

## Article

# Research on Multiobjective Optimization Algorithm for Cooperative Harvesting Trajectory Optimization of an Intelligent Multiarm Straw-Rotting Fungus Harvesting Robot

Shuzhen Yang <sup>1,2,\*</sup> , Bocai Jia <sup>2</sup>, Tao Yu <sup>2</sup> and Jin Yuan <sup>3</sup>

<sup>1</sup> School of Intelligent Manufacturing and Control Engineering, Shanghai Polytechnic University, Shanghai 201209, China

<sup>2</sup> School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200444, China; 18800205923@163.com (B.J.); yutao@shu.edu.cn (T.Y.)

<sup>3</sup> College of Mechanical and Electronic Engineering, Shandong Agricultural University, Tai'an 271018, China; jyuan@sdau.edu.cn

\* Correspondence: szyang@sspu.edu.cn; Tel.: +86-21-5021-6899

**Abstract:** In view of the difficulties of fruit cluster identification, the specific harvesting sequence constraints of aggregated fruits, and the balanced harvesting task assignment for the multiple arms with a series-increasing symmetric shared (SISS) region, this paper proposes a multi-objective optimization algorithm, which combines genetic algorithm (GA) and ant colony optimization (ACO) stepwise, to optimize the multiarm cooperative harvesting trajectory of straw-rotting fungus to effectively improve the harvesting efficiency and the success rate of non-destructive harvesting. In this approach, firstly, the multiarm trajectory optimization problem is abstracted as a multiple travelling salesman problem (MTSP). Secondly, an improved local density clustering algorithm is designed to identify the cluster fruits to prepare data for harvesting aggregated fruits in a specific order later. Thirdly, the MTSP has been decomposed into M independent TSP (traveling salesman problem) problems by using GA, in which a new DNA (deoxyribonucleic acid) assignment rule is designed to resolve the problem of the average distribution of multiarm harvesting tasks with the SISS region. Then, the improved ant colony algorithm, combined with the auction mechanism, is adopted to achieve the shortest trajectory of each arm, which settles the difficulty that the clustered mature fruits should be harvested in a specified order. The experiments show that it can search for a relatively stable optimal solution in a relatively short time. The average harvesting efficiency is up to 1183 pcs/h and the average harvesting success rate is about 97%. Therefore, the proposed algorithm can better plan the harvesting trajectory for multiarm intelligent harvesting, especially for areas with many aggregated fruits.



**Citation:** Yang, S.; Jia, B.; Yu, T.; Yuan, J. Research on Multiobjective Optimization Algorithm for Cooperative Harvesting Trajectory Optimization of an Intelligent Multiarm Straw-Rotting Fungus Harvesting Robot. *Agriculture* **2022**, *12*, 986. <https://doi.org/10.3390/agriculture12070986>

Academic Editor: Francesco Marinello

Received: 30 May 2022

Accepted: 6 July 2022

Published: 8 July 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 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/>).

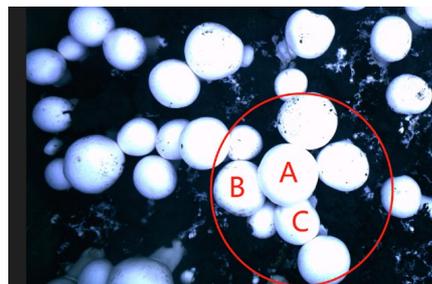
**Keywords:** straw-rotting fungus; multiarm harvesting trajectory optimization; multiobjective optimization; cluster fruit; genetic ant colony stepwise algorithm

## 1. Introduction

Straw rotting fungi are fungi that absorb the decayed humus of grass straws (such as straw and wheat straw) as the main source of nutrition [1], mainly including *Agaricus bisporus*, *Agaricus blazei*, straw mushrooms, capsule mushrooms, etc. Its factory production usually adopts the bed planting method. The growth of its fruit is similar to a round or spherical shape, easy to cluster, the combination of fruit and culture medium (soil) is relatively rigid and compact, the ripening time of each fruit is different, and the fruit is tender and vulnerable. This kind of fruit usually needs to be harvested by selective harvesting, and the aggregated mature fruits need to be harvested in the specified order; otherwise, the fruit is easy to be damaged. Therefore, it is difficult to achieve effective

automatic nondestructive harvesting [2]. Under the trend of serious labor shortages, although the cultivation of such crops can achieve factory-like and intensive production, the labor-intensive harvesting process [3] still relies on manual labor, which has become the main bottleneck affecting further production and efficiency improvement. Therefore, there is an urgent need for harvesting robots that can adapt to the factory and intensive cultivation environment for intelligent and efficient harvesting.

In the 1990s, Reed, J. N. and Tillett, R.D. [4] proposed the first selective harvesting robot for *Agaricus bisporus*, which made selective autonomous harvesting of straw-rotting fungus feasible. Since then, scholars have carried out many researches to on the edible fungus harvesting robot technology but mainly focused on visual recognition and end-effectors to improve the harvesting success rate [5–13], and few studies on trajectory planning. Yang [14] aimed at the issue of mushroom harvesting path planning. GA was used to optimize the harvesting path to improve the harvesting efficiency after obtaining the location coordinates of all the mushrooms that can be harvested. In order to improve the harvesting efficiency of the mushroom harvesting robot, Hu et al. [15] proposed an improved simulated annealing algorithm to find the optimal path, which can increase the harvesting efficiency by 14–18%. However, both of the above studies are all optimized for the path of the single-arm harvesting robot, which can improve the harvesting efficiency to a certain extent, but far from manual efficiency. In addition, these methods do not consider the specific harvesting order of the aggregated fruits and are merely suitable for harvesting the fruits, which are relatively sparse. When the fruits grow densely, take *Agaricus bisporus* as an example, as shown in Figure 1, for the clustered fruits in the red circled area, the height of fruit body A is higher than B and C. If B or C is harvested before A, B and C will be damaged or even be harvested unsuccessfully because their cap is covered by A. Moreover, A may be pushed down, which will cause its cup center to deviate greatly, resulting in failure when picking A. So, the aggregated fruits should be harvested in order of height. Otherwise, the success rate of non-destructive harvesting and the harvesting quality will be reduced. So, it is significant to greatly improve harvesting efficiency; meanwhile, to take into account the harvesting order of aggregated fruits to improve the success rate of non-destructive harvesting further.



**Figure 1.** Schematic diagram of aggregated fruits.

Because the fruit is delicate and vulnerable and the harvesting environment is complex, the harvesting operation is usually limited to slow and time-consuming [16]. So, the harvesting efficiency is normally much lower than manual efficiency, which leads to the situation that the harvesting robot cannot be widely used in actual production. To solve this issue, using multiple harvesting arms is a typical approach.

A cotton harvesting robot with multiple robotic arms has been developed to achieve multiple plucking of crops, which increases the yield by 20–25% [17].

Zion [18] developed a melon harvesting robot with multiple Cartesian arms to accelerate the speed. The robot travels along a two-dimensional field at a constant velocity. The multiarm assignment is modeled as a  $k$ -colorable sub-graph problem and uses a greedy algorithm to achieve an optimal solution. Because the greedy algorithm focuses on local optimization, the effect of global optimization may not be very good.

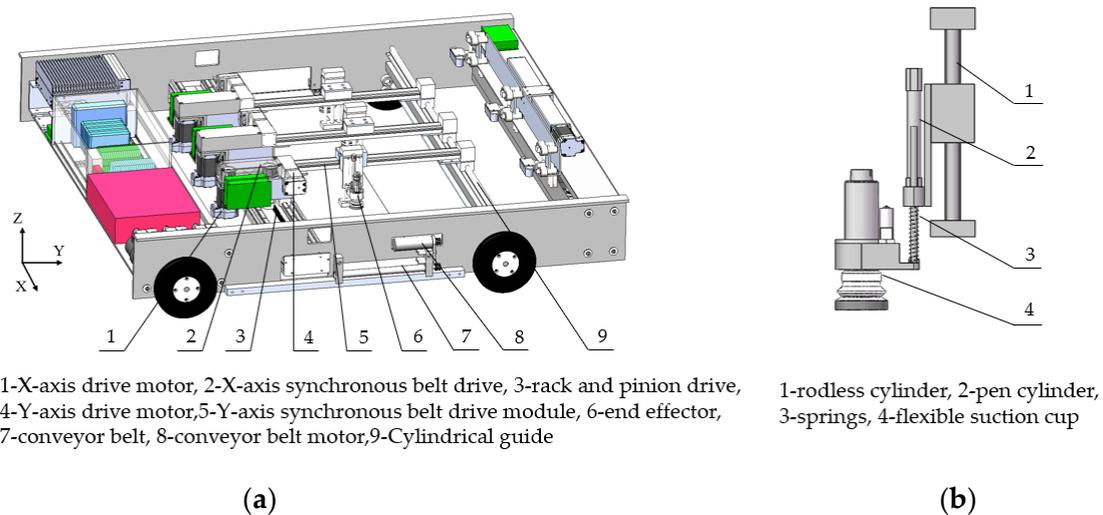
A strawberry harvesting robot with dual cartesian arms was developed to reduce the cost and optimize the harvesting efficiency. The fruits were partitioned into several subsections equally for each arm to harvest. The harvesting speed can be reduced to 4.6 s [19].

