applied sciences

Article

Path Optimization of Two-Posture Manipulator of Apple Packing Robots

Rong Xiang * and **Binbin Feng**

College of Quality and Standardization, China Jiliang University, No. 258 Xueyuan Street, Higher Education Zone of Xiasha, Hangzhou 310018, China; p23060855010@cjlu.edu.cn
* Correspondence: xr_rongge@cjlu.edu.cn

Featured Application: Automatic packaging of fruit grading production line.

Abstract: Automated packing is urgently needed in apple production. This paper proposes an improved genetic algorithm fused with an optimal parameter selection algorithm to optimize the two-posture manipulator working path of packing robots. First, the structure and working principle of the packing robot were designed. Second, the kinematics and packing paths of the two-posture manipulator were analyzed. Finally, the path optimization method for the two-posture manipulator was introduced. The method was based on the improved genetic algorithm by using a two-level coding and region crossover operator. The parameter values can be automatically determined by the optimal parameter selection algorithm. Ten repeated comparative tests show that the total packing time is 23.86 s under the working conditions of four grasping points and fourteen placing points. The optimal performance of the proposed algorithm is better than that of the traditional genetic algorithm, and the average optimization amplitudes are 14.63%, 15.42%, 16.24%, and 13.82% for 9-groove, 12-groove, 14-groove, and 16-groove trays, respectively. The proposed algorithm can effectively prevent the premature convergence problem of the traditional genetic algorithm and the optimization process instability problem, improve the range of optimization, and reduce the manipulator working time during packing.

Keywords: apple; packing robot; manipulator; path optimization; genetic algorithm; parameter optimization

Citation: Xiang, R.; Feng, B. Path Optimization of Two-Posture Manipulator of Apple Packing Robots. *Appl. Sci.* **2024**, *14*, 8849. https://doi.org/10.3390/app14198849

Academic Editor: Juan A. Gómez-Pulido

Received: 30 August 2024
Revised: 27 September 2024
Accepted: 30 September 2024
Published: 1 October 2024

1. Introduction

Fruit production is time- and labor-consuming. Apple production processes need to be automated and intelligent to optimize and upgrade the apple industry chain [1]. The automation of several processes has been investigated, and several types of robots, such as harvesting [2] and grading [3] robots, have been proposed and applied in actual production. As an important part in apple production, packing mainly depends on manual operations and becomes a bottleneck for the entire process of automation in apple production. Therefore, automated packing is urgently needed. The automatic packing of apples mainly includes the following steps: the position information of grasping and placing points is automatically obtained; manipulators circularly move apples from the grasping points to the placing points in accordance with the position information; and manipulators return to the starting point when one apple tray is full and wait to start the next loop.

In the process of moving the manipulator, there are multiple starting points and multiple ending points. The manipulator moves back and forth from the grasping points to the placing points until the placing points are all loaded with apples. In this process, the reasonable planning of its movement trajectory is very important to improve the efficiency of apple packing.

This paper proposes a path optimization method for a designed two-link manipulator using an improved genetic algorithm fused with a two-level chromosome coding algorithm and an optimal parameter selection algorithm. The purposes were (1) to realize the path optimization of two-posture manipulator, (2) to realize automatic selection of optimal parameters of the genetic algorithm, and (3) to improve the optimization amplitude of the genetic algorithm.

The innovation points of this study are as follows: an improved genetic algorithm based on two-level chromosome coding is proposed for the path optimization of manipulators moving from multi-fixed points to multi-varying points and with dual postures, and an optimal parameter combination selection method is proposed for the improved genetic algorithm.

2. Literature Background

As the core part of a robot, manipulators are consistently investigated in agricultural robots. Since the 1980s, researchers have conducted several studies on the structural design of manipulators for grasping and picking fruits and vegetables in Japan, the Netherlands, the UK, France, and other countries [4–7]. Li et al. [8] developed a manipulator with multi-end effectors for grasping various spherical fruits. Fu et al. [9] and Mu et al. [10] studied the manipulator of a picking robot safely separating kiwifruit from fruit stalks under natural growth conditions. Bulanon et al. [11] conducted research on a manipulator positioning and grasping apples based on visual detection. Xiong et al. [12] designed a new robot for picking strawberries, and the picking tongs of this robot come with a fruit container that can effectively reduce the traveling time for the manipulator between each berry and a separate punnet.

Packaging efficiency is one of the main indexes used to evaluate the performance of apple automatic packaging, and the path optimization of the mechanical arm for packaging is the basis for ensuring the efficiency of apple automatic packaging. Path optimization of manipulators has been widely investigated in recent years to smoothly grasp and place objects and improve the work efficiency [13–18]. Methods for path optimization include algorithms for optimal path planning in discrete spaces [19,20], optimal path planning in continuous spaces [21], and trajectory generation in continuous spaces with constraints [22]. Li et al. [23] studied piecewise polynomial interpolation in trajectory planning by setting constraint conditions for the trajectory equation to realize smooth operation for the connecting rod of excavators. However, the computational amount is relatively large because of many trajectory equations, and the solution efficiency is low. Reynoso et al. [24] investigated the friction problem in trajectory planning and proposed a discretization solution that plays a good role in vibration reduction and stability. However, the calculation amount is extremely large. The method based on the trajectory equation has a large amount of calculation.

Genetic algorithms have the advantage of wide applicability, strong global search ability, strong parallelism, and no need for derivative information. In the path optimization of the manipulator of apple packing robots, the combined optimization of multiple factors must be considered, such as the grasping position, placing position, and posture of the manipulator, and the advantage of the strong global search ability of genetic algorithms is very suitable. The solution process of the genetic algorithm is simpler than that of the trajectory equation. Ju et al. [25] used the particle swarm optimization and genetic algorithm to plan the set path. The test results showed that the optimized performance of the genetic algorithm is better than that of the particle swarm optimization. However, the genetic algorithm is prone to premature convergence when the number of path points and data are large [26,27]. He et al. [28] used an improved genetic algorithm to realize the path optimization for automatic transplanting of tray seedlings, thereby reducing the operating time and improving the transplanting efficiency. Although the operation of reinserting offspring individuals in every iteration increases the possibility of chromosome optimization, it reduces the rapid convergence of the algorithm. Chaymaa et al. [29] improved the premature phenomenon of the genetic algorithm by enhancing the crossover operator and fitness function, but the application environment of the operator was relatively limited. The main problems of the current genetic algorithms are that they easily fall into a premature state and the optimization process is not stable [30].

Moreover, in contrast to the existing multi-path optimization methods applied to one fixed point to one fixed point, one fixed point to multiple varying points, and multiple varying points to multiple varying points, the problem to be solved in this study is the overall optimal solution of multiple paths under multiple fixed points to multiple varying points.

In addition, the parameter setting of the current genetic algorithms is mostly based on empirical values. The parameters set in this method are not precise, which affects the optimization effect.

It must be pointed out that the current genetic algorithm is aimed at the path optimization of the single-state manipulator, that is, there is only one path for objects with the same starting point and ending point. The manipulator designed in this study is a two-posture manipulator, that is, there are two packing paths for each apple with the same starting point and ending point.

3. Materials and Methods

3.1. Materials

The main device is the manipulator of the apple packing robot and its control system; the detailed content is shown in Sections 3.2.1 and 3.2.2. The processing material of the manipulator is aluminum alloy 6061. The motor model is 86HBP98AL4 (Corenergy Technology Limited, Zhangjiagang, China). The microcontroller unit of the manipulator is an STM32F103 (STMicroelectronics, Shanghai, China). The computer is an Asus K550V I5-3230m (Asus, Hangzhou, China). Trays were purchased online.

The variety name of apples used in the experiment is "Guoguang", which is grown in China.

3.2. Path Optimization Method for the Manipulator of Apple Packing Robots

3.2.1. Working Principle of Packing Robot

An apple packing robot is mainly composed of four parts, namely, main control, visual detection, apple tray transmission, and manipulator units. The schematic of the packing robot is shown in Figure 1.

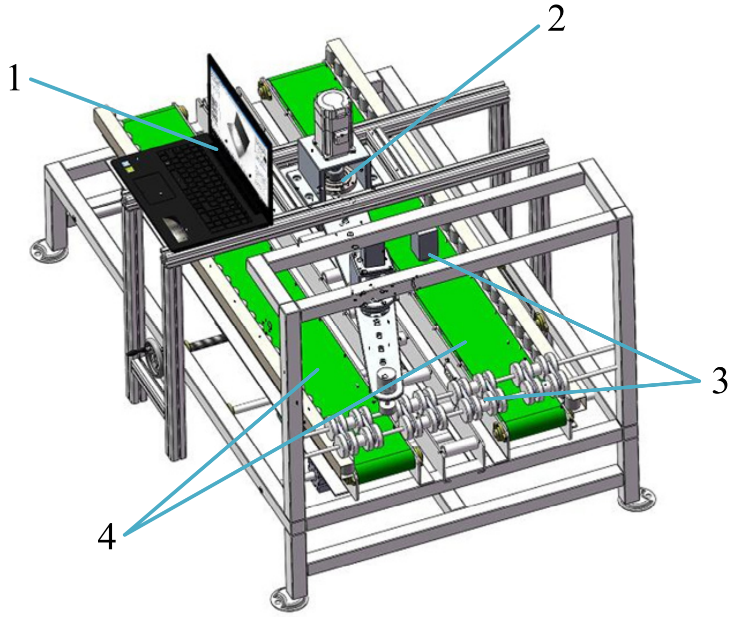


Figure 1. Apple packing robot: (1) Main control unit, computer (2) Manipulator unit, including motor, manipulator, and end effector (3) Visual detection unit, including camera and positioning mechanism (synchro wheel) (4) Apple tray transmission unit, conveyor belts.

