved Genetic Algorithm Combining the Historical Optimal Population

#### 3.1. Encoding and Population Initialization Methods

The first step of the genetic algorithm is encoding, which involves determining how to express the solution to a problem in the form of individuals. The encoding method generally needs to satisfy the principles of completeness, soundness, and non-redundancy. For TSPs, the general encoding methods include neighbor representation, sequence representation, and path representation. Among these, path representation is simple and clear. That is, if there are nine cities, a feasible path can be expressed as (5 1 6 8 9 4 7 2 3). When the genetic algorithm solves the TSP, it generally adopts the method of random initialization or nearest-neighbor initialization. The random initialization method randomly selects each city to ensure the diversity of the initial population, but the quality of the initial population is poor. The nearest-neighbor initialization method, also known as the greedy initialization method, uses the method of randomly selecting the starting point and finding the nearest point as the next point to construct the path, which ensures the quality of the initial population. When the genetic algorithm solves the TSP, if the population size is about twice the size of the cities, better results can be obtained [19]. However, the nearest-neighbor initialization method must have duplicate individuals in the population, which reduces the diversity of the population. Studies have shown that for medium- and small-scale TSPs, most of the points in the optimal path are connected to the four nearest points [20]. Moreover, for a certain type of conformal antenna, the number of feature points on it will not exceed 300, which is a medium- and small-scale problem. Therefore, a probability-based four-nearest-neighbor method is proposed for initialization, and the specific steps for randomly forming a path are as follows:

- Step 1. Randomly select a point from the city set as the starting point;
- Step 2. Record the four points closest to the point, set the probability of the four points being selected as 0.05, 0.1, 0.15, and 0.7 from far to near, and select the next passing point according to the cumulative probability;
- Step 3. Determine whether a complete path has been formed; if so, complete the initialization of the individual, otherwise, go to Step 2;
- Step 4. Determine whether the population has been generated; if so, end the algorithm, otherwise, go to Step 1.

Take the bayg29 problem in TSPLIB as an example. Combined with Formula (2) [21], the population diversity and average path length of the initial population formed by the above three methods were compared, and the results are shown in Table 1.

$$D = \sqrt{\frac{1}{N} \sum_{q=1}^N (f_q - \frac{1}{N} \sum_{l=1}^N f_l)^2} \tag{2}$$

where  $D$  represents the diversity of the population, and the larger the value, the better the diversity of the population;  $f_q$  represents the fitness of the individual  $q$ ; and the total length of the path is used as the fitness in this paper.

**Table 1.** Comparison of the average path length and population diversity of the three initialization methods.

Method	Item	1	2	3	4	Average
Random	Diversity	1123.97	94.1	292.1	34.74	386.23
	Path length	$2.63 \times 10^4$	$2.65 \times 10^4$	$2.65 \times 10^4$	$2.63 \times 10^4$	$2.64 \times 10^4$
Nearest neighbor	Diversity	61.00	58.24	36.8	15.39	42.86
	Path length	$1.1 \times 10^4$	$1.1 \times 10^4$	$1.11 \times 10^4$	$1.1 \times 10^4$	$1.1 \times 10^4$
Probability-based four-nearest-neighbor	Diversity	99.18	52.91	186.74	78.25	104.27
	Path length	$1.39 \times 10^4$	$1.37 \times 10^4$	$1.36 \times 10^4$	$1.36 \times 10^4$	$1.37 \times 10^4$

It can be seen from Table 1 that compared with random initialization, both the probability-based four-nearest-neighbor initialization and the nearest-neighbor initialization greatly reduced the path length. The probability-based four-nearest-neighbor initialization was only at most 26% longer than the nearest neighbor initialization, while the diversity increased at most by 1850%. Therefore, the probability-based four-nearest-neighbor initialization can be used as a better initialization method to maintain the diversity of the population under the premise of generating better paths.

### 3.2. Adaptive Crossover and Mutation Operators

The crossover operation tries to retain the good genes of the parents by simulating the reproductive behavior found in nature. The crossover operation is done through crossover operators, usually including partially mapped crossover (PMX), sequential crossover, multi-parent partially mapped crossover (MPPMX), multi-parent sequential crossover, etc. [14]. These crossover operators generally feature measures to avoid repetition and have achieved good results in solving TSPs. In this paper, a sequential crossover operator with a good effect was used to perform the crossover operation, and the operator was slightly improved. The crossover process of the operator is shown in Figure 4. The improvement was mainly reflected in the generation of offspring 2, that is, the selected fragment of parent 2 was placed at the front of offspring 2 instead of at the same position as in parent 2. This kind of operation improves the diversity of the population to a certain extent. It also prevents the problem that the original sequential crossover operator can only produce two individuals that are exactly the same as the parents when the two parents are the same in a few cases. As an example, two identical parents, (1 2 3 4 5), were randomly selected for crossover at two positions: (1 | 2 3 4 | 5). The crossover results were (1 2 3 4 5) and (2 3 4 1 5), respectively, producing new individuals instead of the same individuals as the parents.

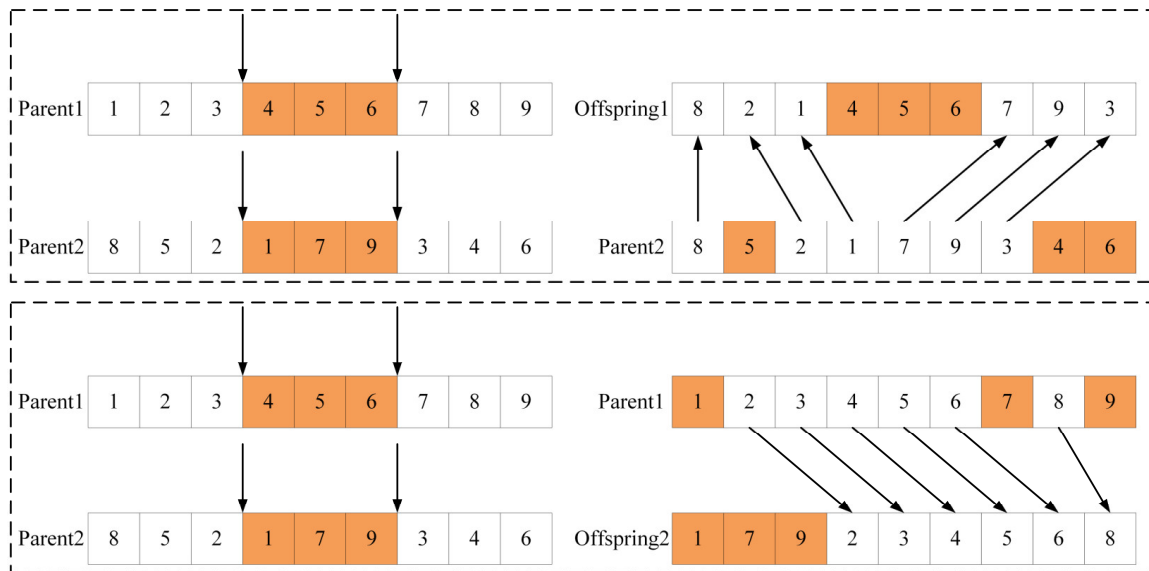


Figure 4. Schematic diagram of improved sequential crossover operator.