An oyster mushroom harvesting robot with four harvesting arms connected in parallel on a common mobile platform is provided to improve efficiency [20]. Each arm is allocated to harvest in the divided area independently, which is divided from left to right with an approximately equal number of target fruits, and the mature fruit closest to the end actuator is picked first in each area.

A multiple robot arm system for kiwifruit harvest has been designed to reduce the harvesting cycle time, thereby increasing the efficiency to meet the requirements of commercial applications [21]. The multiarm harvesting robot sorts and partitions target fruits according to their x coordinates and assigns them to the harvesting arms so that the amount of the harvested fruit is approximately equal for each arm. However, for fruits growing in clusters, the clustered fruits are completely allocated to the same arm to be harvested from low to high to avoid touching or moving other fruits' positions. Although this method can improve the picking success rate of fruit clusters, it will affect the distribution uniformity of each arm, thereby reducing the picking efficiency.

Most of the above multiarm harvesting trajectory planning uses the traditional method, which partitions target fruits approximately evenly according to the arrangement direction of the harvesting arms and allocates them to the robot arms, and each arm harvests fruits from left to right or from right to left. Normally, this method can achieve a good result. However, the trajectory planning effect of this method will be greatly reduced when the fruits are distributed seriously ununiformly and even with many fruit clusters. Moreover, this method assumes that each fruit can be picked by each arm; that is, the accessibility of the task to be executed to the individual is consistent. While the fact is that not every fruit can be picked by each arm due to the size of the arm, which indicates that the accessibility of the task to be executed in the instance is inconsistent. This shortcoming will also increase the difficulty of equal allocation among each arm. Therefore, a more global and flexible optimization approach is required to resolve the above issues. In order to effectively improve harvesting efficiency and adapt to the environment of straw-rotting fungus factory bed planting, the harvesting robot can be designed as a highly cost-effective Cartesian coordinate harvesting robot as shown in Figure 2, equipped with multiple harvesting arms [22]. Here, because the mature fruits of straw-rotting fungus are little difference in height, it can be assumed that the time spent in the height direction (Z-axis) of each fruit during harvesting is the same, so that the dimension of the harvesting trajectory planning of such fruits can be reduced as a two-dimensional trajectory planning problem in the XY plane. In addition, since the shape of the straw-rotting fungus is basically spherical, the projection of the shape of the fruit on the XY plane is further approximated as a circle. Just as the harvesting sequence planning problem of a single-arm harvesting robot can be regarded as a TSP problem [23]. Given the above assumptions, the multiarm cooperative harvesting trajectory planning problem studied in this paper can be simplified as a typical MTSP as well.

Similar to most fruit and vegetable picking robots mentioned above, the harvesting time for a straw-rotting fungus fruit is also time-consuming (about 5 s) due to the operation of grasping the fruit and detaching it from the culture medium or soil cannot be too fast to avoid damaging the fruit. Compared with the harvesting time (time required for harvesting at the target fruit position), the moving time (time used to move from the picked fruit to the next target fruit) is much shorter. Therefore, it is much more significant to allocate the harvesting tasks to each arm as uniformly as possible before optimizing the shortest path for each arm [24–26]. In the meantime, it is also necessary to take into account the harvesting sequence of aggregated fruits to improve the success rate of non-destructive harvesting.



**Figure 2.** Detailed mechanical design of MFA. (a) Detailed mechanical structure of MFA; (b) detailed mechanical structure of end-effector.

For the sake of the uniform assignment of tasks for each harvesting arm, in addition to the region segmentation method described in the previous literature analysis, the stepwise algorithm used for MTSP is more appropriate to solve the issue, which is also superior to the general heuristic algorithm [27–30]. A stepwise algorithm is proposed to resolve the MTSP of multi-UAV cooperative airport bird repelling, which adopts a genetic algorithm to divide the MTSP into  $M$  independent TSP. This paper shows that GA is suitable to solve the task balance assignment of MTSP [31]. Lu et al. [32] combined the K-means clustering algorithm with GA to solve multiobjective MTSP, although a better-balanced task assignment is obtained, but it still has poor trajectory searchability.

The ant colony algorithm is widely used in combinatorial optimization problems due to its strong search ability and fast convergence speed [33]. Necula et al. [34] used ACO as the bi-standard surface for solving the multitraveling salesman problem. Changdar et al. [35] adopted ACO to resolve the multi-stop multitraveling salesman problem with non-random parameters. Although both of them achieve a good result, it is easier to fall into the local optimal solution prematurely in MTSP with high complexity problems.

In addition to solving the problem of equal allocation of multiarm tasks under SSIS restrictions to improve efficiency, it is also necessary to take into account the harvesting order of aggregated fruits to improve the success rate of non-destructive harvesting mentioned above. To achieve this aim, the above algorithm for solving similar MTSP should be further improved by combining a multiobjective optimization method. More than this, another challenge is how to improve the algorithm so that the fruit clusters that need to be picked in a specific order can be split and allocated to multiple different arms, to overcome the shortcoming of the whole fruits in the same cluster being merely allocated to the same arm, so as to further increase the efficiency even though the mature fruits are seriously unevenly distributed on a culture medium (soil) and with many fruit clusters.

Through the above analysis, this paper proposes an improved genetic ant colony multiobjective optimization algorithm, which makes comprehensive use of the advantages of both the genetic algorithm and the ant colony algorithm to resolve the difficulties of multiarm cooperative harvesting of straw-rotting fungus and achieve both high harvesting efficiency and a high success rate. The main contributions of this approach are as follows:

- (1) The trajectory planning problem of multiarm cooperative harvesting of straw-rotting fungus is transformed into an MTSP problem;
- (2) To resolve the difficulty of accurately recognizing the fruit clusters of straw-rotting fungus due to their different shapes and uncertain density, a density-based clustering

- algorithm is improved by designing a new method for calculating local density, which can better meet the clustering analysis of straw-rotting fungus;
- (3) A multiobjective optimization model is built for the trajectory optimization of an intelligent multiarm straw rotting fungus harvesting robot;
  - (4) The improved ant colony algorithm combined with the auction mechanism is used to achieve the shortest trajectory of each TSP problem. Meanwhile, the fruit clusters that are required to be picked in a specific order can be allocated to different arms instead of being allocated to a single arm by combining the auction mechanism with an ant colony.

## 2. Description of Multiarm Cooperative Harvesting Trajectory Planning Problem

### 2.1. Intelligent Multiarm Straw-Rotting Fungus Harvesting Robot

As shown in Figure 2, an intelligent multiarm straw rotting fungus harvesting robot consists of the following four main parts: (i) a mobile platform (MP) with a conveyor belt; (ii) a visual position system (VPS), and (iii) multiple flexible arms (MFA); (iv) control system (CS). The MP is placed on the rails of the multistory shelves to move along the rail with the help of the MP, the VPS can recognize and locate all mature fruits. The CS can plan the harvesting trajectory and control MFA to harvest the mature fruits and send them to the convey for blanking.

MFA consists of at least two Cartesian arms with three DOF. As shown in Figure 2a, the X-axis adopts the gear rack motor traverse multi-axis mechanism, and each is mounted on a guide rail. The X-axis motor drives the gears in the gear rack through a synchronous belt drive structure to move the end-effector along the X-direction. The Y-axis uses the synchronous belt drive module to move the end-effector in the Y direction. The end-effector is designed into a two-stage driving structure, as shown in Figure 2b, to adapt to the narrow layer height of the edible fungus culture rack (available design height is only about 250 mm while travel should be 160 mm). The first stage is driven by a rodless cylinder, and the second stage is driven by a pen cylinder. The suction cup is connected to the pen cylinder, which is connected to the rodless cylinder. During harvesting, the rodless cylinder drives the pen cylinder lower. Firstly, the rodless cylinder inflates to drive the pen cylinder down. Then, the pen cylinder deflates to cause the suction cup to drop under the action of gravity until it touches the surface of the target fruit. Finally, vacuum to grasp the target fruit, rotate or wobble, and pull up to detach the fruit from the soil. The structure can better adapt to the large height difference in fruits and realize picking action by wobble or rotation motion.

The working flow of the proposed robot is shown in Figure 3, as follows: First, the camera traverses all fruit images within the current visual area; Secondly, the image is transferred to the host computer, and the positions of the mature fruits are identified by image processing; Thirdly, all of the coordinates of fruits to be picked are scheduled and allocated to multiple harvesting arms; Fourth, the controller drives the arms to harvest fruits after receiving the assigned task; Finally, the robot moves forward as a whole by MP to start the next harvesting cycle until the harvest of one layer is finished. In addition to the use of three arms, in order to improve efficiency, the following are also performed: (i) Multiple economical depth cameras are used to shorten the photographing and identification time; (ii) Visual processing and harvesting operation work in parallel rather than in series. In the current cycle, the multiarm harvester harvests the fruits identified in the previous cycle.

As shown in Figure 4, the control system takes the motion controller (TRIO MC4N) as the core. The motion controller communicates with the PC through the Ethercat bus to obtain the picking task for each arm, then drives the motor of each axis and controls the cylinder, sucker, and other actuators of the end-effector to work monitor the working status of each actuator in real-time and returns it to the upper computer. The economical and compact stepping servo motor is used for each axis motor, and the communication between the motor and the controller is also via the Ethercat bus.