Automatic apple packing is shown as follows. First, the fixed grasping and placing points are detected by the visual detection unit. Second, apples on grasping and placing points are detected through image processing methods, and detection results are sent to the main control unit. Third, the main control unit controls the manipulator in transferring apples from the grasping points to the vacant placing points. Grasping points are those positions where apples waiting to be grasped are located after the transportation is completed, and placing points are the grooves on apple trays. The transfer will continue when all grooves on one tray are filled with apples. The tray is then transferred to the fixed position.

3.2.2. Kinematic Analysis of the Manipulator

The manipulator is a self-made two-link manipulator, as shown in Figure 2.

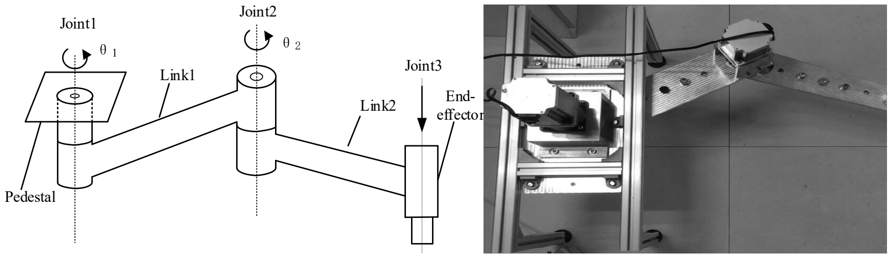


Figure 2. Manipulator of the apple packing robot.

As shown in Figure 2, Joint 1 of the manipulator is fixed at the center of the pedestal, enabling Link 1 of the manipulator to rotate horizontally with the central axis of the pedestal, which is regarded as the datum line within the rotation range. Links 1 and 2 are connected by Joint 2. Link 2 can also rotate horizontally with the axis of rotating Joint 2, regarded as the datum line, which is similar to Link 1. The sliding of the end effector of the manipulator is accomplished with the reciprocating movement of the cylinder. The manipulator has two rotatory joints in the horizontal direction and one sliding joint in the vertical direction. The combination of the three joints enables the manipulator to move freely in the 3D coordinate system (x, y, z). This study focuses on the optimization of the motion trajectory of the manipulator, and it does not involve the motion of the gripper device. The design of the gripper device, and the calculation and analysis of the gripping force, will be carried out in future work.

The kinematics of the manipulator are analyzed to clearly describe the position relationship between the end effector and each link [31,32]. Kinematic analysis is the premise and basis of the joint motion and path optimization of the manipulator. The kinematics of the manipulator are analyzed using a geometric method because the number of links of the manipulator is small, and the rotatory joints are all in the horizontal direction. A rectangular coordinate system is established, with the central point of the pedestal as the origin, and the geometric analysis model is shown in Figure 3.

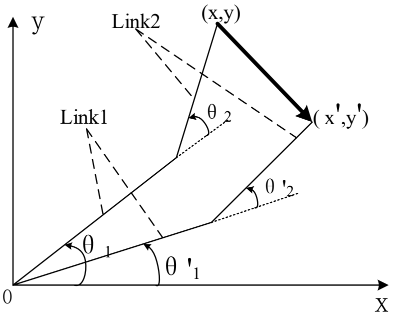


Figure 3. Geometric analysis model.

As shown in Figure 3, the equation for acquiring the coordinates (x, y) of the end effector of the manipulator is shown in Equation (1).

$$\begin{cases} x = link2 \cdot cos(\theta_1 + \theta_2) + link1 \cdot cos\theta_1 \\ y = link2 \cdot sin(\theta_1 + \theta_2) + link1 \cdot sin\theta_1 \end{cases}, \tag{1}$$

where θ_1 and θ_2 are the joint rotation angles of Links 1 and 2, respectively, and *link*1 and *link*2 are the lengths of Links 1 and 2. For the convenience of calculation, joint rotation angles θ_1 and θ_2 are positive when the arm rotates counterclockwise and are negative when the arm rotates clockwise. Every rotatory joint has two different postures when the end effector of the manipulator reaches any point within the rotation range because the manipulator is a horizontal rotating structure, as shown in Figure 4. The shape of the apple is approximately spherical, and it is approximately circular in all directions in the horizontal plane. The resulting change in end effector orientation does not affect the grasp of the apple when the manipulator is in different postures. Therefore, two values of θ_1 and θ_2 are found for the coordinates of any points in the rotation range, as shown in Equation (2).

$$\begin{cases} \theta_1 = across\left[\frac{x \cdot link1 + x \cdot link2 \cdot cos\theta_2 + y \cdot link2 \cdot sin\theta_2}{x^2+y^2}\right] \\ \theta_2 = \pm arccos\left[\frac{x^2+y^2-link1^2-link2^2}{2 \cdot link1 \cdot link2}\right] \end{cases}, \tag{2}$$

where (x, y) are the coordinates of the end effector within the rotation range of the manipulator. Based on the actual working range, the range of θ_1 is [30°, 150°] and that of θ_2 is [−120°, 120°]. The length of the two-link horizontal manipulator is 750 mm. The lengths of *link*1 and *link*2 are 400 and 350 mm, respectively.

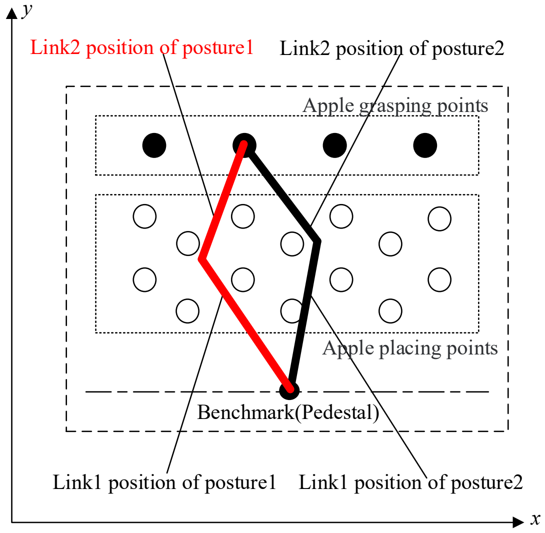

y
Link2 position of posture1
Link2 position of posture2
Apple grasping points
Apple placing points
Benchmark(Pedestal)
Link1 position of posture1
Link1 position of posture2
x

Figure 4. Two different postures of the manipulator at one point.

3.2.3. Packing Path Analysis of the Manipulator

Apple trays are commonly used in automatic apple packing robots in fruit markets. Different types of trays have 9, 12, 14, or 16 grooves, as shown in Figure 5. The number of placing points is equal to the number of grooves of one tray, and the number of grasping points is equal to the channel number of packing robots. In this study, 4 apple grasping points and 14 apple placing points are taken as an example to explain the principle of path optimization. The external size of the plastic tray with 14 grooves is 470 mm × 280 mm. Figure 6 shows an example of a packing path. In Figure 6, 4 black solid circles represent 4 grasping points, and 14 white hollow circles represent 14 grooves of the tray, that is, 14 placing points. The positions of all 18 points are fixed in the automatic packing robot. On the premise of continuous supply of apples at every grasping point, the end effector starts from the initial position, grasps apples at grasping points, and places them in tray grooves. The end effector returns to the initial position when all the grooves of a tray are filled. The initial position is defined as the position of the end effector when the two links of the manipulator naturally extend forward in the front direction of the pedestal, which is labeled as 0 in Figure 6. Taking the datum of the central point of the pedestal as the origin, the coordinate of the initial position is (0, 750) mm.

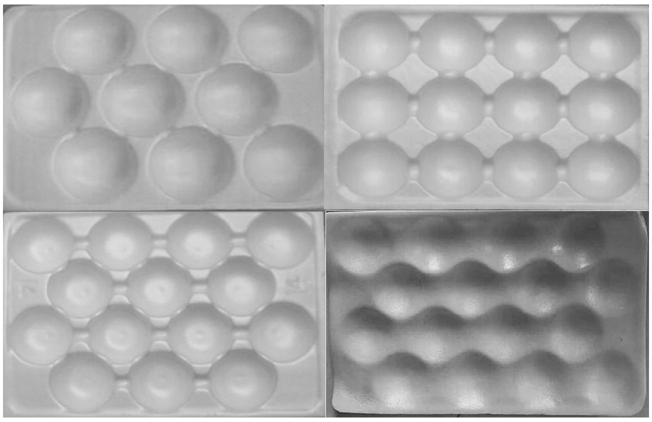


Figure 5. Examples of common apple trays.