The mutation operation changes the genetic information of organisms by simulating the variation of chromosomes. Embodied in the genetic algorithm, it can avoid iterations falling into a local optimal solution, and there is a certain probability of jumping out of the local optimal solution. The mutation operation is completed by mutation operators, usually including two-point exchange mutation, 2-opt, sliding mutation, partial reverse sequence mutation, center inverse mutation, etc. To overcome the limitations of using a single mutation operator, this work used a combination of multiple mutation operators to perform mutation operations. A schematic diagram of some mutation operations is shown in Figure 5. Among them, 2-opt mainly plays the role of eliminating crossover

between lines. First, the lines that are impossible to intersect are eliminated through the rapid repulsion test. Second, the intersecting lines are quickly located through the straddle test, so as to eliminate the intersection, as shown in Figure 6. Compared with the traditional method of calculating the intersection point of two straight lines and judging whether the intersection point is within the range of the line segment [22], this method has a small amount of calculation and faster operation speed.

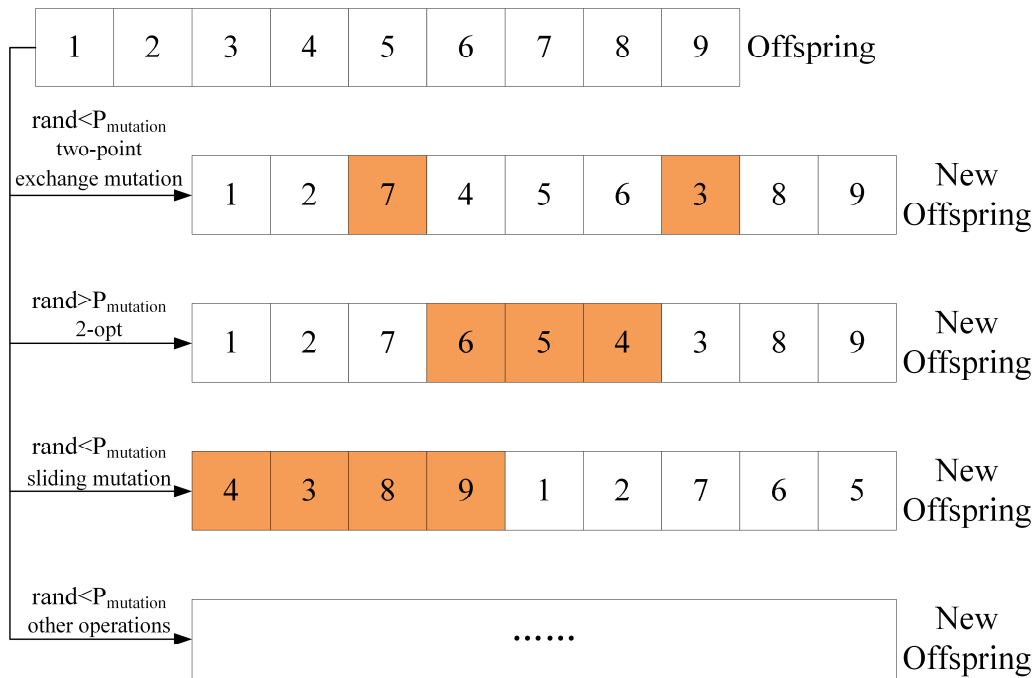


Figure 5. Mutation operation.

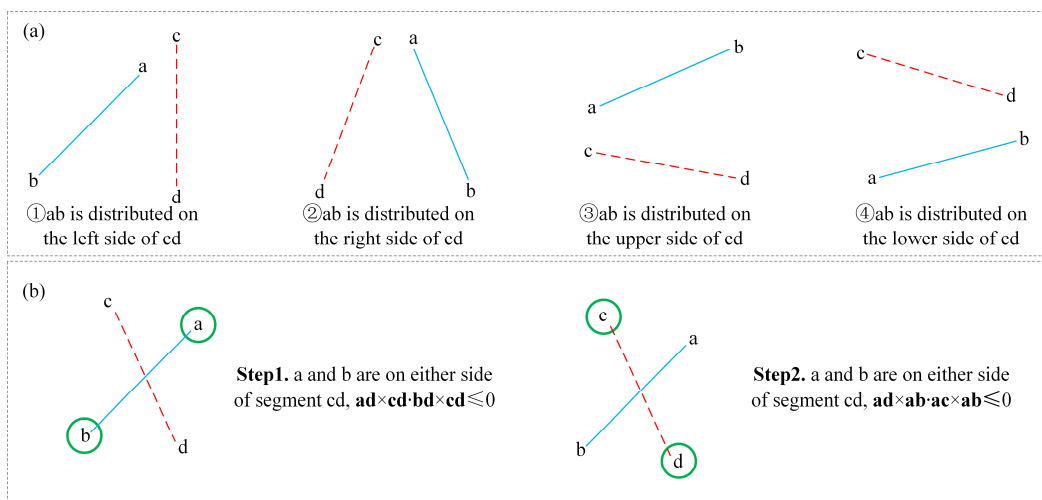


Figure 6. (a) The rapid repulsion test; (b) the straddle test. Bold  $\mathbf{ad}$ ,  $\mathbf{cd}$ , etc. represent vectors.

In addition to the implementation of crossover and mutation operators, the probability of crossover and mutation will also affect the iteration results. Taking the crossover operator as an example, a larger crossover probability will increase the computational cost, while a smaller crossover probability will cause the search to slow down. The research of Hassanat et al. [23] showed that when solving medium- and small-scale TSPs, the linear decrease of the crossover probability and the linear increase of the mutation probability can produce better iterative results. This work built on this foundation by introducing

an adaptive adjustment strategy of crossover and mutation probability: when the fitness did not change for multiple generations, the change speed of crossover probability and mutation probability was accelerated, as shown in Formulas (3) and (4).

$$P_{cross} = \begin{cases} P_{cross} - \frac{1}{GENERATION}, & \text{if } t < THRESHOLD/10 \\ P_{cross} - \frac{2}{GENERATION}, & \text{if } t \geq THRESHOLD/10 \end{cases} \quad (3)$$

$$P_{mutation} = \begin{cases} P_{mutation} + \frac{1}{GENERATION}, & \text{if } t < THRESHOLD/10 \\ P_{mutation} + \frac{5}{GENERATION}, & \text{if } t \geq THRESHOLD/10 \end{cases} \quad (4)$$

where  $P_{cross}$  represents the crossover probability of each round, for which the initial value is 0.9 and the minimum value is set to 0.4;  $P_{mutation}$  denotes the mutation probability of each round, for which the initial value is 0.1 and the maximum value is set to 0.9;  $GENERATION$  represents the upper limit of iterations, which is set to 1000;  $THRESHOLD$  denotes the upper limit of the number of generations the fitness remains unchanged, which is set to 100;  $t$  represents the number of generations for which the optimal path length is continuously constant; and the iteration ends if  $t$  exceeds  $THRESHOLD$ .