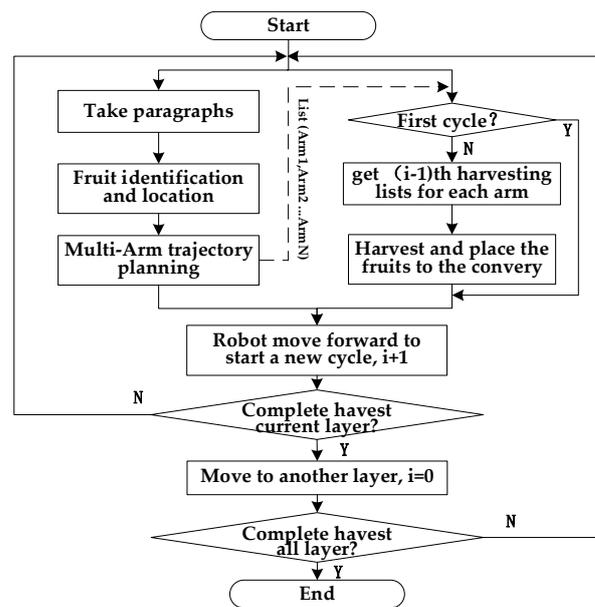


Figure 3. Workflow of the multiarm harvesting robot.

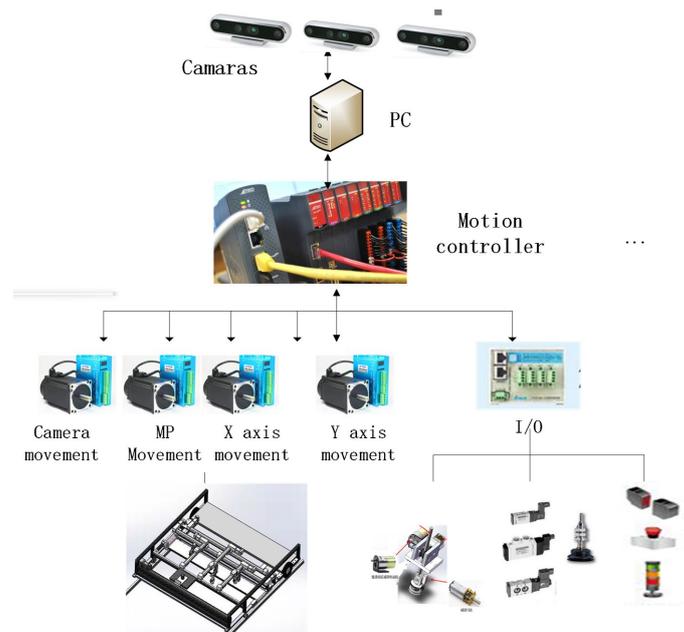


Figure 4. Hardware composition diagram of control system.

### 2.2. Accessibility Analysis of Multiple Harvesting Arms

Generally, the MTSP problem assumes that all cities can be visited by each traveler, but this is not the case with the multiple harvesting arm structure in this paper. As shown in Figure 2a, multiple arms are arranged in a series along the X-direction. Due to the unignorable width of the end-effector, each arm has a certain inaccessible area, and the reachable range of each arm is different.

The accessibility of the picking arm in this paper is characterized by exclusive area, partially shared area, and fully shared area, and they are symmetrically distributed as shown in Figure 5, which is called the serial increasing symmetric shared (SISS) area. The definition of the SISS is described in detail in [36]. Such an SISS area makes it more difficult to assign the harvesting tasks equally among each arm.

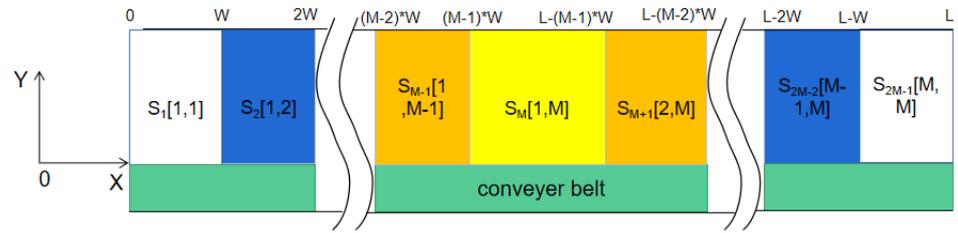


Figure 5. Distribution of accessible area.

The range of each harvesting region can be described as follows:

$$S_k = (X_{k-1}, X_k), k \in \{1, 2, \dots, 2M - 1\} \tag{1}$$

$$X_k = \begin{cases} W \times k, & k \in \{1, 2, \dots, M\} \\ L - (2M - 1 - k) \times W, & k \in \{M + 1, M + 2, \dots, 2M - 1\} \end{cases} \tag{2}$$

where  $L$  is the length of the harvesting region,  $W$  is the width of arm and end-effector,  $M$  is the amount of harvesting arm,  $S_k$  is the  $k$ -th harvesting region, and  $X_k$  is the coordinate of the end point in the  $X$ -direction of the  $k$ -th harvesting area.

Equation (3) shows the accessible harvesting arms per area as follows:

$$S_k(j1, j2) = \begin{cases} S_k(1, k), & k \in \{1, 2, \dots, M\} \\ S_k(k - M + 1, M), & k \in \{M + 1, M + 2, \dots, 2M - 1\} \end{cases} \tag{3}$$

where  $(j1, j2)$  indicate harvesting arms  $j1$  to  $j2$  can reach the  $k$ -th area,  $1 \leq j1 \leq j2 \leq M$ .

### 2.3. Description of Trajectory Planning for Multiarm Cooperative Harvesting

As described in Section 1, the multiarm trajectory planning problem of this paper can be abstracted to an MTSP problem with an SSIS Area. In which case, the aggregated fruits should be harvested in the order from high to low in the  $Z$  direction. The goal of trajectory optimization is to achieve both high harvesting efficiency and a high success rate with the following constraints:

- (1) Each harvesting arm’s accessible region is limited by the SISS region shown in the above section;
- (2) To avoid collision between the adjacent harvesting arms, a safety distance  $Dist$  is required;
- (3) After completing the harvesting tasks, each harvesting arm should go back to its start point independently, which is  $(0,0), (W,0), \dots, ((M - 1) \times W, 0)$ .

## 3. Mathematical Model of Multiarm Cooperative Harvesting Trajectory Planning Problem

It can be known from Section 2.3 that the trajectory optimization problem of an intelligent multiarm straw-rotting fungus harvesting robot is regarded as a multiobjective optimization problem. Researchers mainly use multiobjective optimization methods such as the weighted coefficient method, multiobjective genetic algorithm, multiobjective particle swarm optimization algorithm, etc. [37,38]. Among them, because the optimization problem in this paper is a two-dimensional, that is, a low-dimensional objective optimization problem, and to reduce the complexity of the problem, the simple and easy-to-use weighted coefficient method is adopted, which decomposes multiobjective into a single objective and then optimizes this single objective.

Suppose the robot has  $M$  arms and  $N$  fruits to be harvested. The average harvesting efficiency of the robot is  $C$  pcs/h and the average manual working efficiency is  $M$  pcs/h;  $R$  is the ratio of the average working efficiency of the robot and man. In order to obtain the optimal solution to double the objectives of harvesting efficiency and success rate, take

R and success rate P to solve the objective function E by the objective weighting method as follows:

$$\text{Max } E = R \times K_1 + P \times K_2 \tag{4}$$

$$R = \frac{C}{M} = \frac{N}{T \times M} \tag{5}$$

$$P = \frac{N - N_{\text{lose}}}{N} \times 100\% \tag{6}$$

$$T = [\max(t_{\text{ARM}_i}) + T_c]/3600 \tag{7}$$

$$t_{\text{ARM}_i} = \sum_{j=1}^N \sum_{k=1}^N t_{jk} x_{ijk} + \sum_{j=1}^N t_{d_j} x_{id_j} + \sum_{j=1}^N t_{d_i} x_{ijd_i}, \forall i \in M \tag{8}$$

$$x_{ijk} = \begin{cases} 1, & \text{when Arm } i \text{ from fruit } j \text{ to } k \\ 0, & \text{other} \end{cases} \tag{9}$$

where,

$K_1$  and  $K_2$  are weighting coefficients,  $K_1$  and  $K_2 > 0$  and  $K_1 + K_2 = 1$ ;

$N_{\text{lose}}$  is the number of fruits in the clusters that failed to be harvested in the order from high to low in the Z direction when the fruits were gathered;

T is the total harvesting time required for harvesting all mature fruits;

$T_c$  is the avoidance time;

$x_{id_j}$  indicates that Arm i from the starting point  $d_i$  to fruit j;

$x_{ijd_i}$  indicates that Arm i from fruit j to the starting point  $d_i$

$t_{jk}$  is the time taken by the harvesting arm to perform the task from fruit j to k;

The objective function must satisfy the following constraints:

Given the mature fruits set is  $V = \{1, 2, \dots, N\}$ ; U is the set of harvesting arms ( $1 \leq M \leq N$ ),  $U = \{1, 2, \dots, M\}$ ; Since the working range of each harvesting arm is limited,  $d_i$  is the starting point of Arm i.