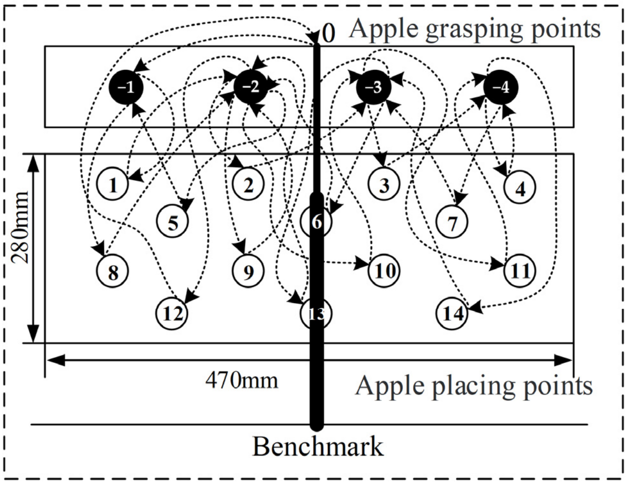


Figure 6. Automatic apple packing points and sample route.

The number of random paths of automatic apple packing is $n! \times (m^n) \times (2^{2n})$, which is 6.28×10^{27} when the number of apple grasping points is m (m is an integer within the working range of the manipulator, which is 4) and the number of apple placing points is n ($0 \leq n \leq G$, which is an integer with a value of 14). Choosing the optimal path through the traversal method is difficult because the packing path is complex and changeable, and the computation of traversing all paths is extremely large. We proposed a path optimization method for the manipulator using a genetic algorithm and an optimal parameter selection algorithm to accurately determine the optimal path for automatic packing.

3.2.4. Packing Path Optimization Algorithm

The packing path optimization algorithm is realized using an improved genetic algorithm fused with an optimal parameter selection algorithm. The improved genetic algorithm mainly includes the following steps: definition of chromosome generation rule, fitness function setting, crossover operation, mutation operation, comparison, and substitution operation. An optimal parameter selection algorithm is proposed to improve the optimization results of the genetic algorithm. The improved genetic algorithm fused with the optimal parameter selection algorithm is shown in Figure 7a, and the improved genetic algorithm is shown in Figure 7b. The parameter combination includes four parameters, namely, population size, crossover probability, mutation probability, and comparison substitution coefficient.

As shown in Figure 7a, the improved genetic algorithm fused with the optimal parameter selection algorithm shows the running time of apple packing under different parameter combinations. The optimal parameter selection algorithm is a fitting process between the results of the improved genetic algorithm and the corresponding parameter combinations. Then, the optimal parameter combination can be determined using an exhaustive method with a fitting equation. The optimal packing path can be acquired by rerunning the improved genetic algorithm on the basis of the selected optimal parameter combination.

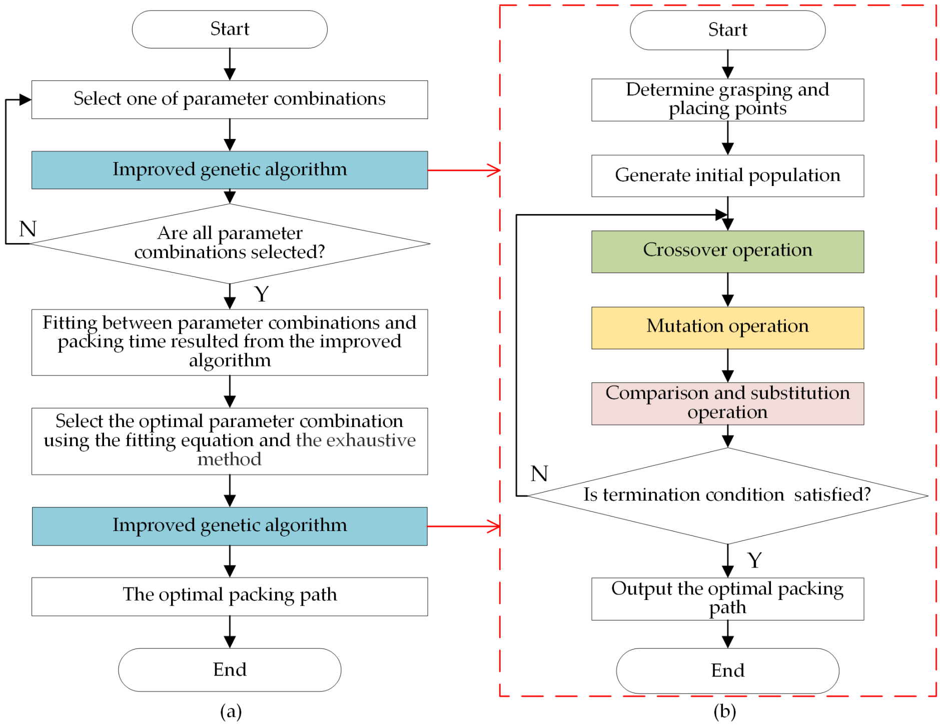

Start
Select one of parameter combinations
Improved genetic algorithm
N
Are all parameter combinations selected?
Y
Fitting between parameter combinations and packing time resulted from the improved algorithm
Select the optimal parameter combination using the fitting equation and the exhaustive method
Improved genetic algorithm
The optimal packing path
End
(a)
Start
Determine grasping and placing points
Generate initial population
Crossover operation
Mutation operation
Comparison and substitution operation
N
Is termination condition satisfied?
Y
Output the optimal packing path
End
(b)

Figure 7. Flowchart of the improved genetic algorithm fused with the optimal parameter selection algorithm: (**a**) flowchart of the improved genetic algorithm fused with the optimal parameter selection algorithm; (**b**) flowchart of the improved genetic algorithm.

Improved Genetic Algorithm

The main steps of the improved genetic algorithm include chromosome coding and population creation, fitness function design, region crossover operation, mutation operation, comparison and substitution operation, and termination condition setting. Chromosome coding and population creation is used to code every path composed of different grasping points, placing points, and manipulator postures to one chromosome. All chromosomes corresponding to all paths formed one population. The fitness function is designed to denote the quality of every population. The region crossover operation (an example is shown in Figure 8) is used to enrich the diversity of offspring. The mutation operation (an example is shown in Figure 9) guarantees the randomness of various posture solutions of the manipulator. The comparison and substitution operation (an example is shown in Figure 10) can select the chromosome with the optimal fitness function value and replace the inferior one with it. The termination condition is set to stop the iteration.

(1) Two-level chromosome coding and population creation

Before path optimization, grasping points with apples and placing points without apples are labeled in order. Apple grasping points are labeled with negative values from left to right ($-1, -2, -3, -4, \ldots, -m$), and apple placing points are labeled with positive values from left to right and from top to bottom ($1, 2, 3, \ldots, n$), as shown in Figure 6. The coordinates of each point within the rotation range are mapped to the coordinates in the visual detection system.