In the later stages of iteration, the paths represented by the individuals in the population are already largely free of crossovers, meaning that the 2-opt mutation is no longer useful. Therefore, this work adopted an adaptive mutation operation, which is shown in the following: for 2-opt, the operator is executed only when the random number  $P > P_{mutation}$ ; for other mutation operations, it is executed only when the random number  $P < P_{mutation}$ , as shown in Figure 5. Together with the adaptive adjustment strategy of mutation probability described in the previous section, it can serve the purpose of optimizing the cross-path mainly using the 2-opt mutation in the early stage, and jumping out to the locally optimal solution mainly using various other mutation operators in the later stages. Taking kroA100 in TSPLIB as an example, the comparison results between this work’s method and the literature [23] method with the same remaining parameters are shown in Figure 7, where (a) and (b) represent the change curves of the crossover and mutation probabilities of this work’s method and the method described in the literature [23] at one time. It can be seen that the method described in this paper could automatically adjust the crossover and variation probabilities when falling into the local optimal solution, and the 2-opt operator was invoked more frequently in the early stage to accelerate the iteration, which had a better chance of reaching or converge to the optimal solution. The final paths obtained by the two methods are indicated by (c) and (d), respectively. The length of the former was 21,285.44, and the length of the latter was 21,448.46, which shows that the method described in this paper achieved a better result, and the result was the optimal solution of kroA100. A comparison of the results of multiple trials is shown in Table 2, where the error rate was calculated using Formula (5).

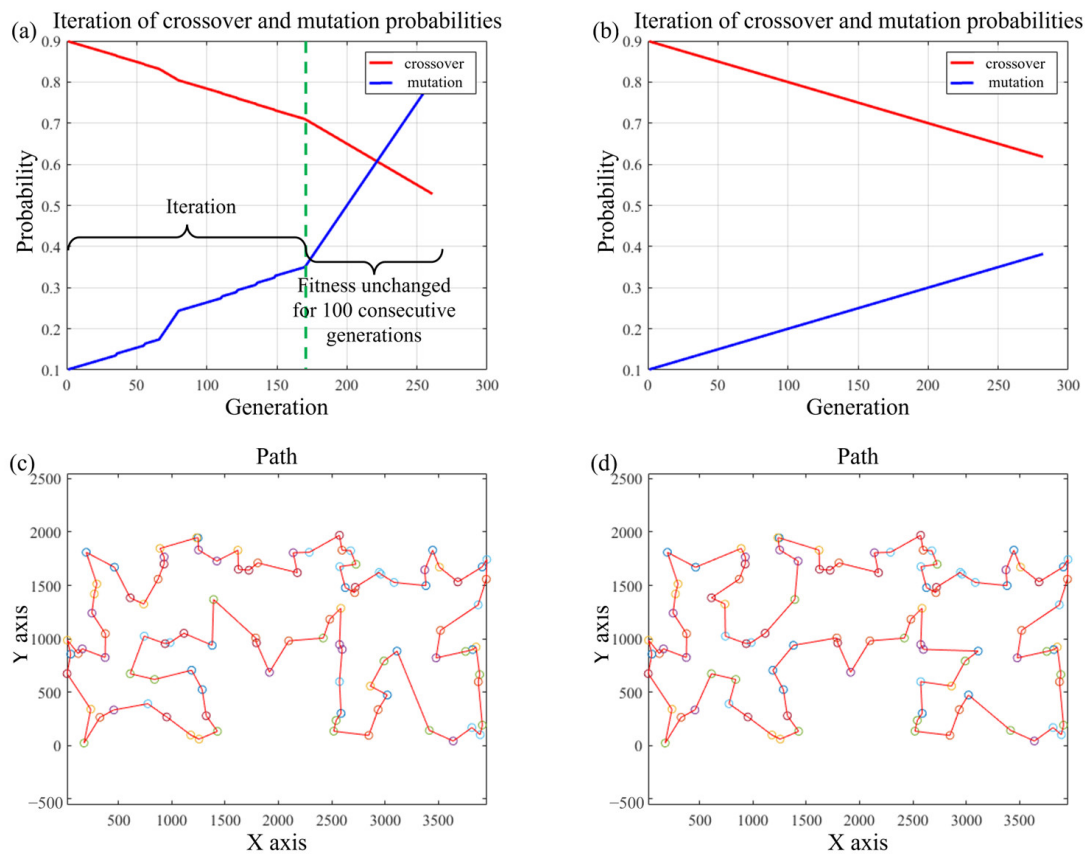
$$P_f = \frac{f_{average} - f_{best}}{f_{best}} \times 100\% \quad (5)$$

where  $P_f$  denotes the error rate;  $f_{average}$  denotes the average value of the path length after 10 trials;  $f_{best}$  refers to the optimal value of the path length, which is 21,285.44 for kroA100.

**Table 2.** Comparison of the results of the method described in this paper with the method described in the literature [23].

Method	Number of Tests	Average Length	Error Rate	Average Number of Iterations
Methodology of this paper	10	21,329.9	0.211%	162.5
Method described in the literature [23]		21,442.2	0.739%	216.8





**Figure 7.** Comparison of the iterative results of the method described in this paper and the method described in the literature [23]: (a) the iterative results of crossover and mutation probabilities of the method described in this paper; (b) the iterative results of crossover and mutation probabilities of the method described in the literature [23]; (c) the kroA100 paths obtained via this paper’s method; (d) the kroA100 paths obtained via the method described in the literature [23].

### 3.3. Fitness Function

When using genetic algorithms to solve TSPs, it is usually possible to use the inverse of the path length or directly use the path length as the fitness function. When using the inverse of the path length as the fitness function, it is necessary to select individuals with larger fitness values; when using the path length, it is necessary to select individuals with smaller fitness values. In this work, the path length was directly chosen to use as the fitness function, which was simple and intuitive to calculate, as shown in Formula (6).

$$f = \sum_{i=1}^N \sum_{j=1, j \neq i}^N k_{i,j} d_{i,j} \tag{6}$$

For the path planning problem for conformal antenna surface detection, the fitness function is more complex. This function is described in Section 4.2.

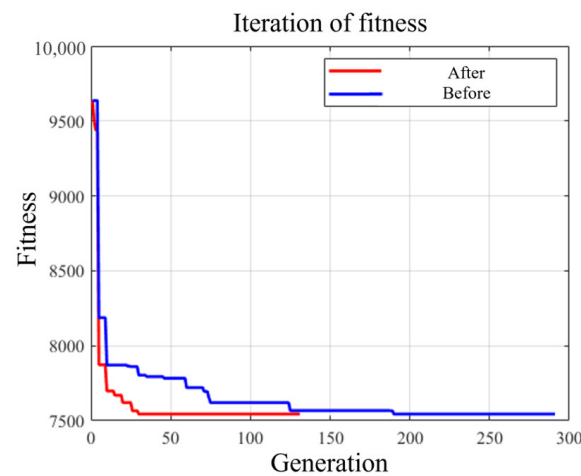
### 3.4. Historical Optimal Population

Biological genetics play a vital role in the evolutionary process in nature. At the same time, organisms can also draw experience and wisdom from their effective predecessors to guide the development of the population. In this paper, the concept of the historical optimal population is introduced, i.e., each time the optimal fitness changes, this contemporary optimal individual is included in the historical optimal population. For the historical optimal population, two main operations are performed: full mutation and merging. In this case, performing the full mutation operation on the historical optimal population is