$$y_{ij} = \begin{cases} 1, & j \in I_i, \text{ when Arm } i \text{ harvests fruit } j \\ 0, & \text{other} \end{cases} \tag{10}$$

where  $I_i$  means the mature fruits subset must be harvested by Arm i,  $I_i \in V$

The sum of fruits picked by each arm should be equal to N as follows:

$$\sum_{j=1}^n y_{ij} = Q_i, \forall i \in M, \tag{11}$$

$$\sum_{i=1}^m Q_i = n \tag{12}$$

where  $Q_i$  indicates the number of fruits allocated to Arm i,  $i \in U$ ;

Each fruit can only be harvested by one arm as follows:

$$\sum_{i=1}^m y_{ij} = 1, \forall j \in V \tag{13}$$

Each arm should start from and come back to its own start point after harvesting all assigned fruits as follows:

$$\sum_{j=1}^n x_{id_j} = \sum_{j=1}^n x_{ijd_i} = 1, \forall i \in M \tag{14}$$

#### 4. Multiobjective Optimization Algorithm for Multiarm Cooperative Harvesting Trajectory

As discussed in Section 1, there are the following two main difficulties required to overcome: (1) The SSIS region restricts the working area of the harvesting arm as mentioned in Section 2.2; (2) The allocation of cluster fruits to ensure the non-destructive success rate.

For the first problem, the stepwise algorithm, which can divide the MTSP into  $m$ -independent TSPs to reduce the complexity of the problem, is an appropriate approach.

For the second sequence planning problem, the normal approach is to regard the clustered fruits as a whole and assign them to a certain arm. However, when the number of fruits in the same cluster is large and the distribution density of mature fruits in the harvesting area is uneven (for example, most of the mature fruits are located on the left and a few on the right), and if the fruits of a cluster can only be assigned to one arm, there will be more harvesting tasks for one arm and fewer for the other arms, which will greatly affect the harvesting efficiency. Therefore, for clustered fruits, it is necessary not only to ensure that they can be harvested in order but also to be assigned to different arms to help achieve the balanced allocation of the total harvesting task to each arm.

Therefore, an improved genetic algorithm and ant colony stepwise multiobjective optimization algorithm (IGAACMO) is proposed. The algorithm flow is shown in Figure 6.

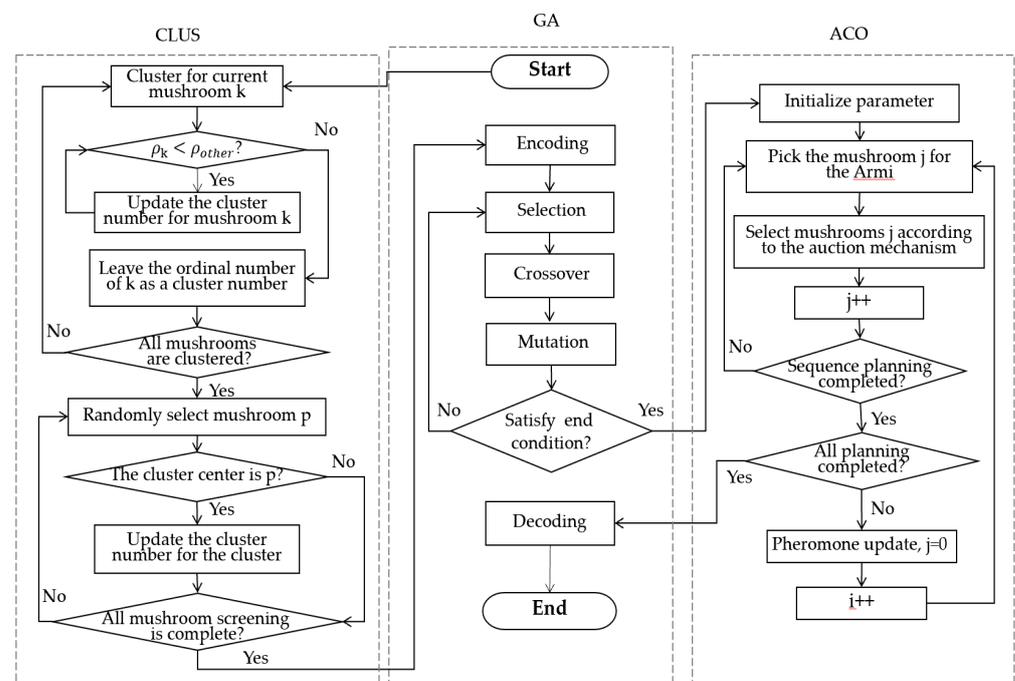


Figure 6. The flow chart of the multitarget optimization algorithm proposed in this paper.

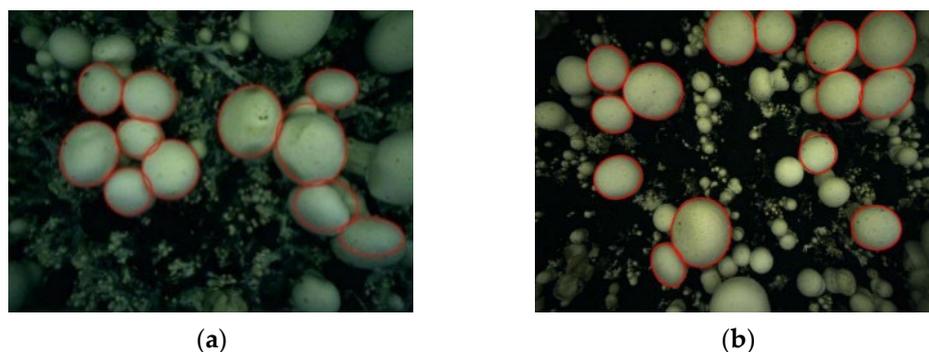
In the first step, an improved local density bi-directional clustering algorithm is designed to identify the clustered fruits to provide preparation for harvesting the clustered fruits in the specified order; Then, in the second step, the MTSP problem is decomposed into  $m$  independent TSP problems by a genetic algorithm with strong global optimization ability, so as to settle uneven task assignment of the MTSP problem with SSIS region; The third step is to use the fast convergence speed of the ant colony algorithm to plan the trajectory of the above  $M$ -independent TSP respectively, and combined it with the auction mechanism to resolve the allocation issue under the restriction of clustering fruit sequence planning.

##### 4.1. Clustering Algorithm Optimization for Fruits of Straw-Rotting Fungus

The clustering algorithms in the current research can be roughly divided into the following five categories: partitional-based, hierarchical-based, grid-based model-based,

and density-based [39]. The partition-based algorithm is suitable for identifying datasets with small sample sizes and spherical cluster shapes. However, it depends on the user to specify the number of clusters in advance, and the processing for large-scale datasets and clusters with complex shapes still needs to be improved further [40]. The hierarchical-based clustering algorithm is sensitive to the noise and abnormal data points in the data and cannot be rolled back after the upward or downward iteration [41]. The grid-based algorithm runs at a high speed because its processing time is only related to the number of cells and has nothing to do with the number of objects. However, the grid-based division method may also reduce the clustering accuracy [42,43]. The advantage of a model-based clustering algorithm is that it can find noise and isolated data points and can automatically identify the number of classes. The disadvantage is that it is not suitable for clustering with a large amount of data [44]. The density-based clustering algorithm can identify clusters with different shapes. It can effectively eliminate abnormal data points or isolated data points in the dataset, and has good noise resistance, but are sensitive to the density of adjacent data points [45,46].

The fruit clustering state of straw-rotting fungus is relatively complex. Taking *Agaricus bisporus* as an example, as shown in Figure 7a, it has the characteristics of complex and different cluster shapes, and the number of clusters is unpredictable in advance, which makes the partition clustering algorithms and hierarchical clustering algorithms unsuitable to discriminate against it. In addition, as shown in Figure 7b, it also has the characteristics of many small clusters and many discrete values globally, which makes grid clustering algorithms and model clustering algorithms less suitable.



**Figure 7.** Cluster shape of *Agaricus bisporus* (a) clusters with different shapes and densities; (b) clusters with many small clusters and many discrete values.

The density-based clustering algorithm can identify clusters with different shapes. However, due to the different diameters of each fruit of the straw-rotting fungus, the density of the fruit clusters of the straw-rotting fungus is uncertain, while the general density-based algorithm is not effective in solving such clusters with variable density. Therefore, an improved local density bi-directional clustering algorithm is designed in this paper. The designed local density calculation method can better adapt to the problems of complex and different cluster shapes, especially for uncertain density, so that the algorithm can better meet the requirements of fruit cluster analysis of straw-rotting fungus.

#### 4.1.1. Fruit Clustering Definition

In order to determine which cluster set should a fruit belong to; the following definitions are given:

- (1) If  $D_{ij}$ , the center distance between fruit  $i$  and  $j$ , is less than or equal to the sum of their radius, as Equation (15), fruits  $i$  and  $j$  belong to the same cluster;

$$D_{ij} \leq r_i + r_j \quad (15)$$

where,

$r_i, r_j$  represent the radius of fruit  $i$  and  $j$ , respectively.