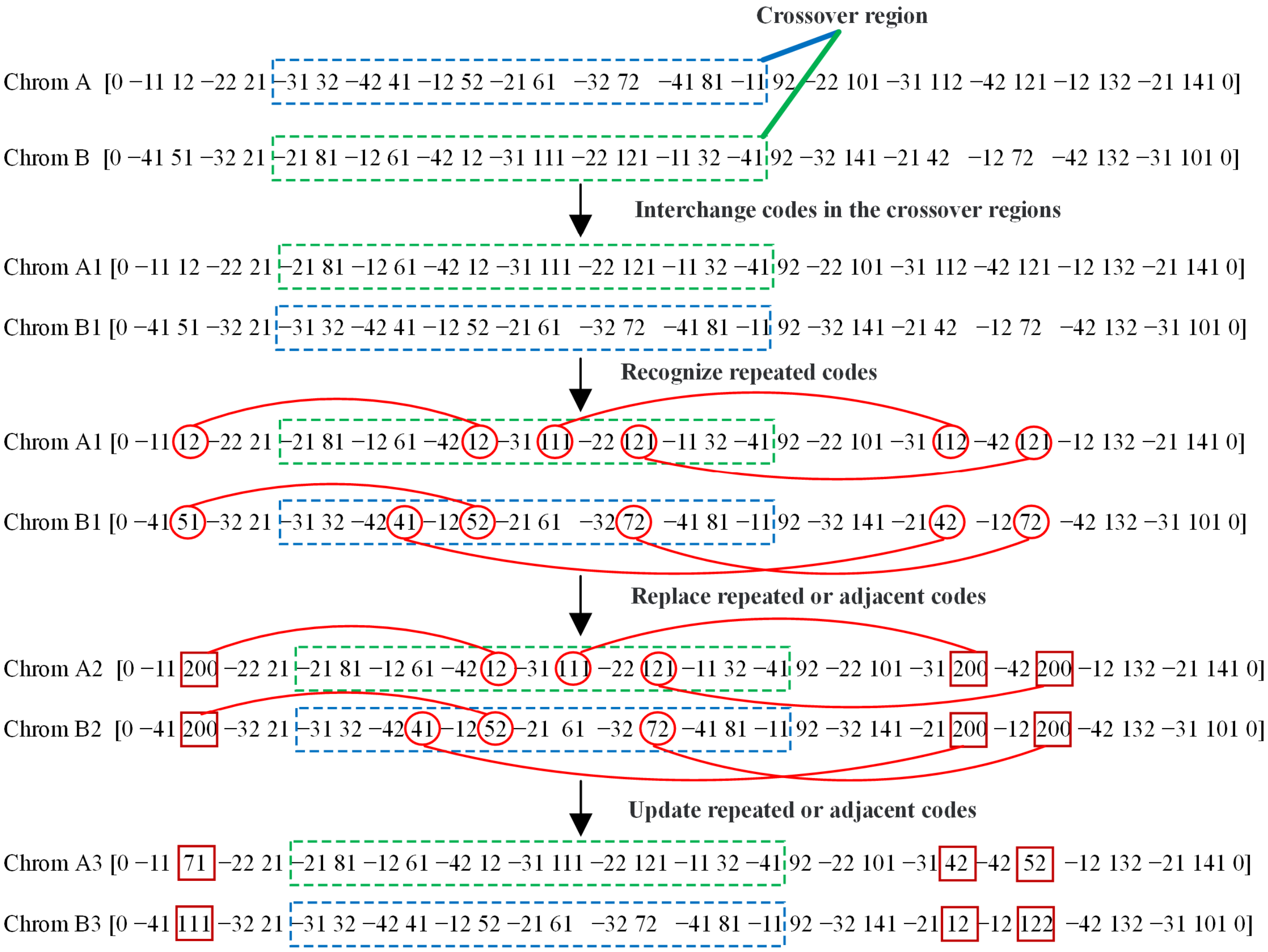

Crossover region
Chrom A [0 −11 12 −22 21 −31 32 −42 41 −12 52 −21 61 −32 72 −41 81 −11 92 −22 101 −31 112 −42 121 −12 132 −21 141 0]
Chrom B [0 −41 51 −32 21 −21 81 −12 61 −42 12 −31 111 −22 121 −11 32 −41 92 −32 141 −21 42 −12 72 −42 132 −31 101 0]
Interchange codes in the crossover regions
Chrom A1 [0 −11 12 −22 21 −21 81 −12 61 −42 12 −31 111 −22 121 −11 32 −41 92 −22 101 −31 112 −42 121 −12 132 −21 141 0]
Chrom B1 [0 −41 51 −32 21 −31 32 −42 41 −12 52 −21 61 −32 72 −41 81 −11 92 −32 141 −21 42 −12 72 −42 132 −31 101 0]
Recognize repeated codes
Chrom A1 [0 −11 12 −22 21 −21 81 −12 61 −42 12 −31 111 −22 121 −11 32 −41 92 −22 101 −31 112 −42 121 −12 132 −21 141 0]
Chrom B1 [0 −41 51 −32 21 −31 32 −42 41 −12 52 −21 61 −32 72 −41 81 −11 92 −32 141 −21 42 −12 72 −42 132 −31 101 0]
Replace repeated or adjacent codes
Chrom A2 [0 −11 200 −22 21 −21 81 −12 61 −42 12 −31 111 −22 121 −11 32 −41 92 −22 101 −31 200 −42 200 −12 132 −21 141 0]
Chrom B2 [0 −41 200 −32 21 −31 32 −42 41 −12 52 −21 61 −32 72 −41 81 −11 92 −32 141 −21 200 −12 200 −42 132 −31 101 0]
Update repeated or adjacent codes
Chrom A3 [0 −11 71 −22 21 −21 81 −12 61 −42 12 −31 111 −22 121 −11 32 −41 92 −22 101 −31 42 −42 52 −12 132 −21 141 0]
Chrom B3 [0 −41 111 −32 21 −31 32 −42 41 −12 52 −21 61 −32 72 −41 81 −11 92 −32 141 −21 12 −12 122 −42 132 −31 101 0]

Figure 8. Process of the crossover operator. The red circles indicate the positive numbers in the unchanged and changed regions that have the same or adjacent numbers. The red boxes indicate the changed positive numbers in the unchanged regions that have the same or adjacent numbers as the positive numbers of the changed regions.

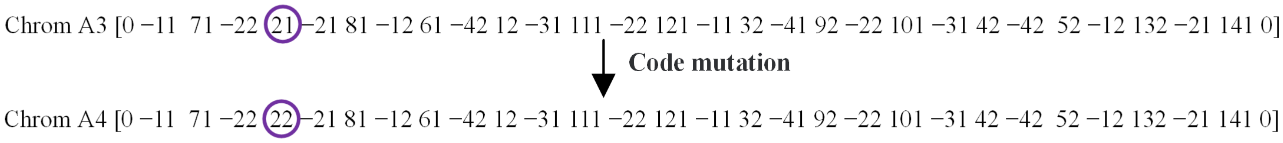
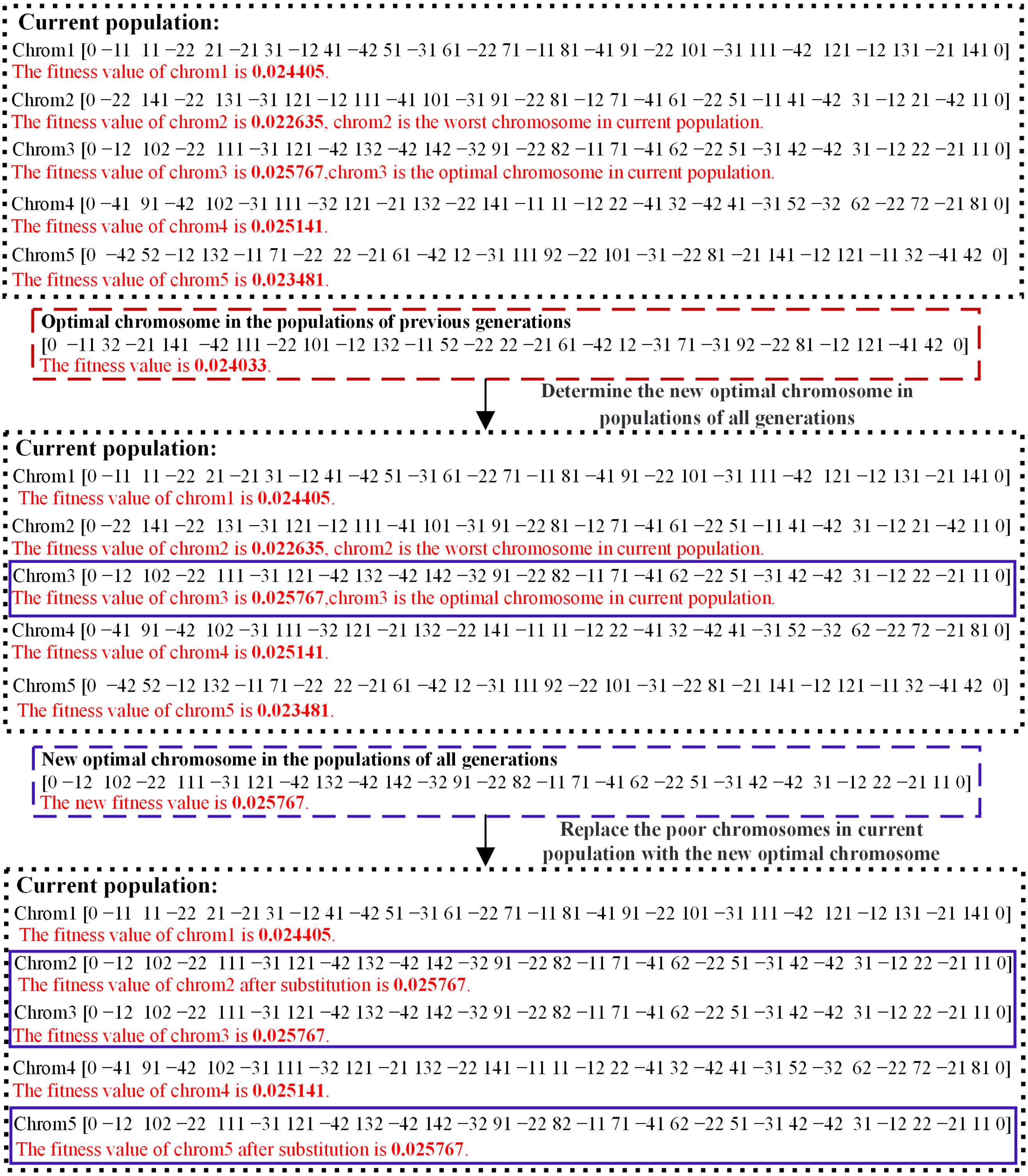

Chrom A3 [0 −11 71 −22 21 −21 81 −12 61 −42 12 −31 111 −22 121 −11 32 −41 92 −22 101 −31 42 −42 52 −12 132 −21 141 0]
Code mutation
Chrom A4 [0 −11 71 −22 22 −21 81 −12 61 −42 12 −31 111 −22 121 −11 32 −41 92 −22 101 −31 42 −42 52 −12 132 −21 141 0]

Figure 9. Process of the mutation operator. The purple circles indicate the posture changed by the mutation operator.