similar to the partheno-genetic algorithm [24], which performs the mutation operation on all individuals, with the same mutation operator as described in Section 3.2, while removing the mutation probability. This operation is performed on only one individual, simplifying the operation and improving computational efficiency. In addition, performing the mutation operation on this particular population has the probability of producing a better solution. The reason for not using the crossover operation is that there is a great deal of similarity between the individuals of the historical optimal population, and performing the crossover operation on them is ineffective. The merge operation refers to the merging of the mutated historical optimal population with the parent and offspring individuals, which together continue to the next step of the local optimization operation. Taking berlin52 in TSPLIB as an example, the comparison results of using the historical optimal population and not using the historical optimal population, with all other conditions being the same, are shown in Table 3, and a representative comparison of the iteration curves is shown in Figure 8. The comparison results show that with the introduction of the historical optimal population, the iteration speed was accelerated and there was a better chance of obtaining an optimal solution.

**Table 3.** Comparison of results before and after the introduction of the historical optimal population.

Use of the Historical Optimal Population	Number of Tests	Average Length	Error Rate	Average Number of Iterations
No	10	7616.64	0.957%	196.1
Yes		7561.63	0.228%	177.6



**Figure 8.** Comparison of iteration curves (Whether or not historical optimal population is used).

### 3.5. Local Optimization Operation

Local optimization operations generally include 2-opt operation, 3-opt operation, neighbor-node exchange operation, etc. Combining local optimization operations with intelligent algorithms can generally achieve better optimization results. In this work, 3-opt and neighbor-node exchange operations were used to locally optimize the merged population, i.e., to optimize the parent, the offspring, and the historical optimal populations. In this case, the principle of the 3-opt operation is shown in Figure 9. The method traverses all points and inserts the point into all feasible locations. If  $d_{i,i+2} + d_{j,i+1} + d_{i+1,j+1} < d_{i,i+1} + d_{i+1,i+2} + d_{j,j+1}$ , the individual is updated, otherwise, the individual is kept unchanged. The time complexity of the operation is  $O(N^2)$  and  $N$  is the city size. To reduce the computation, the probability  $P_3$  is set to 0.1. When the random number  $P < P_3$ , the 3-opt operation is executed for this individual; otherwise, it is not executed. The neighbor-node exchange operation exchanges every two neighboring nodes. If  $d_{i,i+2} + d_{i+2,i+1} + d_{i+1,i+3} < d_{i,i+1} + d_{i+1,i+2} + d_{i+2,i+3}$ , the individual is updated; otherwise, the individual stays unchanged, as shown in Figure 10. Since the time complexity of this

operation is only  $O(N)$ , it is performed for all individuals. The genetic algorithm with the addition of the local optimization operation sped up the convergence of the preliminaries and hence the iterations compared to when no local optimization was introduced, as shown in Figure 11.

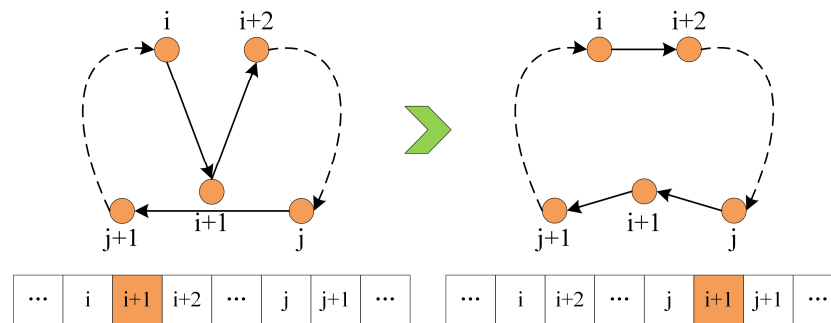


Figure 9. 3-opt operation. “i+1” denotes a node that needs to be replaced in the connection order.

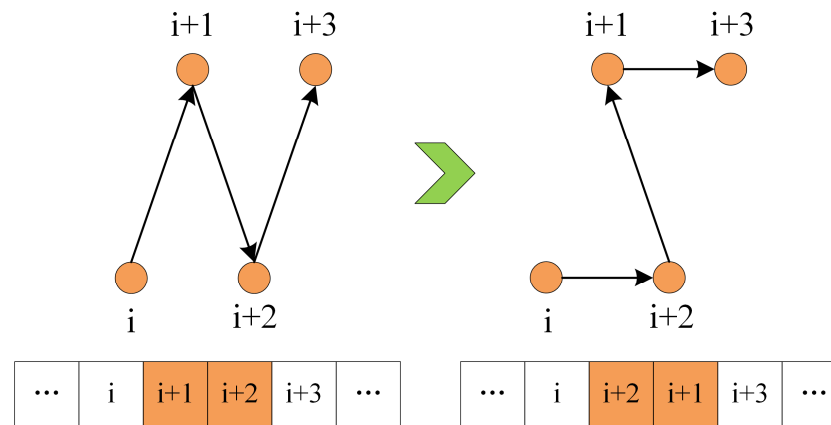


Figure 10. Neighbor-node exchange operation. “i+1” and “i+2” denote nodes that need to be replaced in the connection order.

### 3.6. Selection Operation

The selection operation is an important way to ensure the continuity of good individuals. In this paper, the elite selection strategy was adopted to select the next generation from the population composed of the parent population, the offspring population, and the historical optimal population, and to introduce new individuals into the population at the appropriate time. The specific steps are as follows:

Step 1. Sort the populations from lowest to highest fitness and record the number of individuals  $N$  after removing the historically optimal population;

Step 2. Eliminate duplicated individuals and record the number  $n$  of remaining individuals;

Step 3. Compare the size of  $n$  with  $N/2$ ; if  $n \geq N/2$ , take the first  $N/2$  individuals of the remaining individuals as the next generation to participate in the next cycle; if  $n < N/2$ , take all the remaining individuals as the next generation of individuals and use the probability-based four-nearest-neighbor method to supplement the  $N/2 - n$  individuals.

Compared with the traditional roulette and tournament selection strategies, the elite selection strategy avoids the individuals in the population converging to be the same at a later stage and improves the probability of jumping out of the optimal solution by introducing new individuals.

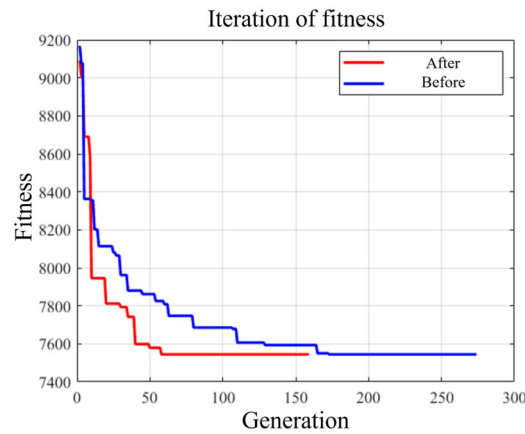


Figure 11. Comparison of iteration curves (Whether or not local optimization operation is used).