- (2) If fruits  $i$  and  $j$  are in the same cluster, while fruits  $j$  and  $k$  are in the same cluster, then fruits  $i$  and  $k$  are also in the same cluster;
- (3) If the center distance between fruit  $i$  and any other fruit fails to satisfy Equation (15), then the fruit  $i$  does not belong to any cluster, which is called discrete fruit.

#### 4.1.2. Local Density Calculation

Let  $N$  be the set of all mature fruits and  $n$  be the set of the number of fruits. The local density  $\rho_i$  represents the number of fruits that belong to the same cluster (that is, meet the definition (1) in Section 4.1). The larger the local density, the more likely the fruit is the center of the cluster center; when the local density is 0, the fruit is a discrete fruit. The calculation process of local density is as follows:

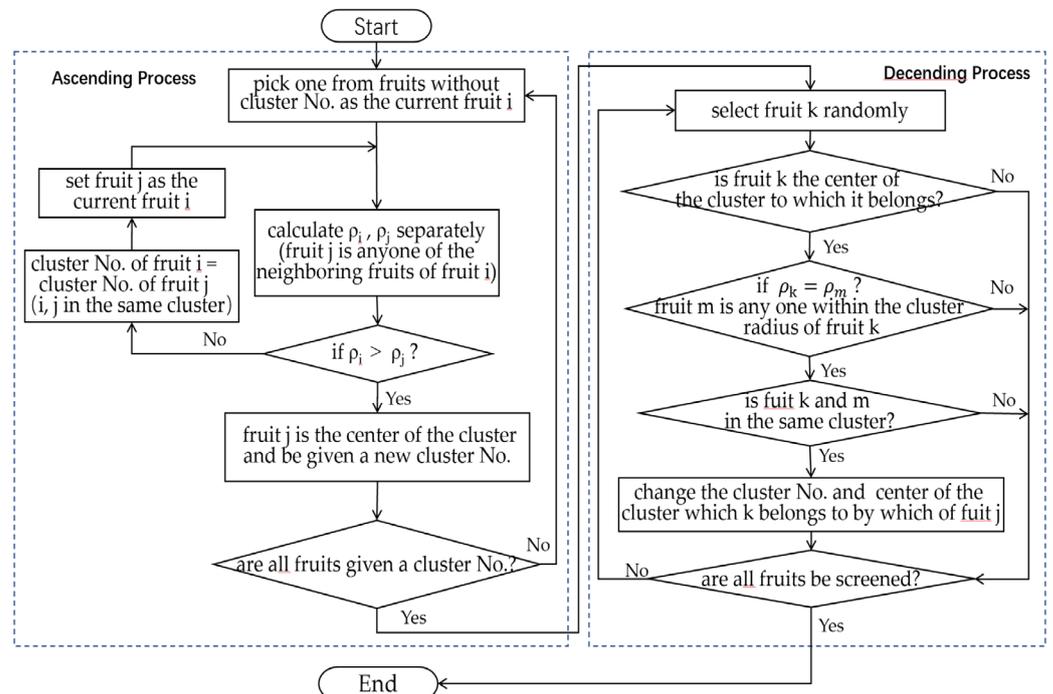
$$\rho_i = \sum_{j=1}^n Clu_{ij}, i \neq j, i, j \in N, \tag{16}$$

$$Clu_{ij} = \begin{cases} 1, & D_{ij} \leq r_i + r_j \\ 0, & D_{ij} > r_i + r_j \end{cases}, i \neq j, i, j \in N, \tag{17}$$

where  $Clu_{ij}$  is used to determine if fruit  $i$  and  $j$  belong to the same cluster.

#### 4.1.3. Improved Density-Based Clustering Algorithm

The improved clustering algorithm is divided into an ascending process and a descending process, as shown in Figure 8.



**Figure 8.** The flow chart of the improved local density-based clustering algorithm.

In the ascending process, calculate the local density of different fruit points, find the high local density point closest to the fruit point, form a data chain from the data points of low local density to high local density, and find the cluster center of the ascending process for all fruits point. The codes in details are shown in Table 1.

**Table 1.** The codes of ascending process of the improved clustering algorithm.

---

**Algorithm 1.**

---

```

Input: N          // set of all mature fruits
Output: Unit[]   // set of cluster centers corresponding to each fruit
1 Initialization();
2 UpProcess(a,b); // a is the data coordinate and b is the data serial number.
3  c = b;          // If no other point within the cutoff distance has a higher cluster
                  // density than it, its cluster center is itself
4  while i < n do;
5    if i! = b and density[i] >= density[b]; // The local density of i is greater
                                           // than the input fruit
6      Distance = math.sqrt((N[i][0]-a [0])2+(N[i][1]-a [1])2);
7      rc = Clu[i][2]+a [2] // Cluster radius
8      if x1 <= x2: // Two fruits are clustered
9        c=i;
10       break;
11      return c;
12 while i < n do; #Ascending process, computing the cluster center for each fruit
13   Unit[i]=UpProcess(N[i],i)
14 return Solution;

```

---

In the descending process, the data point with the highest local density is used as the cluster center, and then the data chain is merged. After all data points are traversed, and finally, clustering is performed to complete the unified operation of all fruit clustering centers in the same cluster, the codes in detail are shown in Table 2. In addition, consolidation operations were added to the descent. Because the growth characteristics of straw-rotting fungus easily lead to the highest local density points within the same cluster, which may not be unique, they need to be integrated into the same cluster. For example, the local density values of A and B in Figure 9 are both equal to 3, which are both the highest local density points in the cluster. In this case, fruit A and B may be the cluster centers of each other, so it is necessary to integrate Fruit A and B into the same cluster. Its processing method is shown in lines 9–10 in Table 2.

**Table 2.** The codes of descending process of the improved clustering algorithm.

---

**Algorithm 2.**

---

```

Input: Unit[]     // The set of cluster centers obtained during the ascent
Output: Unit[]   // The final set of cluster centers corresponding to each fruit
1 Initialization();
2 DownProcess(a,b);
3 if a == b; // If the cluster center of the data point is itself
4   return b;
5 else;
6   a = DownProcess(Unit[a],a);
7   return a;
8 for i in range(n); // Descending process, if the cluster center of a data is another
                    // point, it will be merged into its subclass.
9   if i == Unit[Unit[i]]; // If it is the cluster center with another point, select one
                          // of the points as the cluster center and merge the two.
10  Unit[i] = i;
11  if Unit[Unit[i]]! = Unit[i]; // If the final cluster center corresponding to the
                               // point is found
    Unit[i] = DownProcess(Unit[i],i);
12 return Solution;

```

---

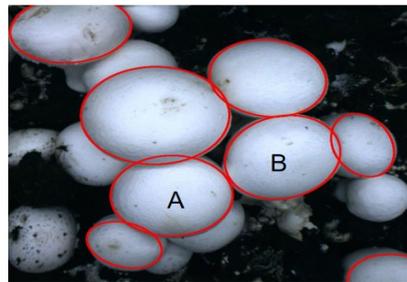


Figure 9. The highest local density point within the cluster is not unique.

4.2. Genetic Algorithm

A genetic algorithm is used to divide the MTSP into 2M-1 independent TSPs for SSIS region constraint by designing an appropriate encoding so that the MTSP can be decomposed into 2M-1 TSPs appropriately.

(1) Encoding

The gene sequence of GA is divided into 2M-1 corresponding segments to the 2M-1 different accessible regions in the SSIS region, as shown in Figure 10 so that the MTSP can be decomposed the MTSP into 2M-1 TSPs. The gene indicates which arm the fruit should be allocated to. The DNA fragment corresponds to the picking region  $S_k$  in Figure 5 one by one.

$S_1$			...	$S_M$				...	$S_{2M-1}$		
1	1	...	...	1	2	M	...	...	M	M	...

Figure 10. Diagram of DNA sequence proposed in this paper.

Each ripe fruit in the reachable region corresponds to each element in the corresponding DNA segment.  $DNA_k(j)$  indicates the picking arm allocated to the j-th fruit in the k-th area. Therefore, in order to be consistent with the harvesting arm allowed to enter each accessible region of Equation (3), the assignment rule of  $DNA_k(j)$  in the initialization of the corresponding population is as follows:

$$DNA_k(j) = \begin{cases} random(1, k), & k \in \{1, 2, \dots, M - 1\} \\ random(k - M + 1, M), & k \in \{M, M + 1, \dots, 2M - 1\} \end{cases} \quad (18)$$

where  $random(1, k)$  represents any integer in the randomly assigned closed interval 1 to k.

(2) Selection operator

Roulette is adopted as the selection operator to improve the optimization ability of the algorithm. In this method, two individuals are selected at a time, and then the individual with the better fitness of the two individuals is selected by the probability of survival.

(3) Crossover operator

To increase the global search ability, a multipoint crossover is used to randomly select multiple segments in the gene sequence for crossover.

(4) Mutation operator

Different mutation rules are required for each DNA segment, and it can merely mutate into the code for the harvesting arm accessible to the corresponding reachable area of the segment.

All of the mature mushrooms in the current cycle were divided into M groups based on the DNA sequences of the best individuals in the population.

### 4.3. Improved Ant Colony Algorithm

In order to solve the problems of sequence harvesting of cluster fruits and collision avoidance when M harvesting arms work together, the respective trajectory planning of each harvesting arm should be carried out in parallel, so that it can be judged in real-time whether there the clustered fruits are harvested in the specified order and whether will be collisions between the arms.