Two types of rotational joint angle solutions are found for each point within the rotation range of the manipulator. Therefore, the manipulator has two different postures at each point. The two postures of each point are labeled as two adjacent real integers. When the point number of the end effector is −2, the two postures of this point are labeled as −21 and −22, respectively. Grasping points and postures are relabeled with negative values from left to right, which are −11, −12, −21, −22, −31, −32, −41, −42, . . ., −*m*1, −*m*2. Similarly, placing points and postures are labeled with positive values, which are 11, 12, 21, 22, 31, 32, . . ., *n*1, *n*2. The original grasping matrix is recorded as *Grasp*, where *Grasp* = [−11 −12 −21 −22 −31 −32 −41 −42 . . . −*m*1 −*m*2]. The original placing matrix is recorded as *Pos*, where *Pos* = [11 12 21 22 31 32 . . . *n*1 *n*2].



Current population:

Chrom1 [0 −11 11 −22 21 −21 31 −12 41 −42 51 −31 61 −22 71 −11 81 −41 91 −22 101 −31 111 −42 121 −12 131 −21 141 0]
The fitness value of chrom1 is **0.024405**.

Chrom2 [0 −22 141 −22 131 −31 121 −12 111 −41 101 −31 91 −22 81 −12 71 −41 61 −22 51 −11 41 −42 31 −12 21 −42 11 0]
The fitness value of chrom2 is **0.022635**, chrom2 is the worst chromosome in current population.

Chrom3 [0 −12 102 −22 111 −31 121 −42 132 −42 142 −32 91 −22 82 −11 71 −41 62 −22 51 −31 42 −42 31 −12 22 −21 11 0]
The fitness value of chrom3 is **0.025767**,chrom3 is the optimal chromosome in current population.

Chrom4 [0 −41 91 −42 102 −31 111 −32 121 −21 132 −22 141 −11 11 −12 22 −41 32 −42 41 −31 52 −32 62 −22 72 −21 81 0]
The fitness value of chrom4 is **0.025141**.

Chrom5 [0 −42 52 −12 132 −11 71 −22 22 −21 61 −42 12 −31 111 92 −22 101 −31 −22 81 −21 141 −12 121 −11 32 −41 42 0]
The fitness value of chrom5 is **0.023481**.

Optimal chromosome in the populations of previous generations
[0 −11 32 −21 141 −42 111 −22 101 −12 132 −11 52 −22 22 −21 61 −42 12 −31 71 −31 92 −22 81 −12 121 −41 42 0]
The fitness value is **0.024033**.

Determine the new optimal chromosome in populations of all generations

Current population:

Chrom1 [0 −11 11 −22 21 −21 31 −12 41 −42 51 −31 61 −22 71 −11 81 −41 91 −22 101 −31 111 −42 121 −12 131 −21 141 0]
The fitness value of chrom1 is **0.024405**.

Chrom2 [0 −22 141 −22 131 −31 121 −12 111 −41 101 −31 91 −22 81 −12 71 −41 61 −22 51 −11 41 −42 31 −12 21 −42 11 0]
The fitness value of chrom2 is **0.022635**, chrom2 is the worst chromosome in current population.

Chrom3 [0 −12 102 −22 111 −31 121 −42 132 −42 142 −32 91 −22 82 −11 71 −41 62 −22 51 −31 42 −42 31 −12 22 −21 11 0]
The fitness value of chrom3 is **0.025767**,chrom3 is the optimal chromosome in current population.

Chrom4 [0 −41 91 −42 102 −31 111 −32 121 −21 132 −22 141 −11 11 −12 22 −41 32 −42 41 −31 52 −32 62 −22 72 −21 81 0]
The fitness value of chrom4 is **0.025141**.

Chrom5 [0 −42 52 −12 132 −11 71 −22 22 −21 61 −42 12 −31 111 92 −22 101 −31 −22 81 −21 141 −12 121 −11 32 −41 42 0]
The fitness value of chrom5 is **0.023481**.

New optimal chromosome in the populations of all generations
[0 −12 102 −22 111 −31 121 −42 132 −42 142 −32 91 −22 82 −11 71 −41 62 −22 51 −31 42 −42 31 −12 22 −21 11 0]
The new fitness value is **0.025767**.

Replace the poor chromosomes in current population with the new optimal chromosome

Current population:

Chrom1 [0 −11 11 −22 21 −21 31 −12 41 −42 51 −31 61 −22 71 −11 81 −41 91 −22 101 −31 111 −42 121 −12 131 −21 141 0]
The fitness value of chrom1 is **0.024405**.

Chrom2 [0 −12 102 −22 111 −31 121 −42 132 −42 142 −32 91 −22 82 −11 71 −41 62 −22 51 −31 42 −42 31 −12 22 −21 11 0]
The fitness value of chrom2 after substitution is **0.025767**.

Chrom3 [0 −12 102 −22 111 −31 121 −42 132 −42 142 −32 91 −22 82 −11 71 −41 62 −22 51 −31 42 −42 31 −12 22 −21 11 0]
The fitness value of chrom3 is **0.025767**.

Chrom4 [0 −41 91 −42 102 −31 111 −32 121 −21 132 −22 141 −11 11 −12 22 −41 32 −42 41 −31 52 −32 62 −22 72 −21 81 0]
The fitness value of chrom4 is **0.025141**.

Chrom5 [0 −12 102 −22 111 −31 121 −42 132 −42 142 −32 91 −22 82 −11 71 −41 62 −22 51 −31 42 −42 31 −12 22 −21 11 0]
The fitness value of chrom5 after substitution is **0.025767**.

Figure 10. Process of comparison and substitution operator.

The chromosome codes are selected in accordance with the path order of the end effector. The chromosome is defined as a linear sequence, which is composed of codes randomly extracted from matrixes *Grasp* and *Pos*. A placing point will not repeatedly place other apples when an apple already exists. Taking 4 grasping points and 14 placing points as an example, a complete chromosome can be created as Chrome *A*, shown in Figure 8, which is [0 −11 12 −22 21 −31 32 −42 41 −12 52 −21 61 −32 72 −41 81 −11 92 −22 101 −31 112 −42 121 −12 132 −21 141], when apples in the grasping points are continuously supplied.

(2) Fitness function design

Considering that the path optimization aims to find the paths with the shortest total running time of the manipulator to fully fill one apple tray, named as one chromosome sequence path, the total running time to complete one chromosome sequence path is regarded as the objective function for path optimization, as shown in Equation (3).

$$Time = \sum_{i=1}^{2n+1} t(i), \tag{3}$$

where *t*(*i*) denotes the time for the end effector to run from a grasping point (or placing point) to the next placing point (or grasping point), and *n* is the number of placing points.

The fitness value is the basis for evaluating the fitness of a chromosome, which is closely related to the final selection of the optimal chromosome. Fitness judgment results play a crucial role in the iterative trend of the algorithm. The value of the objective function is the running time of completing one chromosome sequence path. The chromosome is optimal when the time of chromosome sequence path is the minimum. Therefore, the fitness function of the genetic algorithm can be constructed using the objective function to calculate the fitness value of the chromosome in the current population, as shown in Equation (4).

$$f(x) = 1/Time(x), \tag{4}$$

where *Time*(*x*) is the total time required to complete the *x*th chromosome sequence path in the current population.