### 3.7. The Overall Flow of CHOP-IGA

In summary, the flow of the improved genetic algorithm combining the historical optimal population is shown in Figure 12. The termination condition of the algorithm is set when the maximum number *GENERATION* of iterations is reached or the fitness value is unchanged for successive *ITERATION* generations.

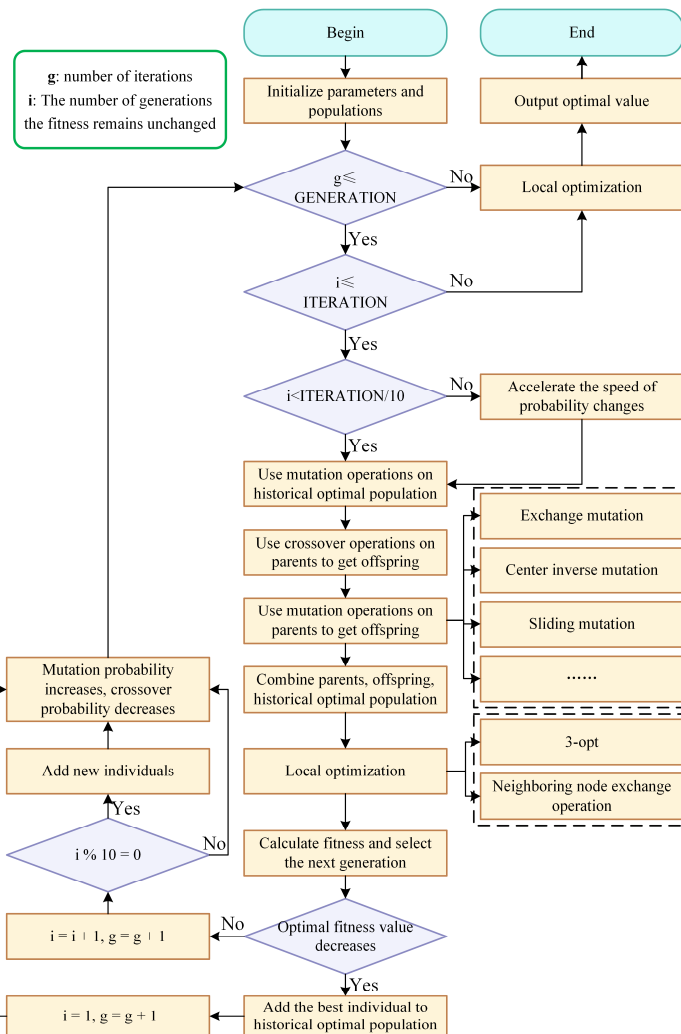


Figure 12. Flowchart of the CHOP-IGA algorithm.

### 4. Experiments

#### 4.1. Standard Two-Dimensional TSP

TSPLIB is a database created and maintained by Heidelberg University which contains symmetric TSPs, asymmetric TSPs, vehicle path problems, etc., and is an authoritative database of combinatorial optimization problems. Therefore, the data in TSPLIB were used to verify the effectiveness of the algorithm proposed in this paper and to compare it with the TGA, the multi-population genetic algorithm based on cross accessibility evaluation (AEMPGA) [25], and the discrete spider monkey optimization algorithm (SMO) [26]. The computer used for the experiment ran Windows 10 and was configured with an Intel Core i5-7300HQ processor and 16 GB of RAM. The algorithm in this work set the population size to be approximately twice the city size; the initial value of crossover probability was 0.9; the initial value of mutation probability was 0.1; the number of trials was 20. The results of the comparison are shown in Table 4. The optimal solutions obtained from some medium- and small-sized datasets are shown in Figure 13a–e, and the iteration curves are shown in Figure 13f. It is important to note that this work used a realistic calculation of lengths, i.e., without rounding the distance between two points, which may have led to different optimal paths. For example, in *eil51*, the path length as a real number is 428.87, which is a more realistic length, while the path length as an integer is 426, when the corresponding paths are different.

Table 4. Path length comparison.

Data Sets	Shortest Distance	Average Length				Minimum Length			
		TGA	AEMPGA	SMO	CHOP-IGA	TGA	AEMPGA	SMO	CHOP-IGA
<i>ulysses16</i>	30.88	74.19	<b>73.99</b> <sup>a</sup>	<b>73.99</b>	<b>73.99</b>	<b>73.99</b>	<b>73.99</b>	<b>73.99</b>	<b>73.99</b>
<i>ulysses22</i>	75.31	75.76	75.51	—	<b>75.31</b>	<b>75.31</b>	<b>75.31</b>	—	<b>75.31</b>
<i>bayg29</i>	9074.1	9224.4	— <sup>b</sup>	9074.2	<b>9074.1</b>	<b>9074.1</b>	—	9074.2	<b>9074.1</b>
<i>dantzig42</i>	679.2	708.22	686.21	—	<b>679.52</b>	685.52	692.91	—	<b>679.2</b>
<i>eil51</i>	428.87	446.02	—	436.96	<b>431.49</b>	432.16	—	<b>428.87</b>	<b>428.87</b>
<i>berlin52</i>	7544.4	8038.4	—	7633.6	<b>7561.6</b>	7735.8	—	<b>7544.4</b>	<b>7544.4</b>
<i>kroA100</i>	21,285	22,454	—	22,024	<b>21,332</b>	21,698	—	21,298	<b>21,285</b>
<i>lin105</i>	14,383	15,264	—	15,114	<b>14,458</b>	14,786	—	<b>14,383</b>	<b>14,383</b>
<i>pr144</i>	58,535	60,599	—	—	<b>58,632</b>	59,244	—	—	<b>58,535</b>

<sup>a</sup>: Data in bold indicate that the algorithm achieved the best result in this dataset. <sup>b</sup>: “—” indicates that the papers used as comparisons did not provide the result of the dataset.

From Table 4, it can be seen that CHOP-IGA achieved better results compared to those reported in the literature [25,26] and the TGA, and produced a known optimal solution in all 20 experiments. As can be seen in Figure 13, CHOP-IGA iterated faster. For smaller TSPs, such as *bayg29*, *dantzig42*, and *eil51*, the optimal solution could be iterated within 50 iterations in the best case; for medium-sized TSPs, such as *pr144*, the optimal solution could be iterated within 100 iterations in the best case, and it had a certain ability to jump out of the local optimal solution. It can be seen that the proposed CHOP-IGA was able to solve medium- and small-scale TSPs better.