The ant colony algorithm has good parallelism and late convergence of the algorithm, so this paper adopts the ant colony algorithm to solve the trajectory planning problem of each of the M harvesting arms and combines the auction mechanism to deal with the sequence harvesting of cluster fruits when the M harvesting arms work together.

In actual harvesting, in addition to harvesting efficiency, the harvesting success rate is also a very important indicator. According to actual harvesting requirements and experiments, the trajectory planning algorithm designed in this paper needs to ensure that the harvesting success rate is more than 95%. Therefore, the following approach is designed so that in the early stage of the evolution of the algorithm, the success rate is the main guide, while after the success rate meets the requirements, the pheromone concentration of the current fruit to be harvested should be temporarily increased to increase the probability of its selection.

The specific calculation process for the success rate is shown in Figure 11.

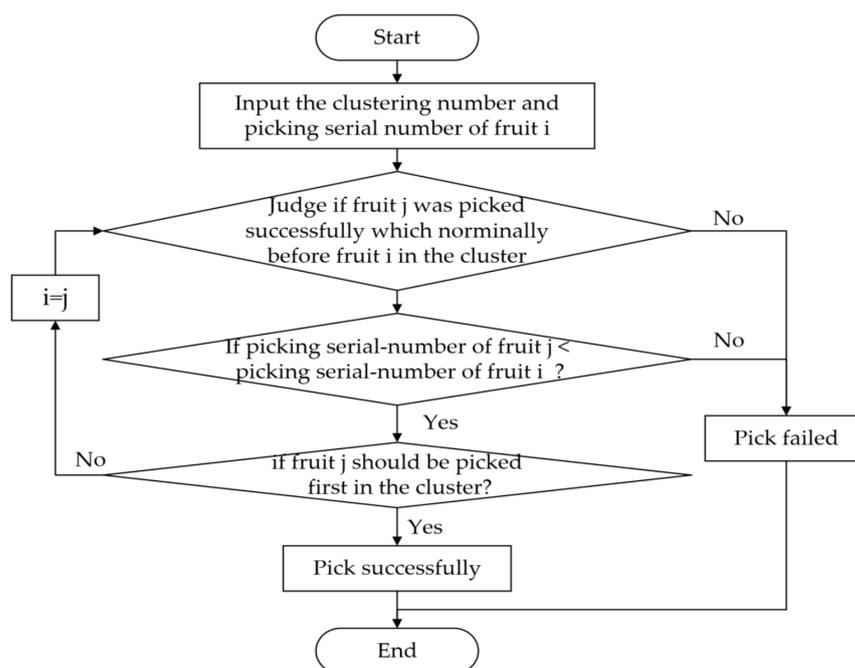


Figure 11. Flow chart for judging whether fruit in the cluster be harvested in specific order.

The following are the detailed steps:

Step 1 Initialize.

Initialize the pheromone matrix, the path taboo table, the set containing the nominal harvesting order of clustered fruits (for calculating the success rate), the set of coordinates of the fruit to be harvested for each arm, and the matrix containing the information corresponding to the time axis and displacement of the X-axis.

Step 2 Build trajectory.

Ants construct m-picking arms in parallel. First, the path taboo table is used to remove the picked fruits and generate a preliminary candidate fruit set. Additionally, then, the auction mechanism is used to determine the current candidate fruit set for each ant and choose the fruit to be picked next from the set according to the pheromone concentration until all ants have completed the trajectory construction.

### Step 3 Evaluation.

The objective function  $E$ , which can be calculated by Equation (4), is used for evaluation. To make the success rate of harvesting meet the requirement of more than 95%, the  $K_1$  and  $K_2$  coefficients in Equation (4) are dynamically adjusted. When the success rate is less than 95%, set  $K_1 = 0.4$ ,  $K_2 = 0.6$ ; after the success rate is greater than or equal to 95%, set  $K_1 = 0.6$ ,  $K_2 = 0.4$ .

### Step 4 Update the pheromone matrix.

The trajectory with maximum  $E$  in Equation (3) is selected to update the pheromone matrix.

### Step 5 Determine the number of iterations.

If the maximum number of iterations is reached, turn to End, otherwise go to step 1.

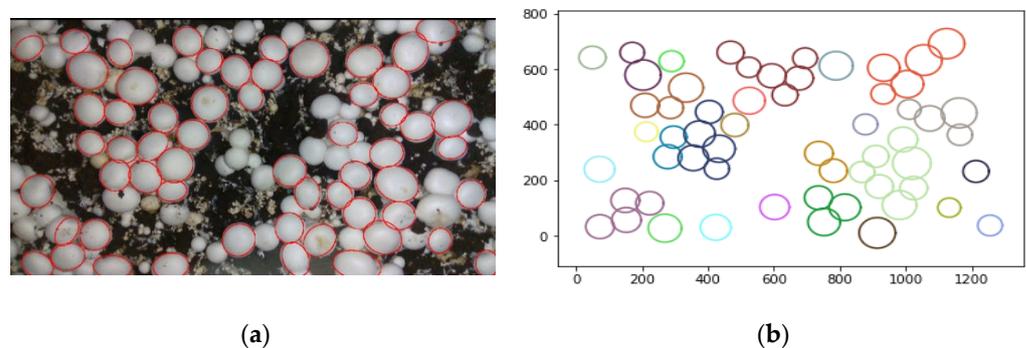
### Step 6 End.

Output the final optimal trajectory.

## 5. Experiments and Analysis

### 5.1. Experiments of Clustering Algorithm

To verify the effect of the improved local density clustering algorithm (ILDCA) in this paper, the agaricus bisporus was taken as an example to test, and the test data were all from the site of the planting factory. The data is shown in Figure 12a. The pictures were taken on the spot by the harvesting robot, and the mature fruits recognized by visual are marked with red circles. The mature fruit data obtained from the image identification are processed by the clustering algorithm proposed in this paper, and the obtained clustering result is shown in Figure 12b, in which the fruits belonging to the same cluster are marked with the same color. Comparing a and b of Figure 12, it can be seen that the success rate of clustering is close to 97%, which fully meets the requirement of clustering identification in robotic harvesting.



**Figure 12.** The processing result of clustering algorithm processing. (a) visual identity map of ripe fruits; (b) The processing result of the improved clustering algorithm.

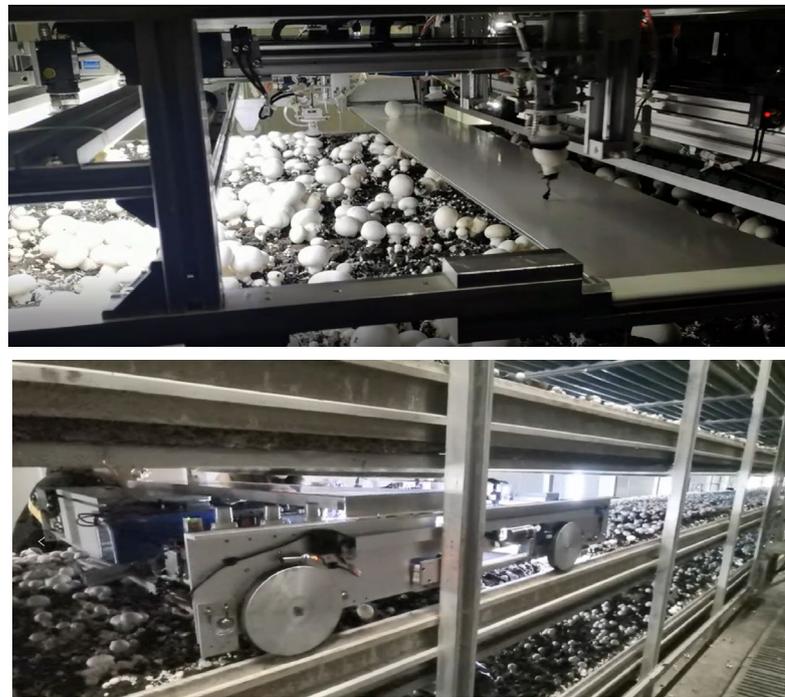
In order to further verify the effectiveness of the algorithm, this paper selects much more samples with different fruit numbers, cluster numbers, and discrete point numbers to conduct multiple sets of experiments and compares them with the commonly used clustering algorithm K-means algorithm and Gaussian mixture algorithm. The results are shown in Table 3. It shows that the K-means algorithm has the worst effect in processing the clustering of the fruits in this paper, whose average success rate is only 68%. Compared with the K-means algorithm, the Gaussian mixture algorithm is more flexible in the shape of the clustering, but it is more difficult to adapt to the characteristics of this paper with many small clusters and many discrete values, and the average success rate is merely 78%. However, the effect of the improved algorithm is much better than the other algorithm, with its average success rate is up to 97%. Additionally, as the number and complexity of clusters increase, the superiority of the improved clustering algorithm remains stable. So, the improved clustering algorithm is suitable to solve the clustering problem of straw-rotting fungus.