(3) Region crossover operation

The current population needs to run the crossover operation to achieve comprehensive population development. The crossover probability is P_c. The region crossover operator is described as follows:

a. The crossover regions are determined. Two positive integers named *acr_node* and *acr_len* are first randomly generated. *acr_node* is the starting code of the crossover region, and the value range is [2, 2*n* + 1]. *acr_len* is the length of the crossover step, and the value range is [0, 2*n* − 1]. The sum of *acr_node* and *acr_len* equals 2*n* + 1 when the sum of *acr_node* and *acr_len* is greater than 2*n* + 1.

b. Codes in crossover regions are interchanged. When two paternal chromosomes that need to be changed are named *A* and *B*, respectively, they are interchanged from the *acr_node*$^{\text{th}}$ code to the (*acr_node* + *acr_len*)$^{\text{th}}$ code, and the progeny chromosomes are *A*1 and *B*1.

c. Repeated codes in interchanged chromosomes are recognized and marked. The unchanged codes in *A*1 and *B*1 are compared with those in the changed regions. When the positive numbers in the unchanged regions have the same or adjacent numbers as the positive numbers of the changed regions, then these numbers in the unchanged regions are marked and replaced with the value of 200. Thus, chromosomes *A*1 and *B*1 are replaced by chromosomes *A*2 and *B*2.

d. Repeated codes in interchanged chromosomes are replaced and updated. The current chromosomes *A*2 and *B*2 are scanned code by code to mark the positive number codes that are not equal to 200. These marked positive numbers are mapped to placing points, indicating that apples are already placed in these marked placing points. Then, these marked placing points are removed from the initial placing point matrix *Pos*, and these unmarked placing points form a new matrix *Pos_New*. Subsequently, every number of 200 in the chromosome is randomly replaced with the marked number of any rotation angle solution for any unmarked placing points. The same placing points cannot be repeatedly selected. The progeny chromosomes are updated to be *A*3 and *B*3. The process of the crossover operator is shown in Figure 8, where the values of *acr_node* and *acr_len* are 6 and 12, respectively.

This region crossover operator preserves the excellent genes of the paternal chromosomes and guarantees the randomness of the crossover results. Therefore, the population can comprehensively develop by maintaining excellent genes. The region crossover operator is valuable in selecting the most efficient solution globally among various posture solutions of position points.

(4) Mutation operation

The mutation operation enables the population to develop in a diverse manner, avoids premature convergence, and effectively evades the problem where population development falls into a local optimum. A single-point mutation method is adopted in this study.

The mutation operation is shown as follows. The mutation probability is P_m. A positive integer value named *mut_node* at the interval [2, 2n + 1] is randomly generated, and the chromosomes are searched. When the value of *mut_node* is odd, then the real number code on this odd bit of the chromosome is judged. When the real number code is positive odd, a value of 1 is added. Otherwise, when the code is positive even, a value of 1 is subtracted. Similarly, when the value of *mut_node* is even, the real number code on this even bit of the chromosome is judged. When the real number code on this even bit is negative odd, a value of 1 is subtracted. Otherwise, when the code is negative even, a value of 1 is added.

The mutation operation is performed on chromosome *A*3 in Figure 8. The corresponding real number in chromosome *A*3 is the positive real number 21 when the randomly generated value of *mut_node* is 5. Thus, this real number adds 1 and equals 22. This finding indicates that the manipulator changes from one posture to another when the end effector is placed at the point marked as 2. The process of the mutation operator is shown in Figure 9.

This mutation operation guarantees the randomness of various posture solutions of the manipulator, reflects the individual differences in the offspring population, and avoids the possibility of early local convergence.

(5) Comparison and substitution operation

The comparison and substitution operation of population chromosomes is conducted after the crossover and mutation operations of every generation of population to realize the rapid convergence and stability of the genetic algorithm.

The fitness value of every chromosome in the current population is calculated in accordance with the fitness function. The chromosome with the maximum fitness value in the current population is labeled as the optimal chromosome, where its fitness value is named f_{max}. The optimal chromosome fitness value f_{max} in the current population is compared with the optimal chromosome fitness value f_{oldmax} in all previous populations. The optimal chromosome in the current population is updated to be the new optimal chromosome when f_{max} is larger than f_{oldmax}. The optimal chromosome of all previous populations is retained as the new optimal chromosome of the current population when f_{oldmax} is larger than f_{max}. Then, the new optimal chromosome of the current population is determined and the fitness value of the new optimal chromosome is f_{newmax}.

The new optimal chromosome of the current population is used to replace the chromosome where its fitness values are less than f_{lim} in the current population, as shown in Equation (5).

$$f_{lim} = (f_{newmax} - f_{min})\cdot c + f_{min}, \tag{5}$$

where f_{min} is the fitness value of the worst chromosome in the current population, c is the substitution coefficient, and the value range is (0, 1).

The population of chromosomes evolves. The process of the comparison and substitution operator is shown in Figure 10 when the population size is 5 and the substitution coefficient c is 0.4.

(6) Termination condition setting

After the crossover, mutation, comparison, and substitution operations, the new population gradually generates a convergent progeny population through repeated cycles. The algorithm terminates and outputs the optimal path result and corresponding working time when the iteration times reach the set value.

Optimal Parameter Selection Algorithm

Genetic algorithms consistently use empirical parameters. However, the use of empirical parameters in genetic algorithms is not necessarily reasonable for this study, so this study proposed an optimal parameter selection algorithm to select the optimal values of the four parameters. Four parameters are used in the genetic algorithm, and they are population size, crossover probability, mutation probability, and comparison substitution coefficient. We propose an optimal parameter selection algorithm to select the optimal values of the four parameters and to optimize the motion path of the manipulator. Then, the four selected optimal parameters are inputted into the genetic algorithm. The optimal path and time are obtained, as shown in Figure 7a.

The main steps of the optimal parameter selection algorithm are as follows.

(1) Determination of the ranges of the four parameters. The range of population size is [10, 100]. The ranges of crossover probability, mutation probability, and comparison substitution coefficient are (0, 1), and the values are rounded to two decimal places. A total of 365 groups of nonrepeating valid parameter values within their ranges are obtained.

(2) Acquisition of the target path and program running times. The target path time denotes the packing time taken to fill up an apple tray corresponding to the optimal paths of the mechanic arm acquired by the genetic algorithm, and the program running time is the time taken to run the genetic algorithm. A total of 365 valid parameter values are inputted into the improved genetic algorithm, and the corresponding target path and program running times are obtained, as shown in Table 1.

(3) Fitting between the two types of time and the corresponding four parameters. The four parameters were regarded as inputs, and the target path and program running times were regarded as outputs. SPSS was used for data analysis to establish a linear regression model between the inputs and outputs. The standardized residuals of the target path and program running times were obtained on the basis of the data in Table 1, as shown in Figure 11.

The results in Figure 11a,b show that the final cumulative probability is distributed along the diagonal, indicating that the regression model satisfies the normality hypothesis.

The fitting equations are expressed as Equation (6).

$$\begin{cases} T_1 = -0.349 + 0.013N + 1.417P_c - 0.002P_m - 0.163c \\ T_2 = 27.15 - 0.0309N - 1.955P_c - 2.122P_m + 5.368c \\ T = T_1 + T_2 \end{cases}, \tag{6}$$

where T_1 and T_2 represent the program running and target path times, respectively, T is the sum of the target path and program running times, N is the population size, P_c is the crossover probability, P_m is the mutation probability, and c is the comparison substitution coefficient. The goodness of fit coefficient R_2 of the target time is 80%, and that of the program running time is 84%.

The sum of the target path and program running times is regarded as the final optimal goal, and the combination of the optimal parameters in the fitting equation is obtained using the exhaustive method. Thus, the population size, crossover probability, mutation probability, and comparison substitution coefficient were set to 65, 0.71, 0.96, and 0.16, respectively.

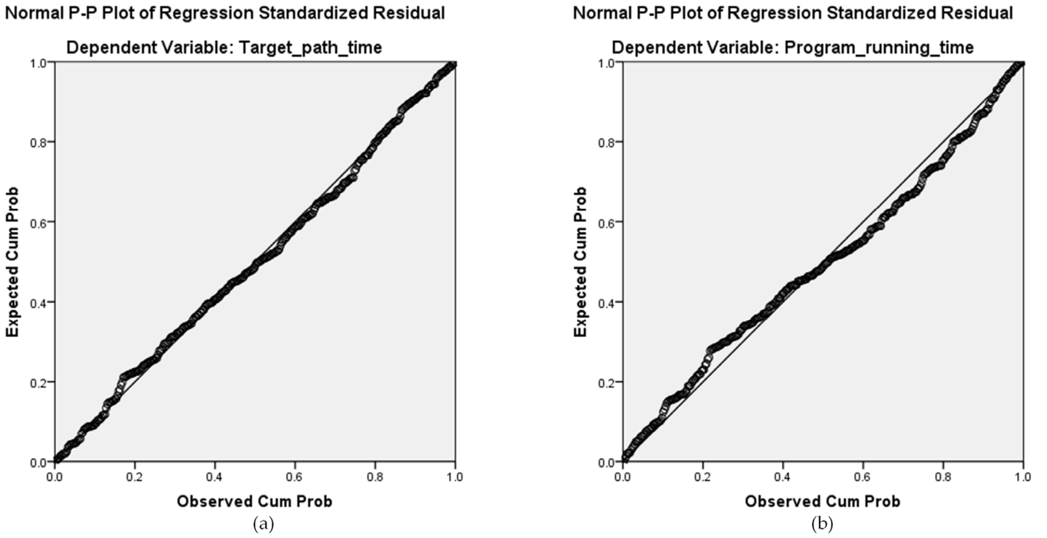


Figure 11. Standardized residuals: (**a**) normalized residuals of target path time; (**b**) normalized residuals of program running time.

Table 1. Parameters and corresponding time.