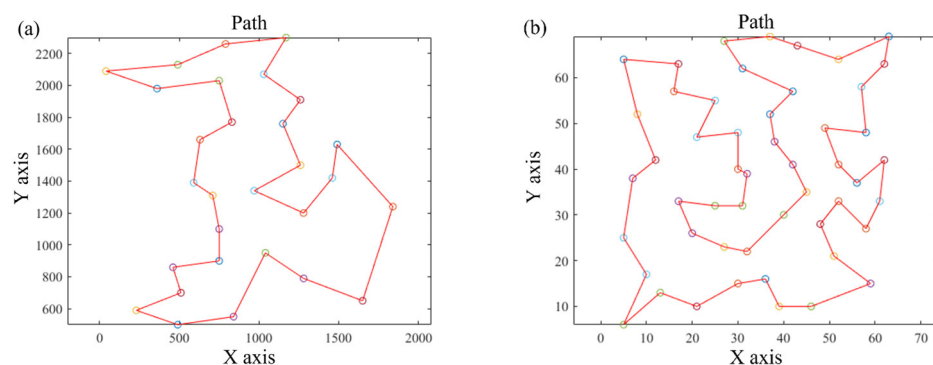
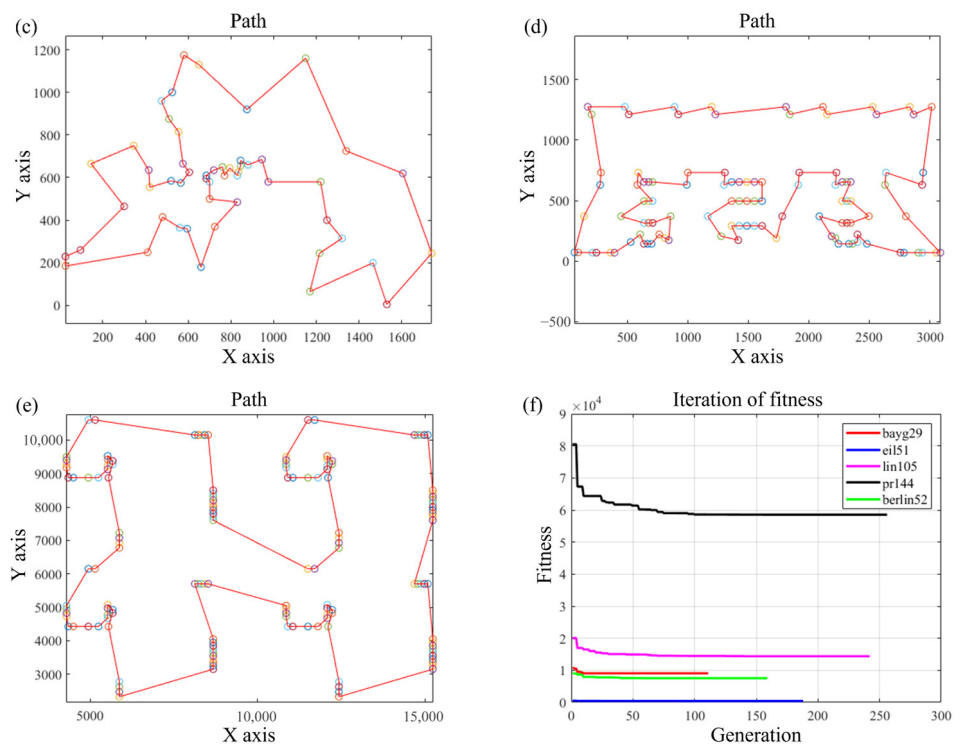


Figure 13. Cont.



**Figure 13.** (a–e) the optimal solutions for the datasets bayg29, eil51, berlin52, lin105, and pr144, respectively; (f) is the iteration curve when solving the above five datasets.

#### 4.2. Path Planning for Conformal Antenna Surface Detection

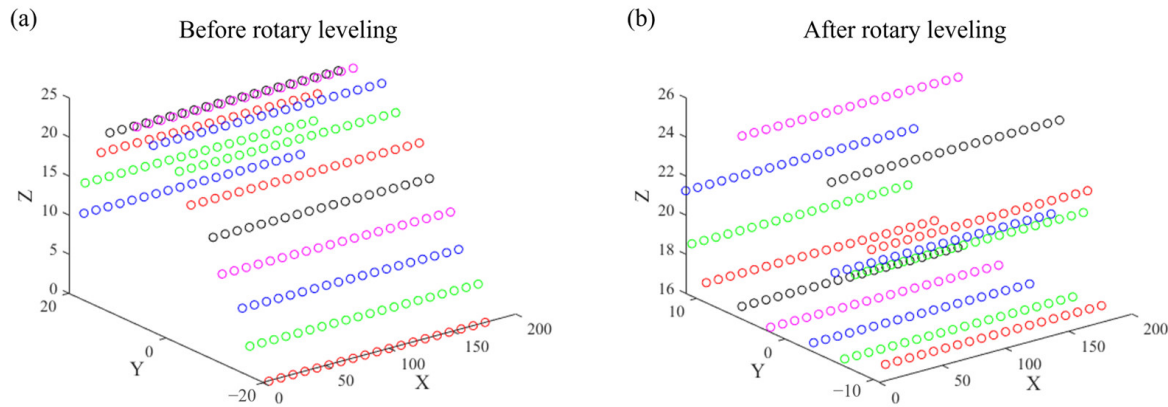
The model of a conformal antenna is distributed with feature points (mounted resistors and feeders), which are uniformly distributed on the curved surface of the antenna. There is a total of 253 feature points distributed on the model that need to be accurately located and inspected after soldering, and the size of the model is not fixed, so the number of feature points and their distribution will change. Therefore, the efficiency of detection of the feature points distributed on this model is crucial.

The algorithm proposed in this paper solved the path planning problem for conformal antenna surface detection well, but some changes were needed, i.e., removing the 2-opt operation and changing the fitness function. A large number of randomized spatial 3D point path planning experiments have shown that in the vast majority of cases of the three-dimensional TSP, it is not possible to intersect a straight line with another line, which eliminates the need to use the 2-opt operation and thus saves a portion of the computational time. In addition, both accurate positioning and AOI require the feature points to be leveled and then processed, i.e., the normal direction of the feature point is adjusted to be vertically upward. A schematic diagram before leveling is shown in Figure 14a. There is a total of 13 rows of feature points on the model; the feature points in the same row are separated by 9.5 mm and indicated by the same color, and the X coordinate of the first feature point is deviated between different rows. To make it easier to see, the XYZ coordinate axes are scaled down to a certain extent. Figure 14b shows a schematic diagram after leveling. The figure represents the position of each row after leveling, i.e., the actual position of each row during either detection or processing. It can be seen that additional adjustment time was required when adjusting the rotation angle to accommodate different point positions, as shown in Figure 15. Therefore, the TSP was transformed into a three-dimensional constrained TSP similar to the one described in the literature [27]. When using Formula (5) to calculate the length of each feasible path, the additional rotation angle generated when crossing different rows needed to be considered. Since the length and angle could not be