**Table 3.** Clustering algorithm comparison of many groups of samples.

Group	Num of Fruits	Num of Cluster	Num of Discrete Points	The Recognition Success Rate of Clustered Fruits		
				K-Means	Gaussian Mixture	Improved Clustering Algorithm
1	40	18	12	82%	88%	99%
2	40	25	20	78%	85%	99%
3	55	27	16	72%	80%	98%
4	55	38	27	68%	78%	96%
5	70	36	28	63%	73%	96%
6	70	45	36	57%	66%	95%
AVG	55	189	139	68%	78%	97%

### 5.2. Experiments of Multiobjective Optimization Algorithm for Multiarm Cooperative Harvesting Trajectory

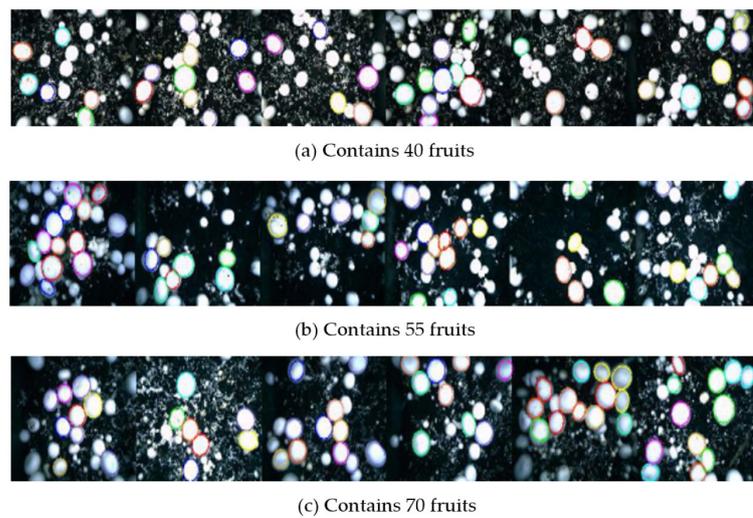
Take the three-arm *Agaricus bisporus* harvesting robot as an example to verify the effect of the proposed approach, as shown in Figure 13.



**Figure 13.** The multiarm intelligent harvesting robot working in the multistory shelf trays in the factory environment.

Three sets of data containing 40, 55, and 70 fruits, respectively, are selected as the first experimental data, which are shown in Figure 14. The detailed harvesting information for the fruits to be harvested in Figure 14 is shown in Appendix A, where  $(X, Y, Z, C)$  is used to express the harvesting information for fruits.  $X$ ,  $Y$ , and  $Z$  represent the coordinates of the center point of the fruit to be harvested.  $C$  indicates the cluster number the fruit should belong to, which can be obtained by the clustering algorithm proposed in this paper.

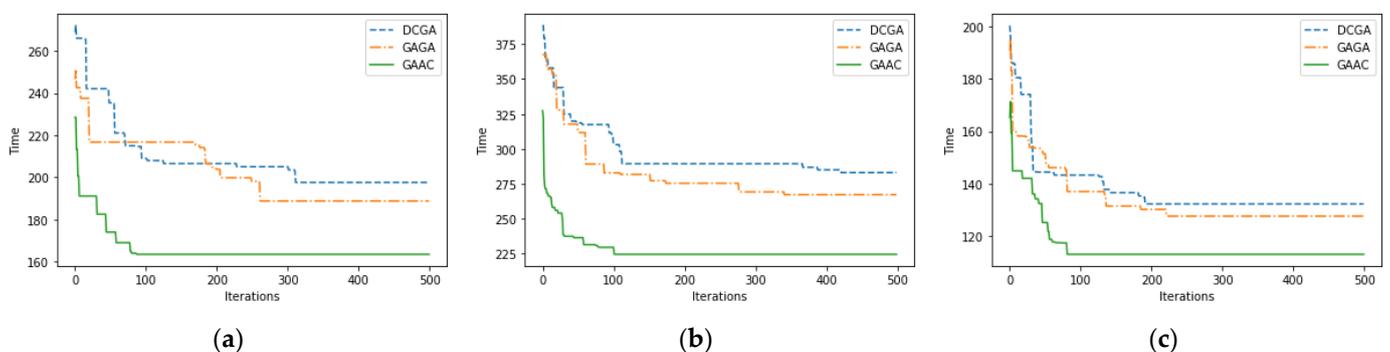
The proposed IGAACMO algorithm is used to optimize the harvesting trajectory of the real fruit data (Figure 14). Furthermore, the two-chromosome genetic algorithm (DCGA) and the genetic stepwise algorithm (GAGA) are also used to plan the trajectory of the three-arm robot with the experimental data to compare with the processing results of the algorithm proposed in this paper.



**Figure 14.** Pictures of fruit harvesting area from planting factory. (a) contain 40 fruits; (b) contain 55 fruits; (c) contain 70 fruits.

The parameter settings are as follows: (1) the crossover probability is set to 0.15, the population is set to 30, the mutation probability is set to 0.015, and the maximum iteration number is set to 500. (2) In the ant colony algorithm, let 1 be the set of the important factors of pheromone, let 30 be the set of the number of ants, let 5 be the set of the intensity of pheromone, and let 10 be the set of the important factors of heuristic pheromone, let 0.1 be the set of the volatile factors and let 500 be the set of the maximum iteration number; (3) the moving speed of the harvesting arm ( $V$ ) is given as 100 mm/s and the harvesting execution time ( $t_1$ ) is given as 5 s.

The convergence performance of the algorithm is shown in Figure 15. It indicates that when the picking scale is 40 (i.e., 40 fruits need to be picked), the iteration number of the proposed algorithm is about 50% less than that of DCGA, 67% less than that of GAGA, and the optimal harvesting time of GAAC is 14% better than DCGA and 11% better than GAGA; When the picking scale is 55, the iteration number of GAAC is about 67% less than that of DCGA, and 75% less than that of GAGA, and the optimal harvesting time of GAAC is 22% better than DCGA, and 15% better than GAGA; When the picking scale is 70, the iteration number of the proposed algorithms is about 28% less than that of GAGA, about 22% less than that of GAGA, and the optimal harvesting time of the proposed algorithm is 26% better than DCGA, and 19% better than GAGA. Therefore, compared with the other two methods, the convergence speed and optimization ability of the algorithm proposed in this paper are better.



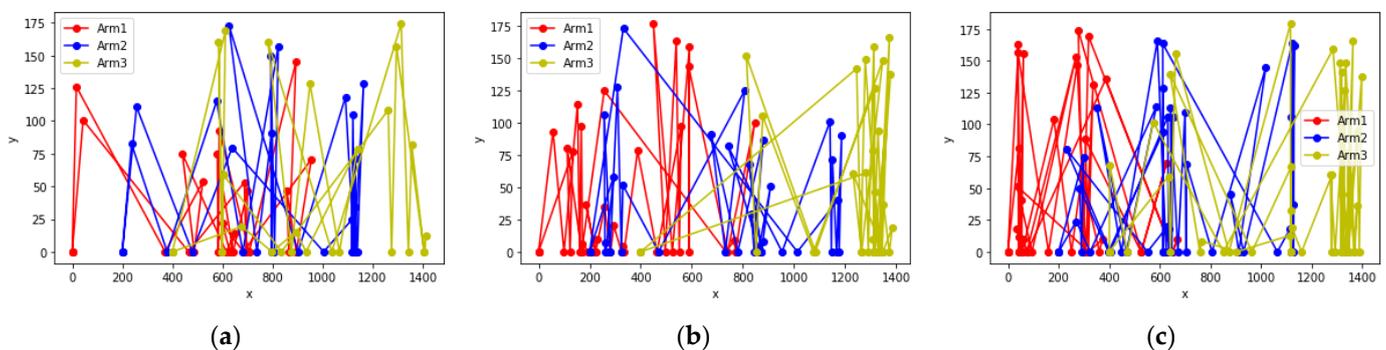
**Figure 15.** Comparison of convergence performance of three algorithms. (a) contain 40 fruits; (b) contain 55 fruits; (c) contain 70 fruits.

The other results obtained by the three algorithms and the important parameters are shown in Table 4. By comparing these parameters, the following can be seen: (1) The algorithm proposed in this paper has the best multiarm task distribution uniformity and the highest utilization of multiarm cooperation. The greater the number of fruits to be harvested, the more obvious the advantages compared with the other two algorithms; (2) The harvesting success rate after using the improved algorithm in this paper can always be guaranteed to be above 95%.

**Table 4.** Comparison of experimental results of the three algorithms.

Num of Fruits	Algorithm	Avoid Times	Harvesting Time(s)	Harvesting Efficiency(pcs/h)	Harvesting Success Rate
40	DCGA	2	130.21	1106	96%
	GAGA	1	126.54	1138	98%
	IGAACMO	0	113.81	1265	100%
55	DCGA	5	196.16	1009	90%
	GAGA	3	185.42	1068	94%
	IGAACMO	1	161.31	1227	98%
70	DCGA	9	283.29	890	82%
	GAGA	6	267.95	940	87%
	IGAACMO	2	224.64	1122	96%

The harvesting trajectory optimized by the IGAACMO algorithm is presented in Figure 16. The harvesting assignment task of each harvesting arm is relative balance, and there is basically no redundancy in the trajectories.



**Figure 16.** Trajectory diagram optimized by IGAACMO algorithm. (a) contain 40 fruits; (b) contain 55 fruits; (c) contain 70 fruits.