No.	Population Size	Crossover Probability/%	Mutation Probability/%	Comparison Substitution Coefficient/%	Target Path Time/s	Program Running Time/s
1	33	0.92	0.93	0.13	23.19	1.52
2	59	0.77	0.17	0.52	27.58	1.74
3	13	0.29	0.10	0.69	32.28	0.25
4	20	0.57	0.32	0.75	29.95	0.51
5	77	0.30	0.51	0.33	26.22	0.93
6	13	0.58	0.98	0.51	27.06	0.53
7	76	0.81	0.74	0.36	23.90	2.12
8	54	0.75	0.34	0.22	24.12	1.81
9	22	0.26	0.29	0.58	29.66	0.35
10	16	0.54	0.03	0.53	32.17	0.37
⋮	⋮	⋮	⋮	⋮	⋮	⋮
361	63	0.2	0.7	0.88	28.12	0.60
362	46	0.61	0.73	0.53	27.78	1.08
363	63	0.27	0.39	0.60	28.54	0.76
364	17	0.75	0.13	0.28	28.63	0.55
365	18	0.07	0.84	0.31	28.02	0.14

4. Results and Discussion

4.1. Comparative Study of the Improved Performance

The experiments were conducted from 2019 to 2020. Tests were repeated 10 times using four genetic algorithms, namely, the traditional genetic algorithm with empirical parameters, the improved genetic algorithm in Section 3.1 with empirical parameters, the traditional genetic algorithm with the optimal parameter selection algorithm in Section 3.2, and the improved genetic algorithm fused with the optimal parameter selection algorithm, to verify the improved performance of the genetic algorithm based on the optimal parameter selection algorithm. In these tests, four grasping points and fourteen placing points were found, and apples were continuously supplied at every grasping point. The angular velocities of the two rotatory joints of the manipulator were 40°/s and 50°/s, respectively, and the final iteration times were set to 100.

Traditional genetic algorithms frequently select good or poor chromosomes on the basis of roulette operators. Crossover and mutation probabilities are constantly fixed in crossover and mutation operations. On the basis of the empirical parameters of the traditional genetic algorithm, the population size was set to 50, the crossover probability was set to 0.8, and the mutation probability was set to 0.2. The average time of target paths was 27.13 s, the average running time of the program was 1.38 s, and the total time was 28.51 s.

The improved genetic algorithm using empirical parameter values was tested to compare the performance of the improved genetic algorithm mentioned in Section 3.1 with that of the traditional genetic algorithm. The average time of target paths was 24.84 s, the average running time of the program was 1.61 s, and the total time was 26.45 s.

Similarly, we tested the performance of the traditional genetic algorithm fused with the optimal parameter selection algorithm to verify the improvement effect of the optimal parameter selection algorithm. The average time of target paths was 25.04 s, the average running time of the program was 1.80 s, and the total time was 26.84 s.

The iterative results of the improved genetic algorithm fused with the proposed optimal parameter selection algorithm were as follows. The optimal parameter selection algorithm gradually and completely converged when the number of iterations reached 70. The average time of the optimal target paths was 22.34 s, the average running time of the program was 1.52 s, and the average total time was 23.86 s.

4.2. Comparative Study of the Optimal Performance under Different Numbers of Grasping and Placing Points

Tests under different numbers of randomly arranged grasping points and different types of apple trays with different specifications were conducted to test the optimal performance of the improved genetic algorithm fused with the optimal parameter selection algorithm under different working conditions. The results are shown in Table 2. The total time of the improved genetic algorithm fused with the optimal parameter selection algorithm was less than that of the traditional genetic algorithm. The average optimization amplitudes were 14.63%, 15.42%, 16.24%, and 13.82% for 9-groove, 12-groove, 14-groove, and 16-groove trays, respectively.

As shown in Table 2, the optimization amplitude is larger when the number of grasping points is greater than four than that when the number of grasping points is less than four. Therefore, the proposed algorithm is effective when the number of grasping points is greater than four.

The fluctuation in optimization amplitude is larger when the number of apple placing points is less than 12 than that when the number is larger than 12. The fluctuation in optimization amplitude is small when the number of apple placing points is greater than 14. The optimization amplitude is larger when the number of apple grasping points is constant and the number of apple placing points (i.e., the number of grooves of apple trays in different specifications) is 12 or 14.

Considering the specific values of the optimization amplitude, the optimization amplitude reaches its minimum when the number of grasping points is two and the number of placing points is twelve. Under the condition where the number of placing points is constant, the value of optimization amplitude is basically the minimum when the number of grasping points is two. This finding is because the number of grasping points is extremely small, optional packing paths are few, and the optimal algorithms have few effects. On the contrary, excessive optional packing paths are found, and the optimized paths obtained by the two algorithms are locally optimal when excessive grasping points are found. The situation is mostly the same when the number of placing points varies and the number of grasping points is constant. The optimization performance of the improved algorithm is optimal. The optimal optimization amplitude is 20.81% when the number of grasping points is 4 and the number of placing points is 12.

Table 2. Results under different specifications of trays and different numbers of grasping points.

S	9-Groove			12-Groove			14-Groove			16-Groove		
No.	T-F /s	T-T /s	O-A /%	T-F /s	T-T /s	O-A /%	T-F /s	T-T /s	O-A /%	T-F /s	T-T /s	O-A /%
2	13.23	14.59	9.32	18.54	20.29	8.62	22.90	26.48	13.52	26.96	31.44	14.25
3	13.77	15.84	13.07	19.08	22.30	14.44	22.32	26.96	17.21	28.18	32.46	13.19
4	13.60	16.77	18.90	18.84	23.79	20.81	23.86	28.51	16.31	28.91	34.37	15.89
5	13.22	16.07	17.73	19.32	23.23	16.83	23.52	28.15	16.45	28.22	31.60	10.70
6	14.02	16.33	14.15	19.71	23.58	16.41	24.02	29.19	17.71	28.54	33.62	15.11
ave.	13.57	15.92	14.63	19.10	22.64	15.42	23.32	27.86	16.24	28.16	32.70	13.82

Note: "S" denotes the "Specifications of apple trays", "No." denotes the "Grasping point number", "T-F" indicates the "Time of the improved genetic algorithm fused with optimal parameter selection algorithm", "T-T" indicates the "Time of the traditional genetic algorithm", and "O-A" indicates the "Optimization amplitude".

The test results show that the optimization amplitude is 10.70%, which is smaller than most values when the numbers of grasping and placing points are 5 and 16, as shown in Table 2. This finding is possibly because the coordinates of the grasping and placing points only fit the principle of proximity when the traditional genetic algorithm is applied.

The improved genetic algorithm fused with the optimal parameter selection algorithm is effective when the number of apple grasping points is greater than 4 and the number of placing points is 12 or 14.

The packing path optimization algorithm of the mechanical arm of the apple packing robot designed in this study has the following advantages:

(1) A two-level coding genetic algorithm can solve the packing path optimization problem of a two-posture manipulator.

(2) The traditional genetic algorithm was improved and a region crossover operator was designed. In contrast to traditional single-point crossover, multi-point crossover, dislocation crossover, and random crossover, the crossover operator is based on the region set by two chromosomes to cross each other, and further modifies the crossover result to eliminate the repeated coding. The crossover operator has the advantage of retaining excellent gene fragments and making the optimization process more stable.

(3) Aiming at the problem that the traditional genetic algorithm is not well targeted based on empirical parameters, an optimal parameter-fitting exhaustive algorithm was designed. The algorithm obtained the corresponding packing time after uniformly sampling each parameter value, and then established the multiple linear regression model between the packing time and each parameter. The parameter values corresponding to the minimum packing time were obtained by exhaustive enumerations, as the optimal parameters of the genetic algorithm and the improved genetic algorithm were combined to realize the optimal path planning of the manipulator. It has the advantage of high time-optimization range, and overcomes the disadvantage of the traditional genetic algorithm, namely, that it can easily fall into a local optimum.

This study is of one of the main works of apple packaging robots. Further studies will focus on developing and optimizing the apple packing robots.

5. Conclusions

To solve the problem of the path optimization of manipulators moving from multiple fixed points to multiple varying points and with dual postures, an improved genetic algorithm based on two-level chromosome coding and optimal parameter selection was proposed. Based on the test results and analysis, the following conclusions can be drawn:

(1) The improved genetic algorithm fused with the optimal parameter selection algorithm can effectively avoid the premature convergence of the traditional genetic algorithm.
(2) The improved genetic algorithm fused with the optimal parameter selection algorithm provides an effective automatic packing method in the apple packing industry by reducing the packing time and improving the efficiency.
(3) The two-level chromosome coding method used in the improved genetic algorithm can solve the packing path optimization problem of the two-posture manipulator.
(4) The optimal parameter-fitting exhaustive algorithm can automatically determine the optimal parameter value of the proved genetic algorithm.

Moreover, the design of the gripper device, and the calculation and analysis of the gripping force, will be carried out in further work. The mechanism of the influence of the size and weight of apples on the motion of the manipulator also needs further study.