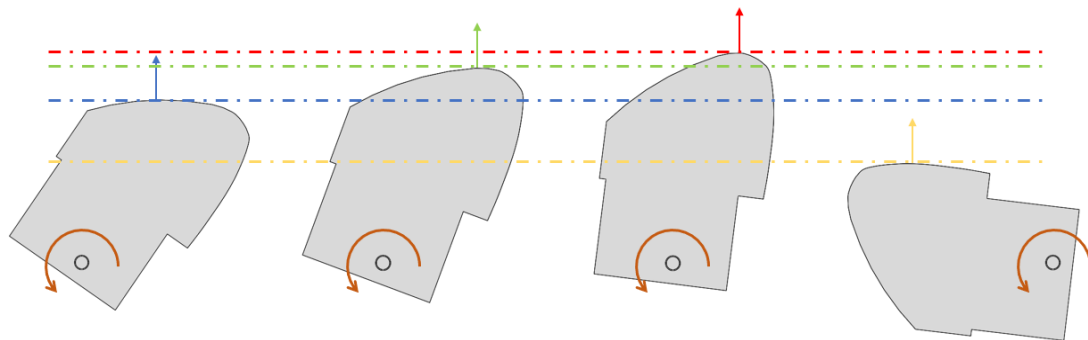
measured uniformly, the time  $t$  was used uniformly as a measure, as shown in Formula (7).

$$t = \sum_{i=1}^n \sum_{j=1, i \neq j}^n k_{i,j} \left( \frac{d_{i,j}}{v} + \frac{\alpha_{i,j}}{\omega} \right) \tag{7}$$

where  $v$  denotes the moving speed of the end-effector;  $\alpha_{i,j}$  denotes the angle of rotation required to rotate from feature point  $i$  facing upward to feature point  $j$  facing upward;  $\omega$  denotes the angular speed of rotation of the A-axis of the 2-DOF rotary platform.



**Figure 14.** (a) 3D coordinate distribution of feature points before leveling; (b) 3D coordinate distribution of feature points after leveling.



**Figure 15.** Changes in height of feature points during leveling.

The feature points in the conformal antenna were extracted and the path planning was performed using CHOP-IGA, and the obtained results are shown in Figure 16. In fact, due to the limitation of the working distance of the detection camera, the scanning path also needed to increase the height of the camera’s focal length  $f$  in the Z-axis direction, i.e., the Z coordinate was shifted upwards by 100 mm. It can be seen from the figure that, to avoid crossing constantly between different rows, which led to a surge in the time spent, the optimal result was to have only one path between two different rows, which reduced the time spent on leveling.

To verify the further validation of the scanning efficiency improvement, a test platform was built, as shown in Figure 17.

A Basler industrial camera was used for detection simulation, which was mounted on the Z axis of a high-precision three-axis linear module. The conformal antenna was fixed on the 2-DOF rotary platform. During the test, three rows of features on both sides and in the middle of the conformal antenna were selected for scanning. The scanning times of the original path and the path optimized by the CHOP-IGA were compared, and the comparison results are shown in Table 5.

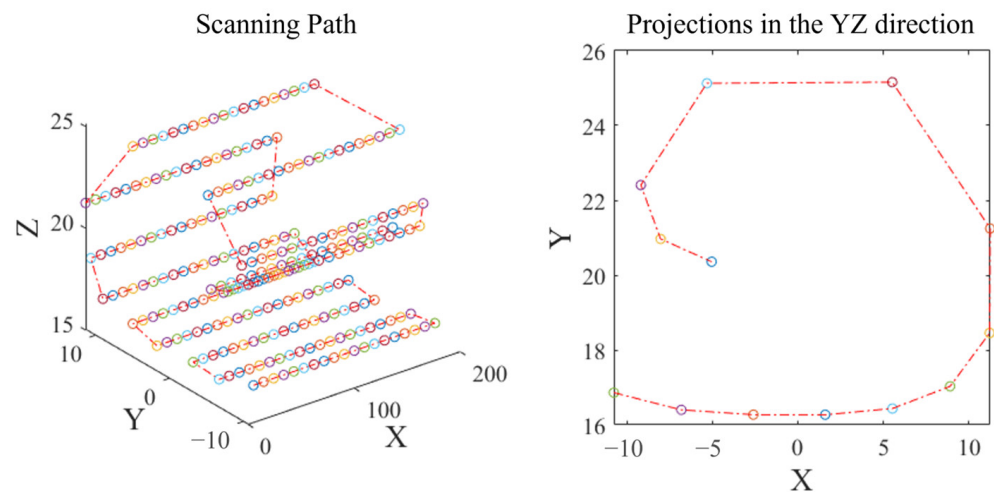


Figure 16. Optimal detection paths for conformal antenna derived by CHOP-IGA.

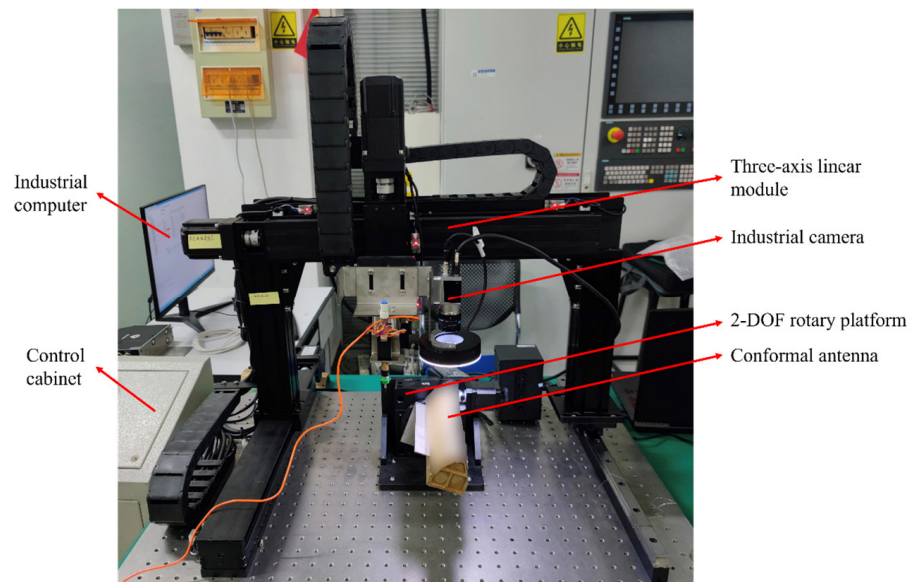


Figure 17. The test platform.

Table 5. Comparison of scanning time.

	Time Spent on Scanning/s	Improvements in Efficiency
Original path	230.18	
Optimized path	118.67	48.44%

As can be seen from the results, the path optimized by the algorithm saved the time spent on feature leveling and improved the scanning efficiency by 48.44%, while also providing a guide for the path planning of AOI, coating, mounting, and welding.

### 5. Conclusions

To address the shortcomings of the TGA when solving the path planning problem for conformal antenna surface detection, this paper proposes an improved genetic algorithm combining the historical optimal population, which improves several key steps of the genetic algorithm. Simulation experiments showed that the algorithm not only solved medium- and small-scale two-dimensional standard TSPs better, but also that it was successfully applied to the detection path planning problem of the conformal antenna.



Further improvements in this paper include the following: the current algorithm only includes mutation operation for the historical optimal population to reach a better solution, and it is necessary to consider how to more fully utilize the information of the good gene fragments contained in the historical optimal population; the current algorithm performs well in solving medium- and small-sized problems, and how to solve larger-scale problems is also a key question; at present, in the path planning problem for conformal antenna surface detection, only a single constraint is considered, and how to construct a path planning problem with multiple constraints needs to be considered.