6. Patents

An optimization method for trajectory planning of a robotic arm based on genetic algorithm, China National invention patent, Patent No. 2019102649512.

An optimization method of apple automatic packing path based on optimal parameter genetic algorithm, China National invention patent, Patent No. 2019111718121.

Author Contributions: Conceptualization, R.X.; funding acquisition, R.X.; methodology, R.X. and B.F.; supervision, R.X.; validation, R.X. and B.F. writing, R.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Zhejiang Provincial Natural Science Foundation of China, "grant number LY17C130006", the National Natural Science Foundation of China, "grant number 61105024".

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Yu, J.H.; Wang, J.S.; Lu, H.; Ma, D.J.; Li, X.F. Research on the Manipulator for Packing Apple Based on its Material Function. *Appl. Mech. Mater.* **2013**, *437*, 517–521. [CrossRef]
2. Ling, X.; Zhao, Y.S.; Gong, L.; Liu, C.L.; Wang, T. Dual-arm Cooperation and Implementing for Robotic Harvesting Tomato Using Binocular Vision. *Robot. Auton. Syst.* **2019**, *114*, 134–143. [CrossRef]
3. Levin, M.; Degani, A. A Conceptual Framework and Optimization for a Task-Based Modular Harvesting Manipulator. *Comput. Electron. Agric.* **2019**, *166*, 104987. [CrossRef]
4. Hayashi, S.; Shigematsu, K.; Yamamoto, S.; Kobayashi, K.; Kohno, Y.; Kamata, J.; Kurita, M. Evaluation of a Strawberry-harvesting Robot in a Field Test. *Biosyst. Eng.* **2010**, *105*, 160–171. [CrossRef]
5. Yu, X.J.; Fan, Z.M.; Wang, X.D.; Wan, H.; Wang, P.B.; Zeng, X.L.; Jia, F. A Lab-Customized Autonomous Humanoid Apple Harvesting Robot. *Comput. Electron. Eng.* **2021**, *96*, 107459. [CrossRef]
6. Zhao, D.A.; Lv, J.D.; Ji, W.; Zhang, Y.; Chen, Y. Design and Control of an Apple Harvesting Robot. *Biosyst. Eng.* **2011**, *110*, 112–122.
7. Arad, B.; Balendonck, J.; Barth, R.; Ben-Shahar, O.; Edan, Y.; Hellstrom, T.; Kurtser, P.; Ringdahl, O.; Tielen, T.; van Tuijl, B.; et al. Development of a Sweet Pepper Harvesting Robot. *J. Field Robot.* **2020**, *37*, 1027–1039. [CrossRef]
8. Zhang, K.X.; Lammers, K.; Chu, P.Y.; Li, Z.J.; Lu, R.F. System Design and Control of an Apple Harvesting Robot. *Mechatronics* **2021**, *79*, 102644. [CrossRef]
9. Bu, L.; Chen, C.; Hu, G.; Sugirbay, A.; Sun, H.; Chen, J. Design and Evaluation of a Robotic Apple Harvester Using Optimized Picking Patterns. *Comput. Electron. Agric.* **2022**, *198*, 107092. [CrossRef]
10. Jun, J.; Kim, J.; Seol, J.; Kim, J.; Son, H.I. Towards an Efficient Tomato Harvesting Robot: 3D Perception, Manipulation, and End-Effector. *IEEE Access* **2021**, *9*, 17631–17640. [CrossRef]

11. Bulanon, D.M.; Kataoka, T. Fruit Detection System and an End Effector for Robotic Harvesting of Fuji Apples. *Agric. Eng. Int. CIGR J.* **2010**, *12*, 203–210.
12. Xiong, Y.; Peng, C.; Grimstad, L.; From, P.J.; Isler, V. Development and Field Evaluation of a Strawberry Harvesting Robot with a Cable-driven Gripper. *Comput. Electron. Agric.* **2019**, *157*, 392–402. [CrossRef]
13. Tang, Z.L.; Xu, L.J.; Wang, Y.C.; Kang, Z.L.; Xie, H. Collision-Free Motion Planning of a Six-Link Manipulator Used in a Citrus Picking Robot. *Appl. Sci.* **2021**, *11*, 11336. [CrossRef]
14. Kukker, A.; Sharma, R. Stochastic Genetic Algorithm-Assisted Fuzzy Q-Learning for Robotic Manipulators. *Arab. J. Sci. Eng.* **2021**, *46*, 9527–9539. [CrossRef]
15. Barakat, A.N.; Gouda, K.A.; Bozed, K.A. Kinematics analysis and simulation of a robotic arm using MATLAB. In Proceedings of the International Conference on Control Engineering & Information Technology, Changsha, China, 23 July 2017.
16. Feng, Q.C.; Zou, W.; Fan, P.F.; Zhang, C.F.; Wang, X. Design and Test of Robotic Harvesting System for Cherry Tomato. *Int. J. Agric. Biol. Eng.* **2018**, *11*, 96–100. [CrossRef]
17. Salloom, T.; Yu, X.B.; He, W.; Kaynak, O. Adaptive Neural Network Control of Underwater Robotic Manipulators Tuned by a Genetic Algorithm. *J. Intell. Robot. Syst.* **2020**, *97*, 657–672. [CrossRef]
18. Wang, M.M.; Luo, J.J.; Zheng, L.L.; Yuan, J.P.; Walter, U. Generate Optimal Grasping Trajectories to the End-Effector Using an Improved Genetic Algorithm. *Adv. Space Res.* **2020**, *66*, 1803–1817. [CrossRef]
19. Qadir, Z.; Ullah, F.; Munawar, H.S.; Al-Turjman, F. Addressing disasters in smart cities through UAVs path planning and 5G communications: A systematic review. *Comput. Commun.* **2021**, *168*, 114–135. [CrossRef]
20. Jovanovic, A.; Uzelac, A.; Kukic, K.; Teodorovic, D. The shortest-path and bee colony optimization algorithms for traffic control at single intersection with NetworkX application. *Demonstr. Math.* **2024**, *57*. [CrossRef]
21. Nazarahari, M.; Khanmirza, E.; Doostie, S. Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm. *Expert Syst. Appl.* **2019**, *115*, 106–120. [CrossRef]
22. Zhang, Y.; Zhang, R.; Li, H.F. Mixed-integer trajectory optimization with no-fly zone constraints for a hypersonic vehicle. *Acta Astronaut.* **2023**, *207*, 331–339. [CrossRef]
23. Li, H.H.; Lin, Z.G.; Du, J. Excavator Autonomous Mining Segmentation Variable Order Polynomial Trajectory Planning. *Trans. Chin. Soc. Agric. Mach.* **2016**, *47*, 319–325. (In Chinese)
24. Reynosomora, P.; Chen, W.; Tomizuka, M. On the Time-optimal Trajectory Planning and Control of Robotic Manipulators along Predefined Paths. In Proceedings of the American Control Conference, Washington, DC, USA, 17–19 June 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 371–377.
25. Ju, H.H.; Fu, R. Time-Optimal Trajectory Planning Algorithm Based on GA for Manipulator. *Control. Eng. China* **2012**, *3*, 112–117. (In Chinese)
26. Lehman, J.; Stanley, K.O. Novelty search and the problem with objectives. In *Genetic Programming Theory and Practice IX*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 37–56.
27. Naredo, E.; Trujillo, L.; Legrand, P.; Silva, S.; Muñoz, L. Evolving genetic programming classifiers with novelty search. *Inf. Sci.* **2016**, *369*, 347–367. [CrossRef]
28. He, L.Y.; Yang, T.W.; Wu, C.Y.; Yu, Y.X.; Tong, J.H.; Chen, J.C. Optimization of Replugging Tour Planning Based on Greedy Genetic Algorithm. *Trans. Chin. Soc. Agric. Mach.* **2017**, *5*, 41–48. (In Chinese)
29. Lamini, C.; Benhlima, S.; Elbekri, A. Genetic Algorithm Based Approach for Autonomous Mobile Robot Path Planning. *Procedia Comput. Sci.* **2018**, *127*, 180–189. [CrossRef]
30. Liu, J.X.; Cai, Y.B.; Cao, Y. A Robot Path-Planning Method Based on an Improved Genetic Algorithm. *Trans. FAMENA* **2024**, *48*, 141–153. [CrossRef]
31. Reiter, A.; Muller, A.; Gattringer, H. On Higher-Order Inverse Kinematics Methods in Time-Optimal Trajectory Planning for Kinematically Redundant Manipulators. *IEEE Trans. Ind. Inform.* **2018**, *14*, 1681–1690. [CrossRef]
32. Jin, X.; Li, D.Y.; Ma, H.; Ji, J.T.; Zhao, K.X.; Pang, J. Development of Single Row Automatic Transplanting Device for Potted Vegetable Seedlings. *Int. J. Agric. Biol. Eng.* **2018**, *11*, 67–75. [CrossRef]

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.