**Author Contributions:** Conceptualization, W.T. and C.W.; validation, W.T. and C.W.; investigation, Y.D., K.L., Z.W. and X.D.; writing—original draft preparation, Y.D., K.L. and Z.W.; writing—review and editing, W.T., C.W. and X.D.; supervision, W.T., C.W. and X.D.; project administration, X.D. and C.D.; funding acquisition, X.D. and C.D. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National key R & D program of China (No. 2020YFB1710300) and the National Defense Basic Scientific Research Program of China (No. JCKY2019210B004).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data are contained within the article. The TSPLIB library used in the article is a publicly accessible library that can be accessed and downloaded at <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/>, accessed on 18 July 2023.

**Acknowledgments:** The authors would like to acknowledge the financial support from the National key R & D program of China (No. 2020YFB1710300) and the National Defense Basic Scientific Research Program of China (No. JCKY2019210B004).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Li, M.; Cai, Z.Y.; Li, Y.; Chen, H. Continuous evaluation method of outfield reliability for airborne electronic countermeasure system. *Electron. Opt. Control* **2022**, *29*, 84–87.
- Li, R.L.; Niu, Z.Y.; Lin, R.S. A novel method for the RCS reduction of conformal microstrip antenna. In Proceedings of the 2011 Cross Strait Quad-Regional Radio Science and Wireless Technology Conference, Harbin, China, 26–30 July 2011; IEEE Press: Piscataway, MA, USA, 2011; pp. 516–519.
- Liu, Z.Q.; Zhang, Y.S.; Qian, Z.P.; Han, Z.P.; Ni, W.M. A novel broad beamwidth conformal antenna on unmanned aerial vehicle. *IEEE Antennas Wirel. Propag. Lett.* **2012**, *11*, 196–199. [[CrossRef](#)]
- Li, X.; Xie, Y. Key process control technology for airborne conformal antenna manufacturing. *Electron. World* **2019**, *2*, 150–151.
- Bellman, R. Dynamic programming treatment of the travelling salesman problem. *J. ACM* **1962**, *9*, 61–63. [[CrossRef](#)]
- Poikonen, S.; Golden, B.; Wasil, E.A. A branch-and-bound approach to the traveling salesman problem with a drone. *INFORMS J. Comput.* **2019**, *31*, 335–346. [[CrossRef](#)]
- Qiao, S.; Lv, Z.M.; Zhang, N. Improved particle swarm optimization algorithm based on Hamming distance for traveling salesman problem. *J. Comput. Appl.* **2017**, *37*, 2767–2772.
- Skinderowicz, R. Improving ant colony optimization efficiency for solving large TSP instances. *Appl. Soft Comput.* **2022**, *120*, 108653. [[CrossRef](#)]
- Wang, L.J.; Cai, R.Y.; Lin, M.; Zhong, Y.W. Enhanced list-based simulated annealing algorithm for large-scale traveling salesman problem. *IEEE Access* **2019**, *7*, 144366–144380. [[CrossRef](#)]
- Zhang, Z.; Han, Y. Discrete sparrow search algorithm for symmetric traveling salesman problem. *Appl. Soft Comput.* **2022**, *118*, 108469. [[CrossRef](#)]
- Sheng, W.S.; Xu, A.P.; Xu, L.J. Simulation of traveling salesman path planning based on ant colony algorithm and genetic algorithm. *Comput. Simul.* **2022**, *39*, 398–402+412.
- Katoch, S.; Chauhan, S.S.; Kumar, V. A review on genetic algorithm: Past, present, and future. *Multimed. Tools Appl.* **2021**, *80*, 8091–8126. [[CrossRef](#)] [[PubMed](#)]
- Xu, J.; Han, F.Q.; Liu, Q.X.; Xue, X.X. Bioinformation heuristic genetic algorithm for solving TSP. *J. Syst. Simul.* **2022**, *34*, 1811–1819.
- Arram, A.; Ayob, M. A novel multi-parent order crossover in genetic algorithm for combinatorial optimization problems. *Comput. Ind. Eng.* **2019**, *133*, 267–274. [[CrossRef](#)]
- Zhang, P.L.; Wang, J.Q.; Tian, Z.W.; Sun, S.Z.; Li, J.T.; Yang, J.N. A genetic algorithm with jumping gene and heuristic operators for traveling salesman problem. *Appl. Soft Comput.* **2022**, *127*, 109339. [[CrossRef](#)]

16. Rokbani, N.; Kumar, R.; Abraham, A.; Alimi, A.M.; Long, H.V.; Priyadarshini, I.; Son, L.H. Bi-heuristic ant colony optimization-based approaches for traveling salesman problem. *Soft Comput.* **2021**, *25*, 3775–3794. [[CrossRef](#)]
17. Akter, S.; Nahar, N.; ShahadatHossain, M.; Andersson, K. A new crossover technique to improve genetic algorithm and its application to TSP. In Proceedings of the 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE), Cox's Bazar, Bangladesh, 7–9 February 2019; IEEE: Piscataway, MA, USA, 2019; pp. 1–6.
18. Ahmed, Z.H. Improved genetic algorithms for the travelling salesman problem. *Int. J. Process Manag. Benchmarking* **2014**, *4*, 109–124. [[CrossRef](#)]
19. Xu, R.C. A comparative study of genetic algorithms for solving TSP problems. *J. Sichuan Univ. Sci. Eng. Nat. Sci. Ed.* **2019**, *32*, 71–78.
20. Deng, W.L.; Hu, G.W. Discrete particle swarm optimization algorithm for TSP. *Comput. Mod.* **2012**, *19*, 1–4.
21. Zhao, H.; Zhu, J.; Zhu, J.; Li, W.R. Gene-level population diversity mathematical model of real-coded GA. *J. Cent. South Univ. Sci. Technol.* **2015**, *46*, 894–900.
22. Wu, G.H.; Ma, M.H. Improve solution of TSP based on route-intersection inspection and elimination method and neighbor nodes interchanging method. *Appl. Res. Comput.* **2011**, *28*, 485–487.
23. Hassanat, A.; Almohammadi, K.; Alkafaween, E.; Abunawas, E.; Hammouri, A.; Prasath, V.B.S. Choosing mutation and crossover ratios for genetic algorithms—A review with a new dynamic approach. *Information* **2019**, *10*, 390. [[CrossRef](#)]
24. Li, M.J. Theory and Applications of Partheno-Genetic Algorithm. Ph.D. Thesis, Hunan University, Changsha, China, 2002.
25. Wang, D.; Gui, W.X. A multi-population genetic algorithm based on cross accessibility evaluation. *J. Guangxi Univ. Nat. Sci. Ed.* **2015**, *40*, 1508–1516.
26. Akhand, M.A.H.; Ayon, S.I.; Shahriyar, S.A.; Siddique, N.; Adeli, H. Discrete spider monkey optimization for travelling salesman problem. *Appl. Soft Comput.* **2020**, *86*, 105887. [[CrossRef](#)]
27. Liu, J.; Lan, J.L.; Li, D. Annealing simulation algorithms for three-dimensional constrained TSP. *J. Univ. Electron. Sci. Technol. China* **1992**, *3*, 241–246.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.