Because the larger the ratio of the number of clusters to the total number of fruits is, the more it will affect the performance of the algorithm. Another 10 more experiments are added to verify the stability of the proposed algorithm further. The number of ripe fruits ranges from 40 to 70, with the proportion of clusters ranging from 20% to 60% as well.

The results of the ten group experiments are shown in Table 5. The average harvesting efficiency optimized by the proposed algorithm is 1183 pcs/h, which is about 21% higher than that of the DCGA algorithm and about 15% higher than that of the GAGA algorithm. In the meantime, the average harvesting success rate is 97%, much better than the other two algorithms as well. All of the group results are basically consistent with Table 4. This indicates that the algorithm designed in this paper can achieve a better harvesting trajectory for the multiarm intelligent harvesting robot for fruits with different distributions.

**Table 5.** Comparison of the three algorithms with 10 groups data.

Group	Num of Fruits	Ratio of Cluster	Harvesting Efficiency (pcs/h)			Harvesting Success Rate		
			GAGA	DCGA	IGAACMO	GAGA	DCGA	IGAACMO
1	40	20%	1162	1182	1308	98%	99%	100%
2	42	38%	1112	1141	1258	97%	98%	100%
3	45	58%	1046	1093	1114	92%	95%	98%
4	49	21%	1068	1114	1238	95%	98%	100%
5	52	30%	1027	1074	1232	92%	95%	98%
6	57	45%	992	1034	1205	90%	94%	98%
7	60	60%	931	983	1161	85%	90%	95%
8	62	23%	945	1008	1186	86%	91%	97%
9	67	46%	892	947	1132	81%	87%	96%
10	70	57%	849	896	1106	78%	85%	95%
AVG	54.4	41%	977	1025	1183	88%	92%	97%

## 6. Discussion

The harvesting trajectory planning of a multiarm straw-rotting fungus harvesting robot is a typical NP-hard problem. It can be better optimized by the IGAACMO algorithm, which is proposed in this paper.

In terms of running speed, the IGAACMO is obviously superior to the other two methods (DCGA and GAGA). Moreover, the larger the processing scale (the more fruits to be picked), the greater the convergence advantage.

In terms of the optimization results, the amount and distribution of ripe fruits have an impact on the results. The algorithm is sensitive to the distribution density of the fruit to be picked. With the increase in fruit density, the picking efficiency will decrease. This shows that the closer the fruit distribution is, the more difficult it is to avoid a collision, which makes some picking arms have to wait and reduces the picking efficiency. However, compared with the other two algorithms, the optimization effect of the proposed algorithm is better under the same conditions, especially in the case of the fruit distribution with high density.

In particular, there is another important issue with the fruit cluster that needs to be harvested in a specific order. There are two ways to deal with this issue. One is to regard the fruits in the same cluster as a whole and assign them to the same arm to harvest them in a specific order, which is mostly adopted at present. The other is to allocate them to multiple different arms on the premise of ensuring the required harvesting order, which is an improved method proposed in this paper. The latter method is superior to the former one, especially when the distribution of fruits in each accessible area is seriously uneven, with large fruit clusters stretching across two different accessible areas as well, as shown in Figure 17. The red circle represents the fruit to be picked, and the black circle represents the immature fruit. The picking robot has three arms, and the working area is divided into five accessible areas, where  $S_1$  (1,1) represents the exclusive area for Arm1, and the fruits in this area can only be harvested by arm1,  $S_2$  (1,2) is the partial shared area that can only be harvested by Arm 1 and Arm 2,  $S_3$ (1,3) is the fully shared area that can be harvested by all three arms,  $S_4$  (2,3) is the partial shared area that can be harvested by Arm 2 and Arm 3, and  $S_5$ (3,3) is the exclusive area that can only be harvested by arm 3. Most of the fruits to be picked are distributed in the exclusive area  $S_1$  (1,1) and the partial shared area  $S_2$  (1,2), and there is a large fruit cluster C1 over the two areas, meanwhile. The comparison of the results of the above two methods is shown in Table 6. All of the fruits in the fruit cluster C1 are allocated to Arm2 and the number of fruits allocated to Arm1 is very few by using the GAAC algorithm, which greatly increases the cycle time. However, by comparison, C1 is split and assigned to Arm1 and Arm2 respectively, resulting in a more uniform harvesting task among each arm, thereby improving the harvesting efficiency further.

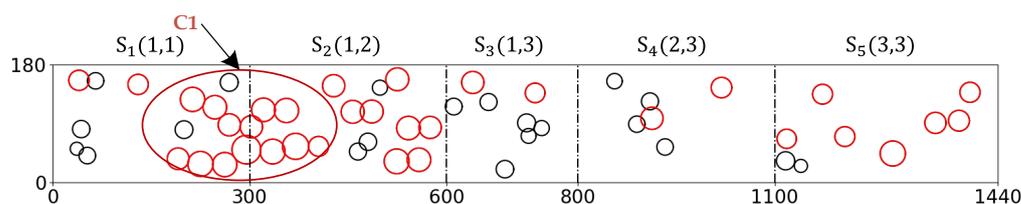


Figure 17. Diagram with serious uneven distribution of fruits.

Table 6. Comparison results of different allocation methods for fruit clusters.

Algorithm	Avoid Times	Harvesting Time(s)	Harvesting Success Rate	N1:N2:N3
GAAC (assigned to the same arm)	1	126.29	100%	03:17:14
IGAACMO (assigned to different arms)	1	105.25	100%	13:12:09

It can be shown that when the distribution of mature fruits is seriously ununiformly, with some fruit clusters across multiple accessible areas as well, it is easy to cause the uneven task assigned to each arm by assigning the total fruits in a cluster merely to the same arm, which results in some arms waiting for no picking tasks meanwhile other arms have too many picking tasks. This will greatly increase the cycle time of harvesting, thereby seriously reducing the picking efficiency. The algorithm proposed in this paper, combined with the auction mechanism, can allocate the fruits in a cluster to different arms on the premise of ensuring the required harvesting order instead of allocating them to a single arm, which can resolve this issue appropriately. Therefore, it can be concluded from all the above discussions that the algorithm proposed in this paper has strong optimization ability and good stability. For fruits with different densities, the picking tasks for each arm can be evenly distributed even though the fruits are not uniformly distributed on the culture medium or soil, with some fruit clusters across multiple accessible areas as well; thereby, it can not only achieve higher harvesting efficiency but also a higher success rate. The algorithm can better adapt to the issues of dense and uneven distribution of fruits caused by the natural growth of straw-rotting fungus.

### 7. Conclusions

This paper takes a straw-rotting fungus multiarm harvesting robot as the research object. Aiming at the problem of uniform task allocation and sequential harvesting for clustered mature fruits in multiarm cooperative harvesting trajectory optimization, an improved multiobjective optimization algorithm, IGAACMO, is proposed. The multiarm cooperative harvesting trajectory planning is abstracted to an MTSP problem. We use an improved local density bi-directional clustering algorithm to identify the clustered fruits to provide preparation for harvesting the clustered mature fruits in the specified order; Then, GA is adopted to decompose the MTSP into  $m$  independent TSP problems, where a new DNA coding method is designed to make the harvesting task of each harvesting arm evenly distributed under the constraining of the SSIS area. Subsequently, we use the ant colony algorithm to plan the trajectory of the above  $M$ -independent TSP, respectively; Here, by combining with the auction mechanism, the clustered fruits can be planned to be harvested in their specified order.

From all the above experiments and discussion, it can be shown that the optimization ability of the proposed algorithm, IGAACMO, is significantly stronger than the other two methods. The average harvesting efficiency optimized by the proposed algorithm is up to 1183 pcs/h, and the average harvesting success rate is 97%.

In addition, since the hourly harvesting efficiency of the multiarm robot has reached the manual efficiency, the daily harvesting efficiency of the robot will be significantly higher

than the manual, even if it can be up to at least twice that of the manual. Because the robot can work for at least 16 h per day (considering battery replacement, layer change, and other auxiliary work), while people generally work for 8 h per day. This efficiency greatly increases the feasibility of the robot applied to the actual harvesting of straw-rotting fungus instead of manual.

However, the operation time of the algorithm is not faster enough. In future research, the algorithm needs to be improved to increase its efficiency of the algorithm to improve its real-time control further.

**Author Contributions:** S.Y.: Writing—original draft, writing—review and editing, conceptualization, methodology, investigation, data curation, formal analysis, software, validation, production of related equipment, funding acquisition. B.J.: writing—original draft, writing—review and editing, data curation, validation, software, production of related equipment. T.Y.: investigation, conceptualization, project administration. J.Y.: investigation, writing-review and editing, conceptualization. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Shanghai Science and Technology Innovation Action Plan-Agriculture, Grant No. 21N21900600; Shanghai Agriculture Applied Technology Development Program, Grant No. 2019-02-08-00-10-F01123; Major Scientific and Technological Innovation Project of Shandong Province, Grant No. 2022CXGC010609-7; Mechanical Engineering (Intelligent Manufacturing) Plateau Discipline of Shanghai Polytechnic University.

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

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Anyone can access our data by sending an email to szyang@sspu.edu.cn.

**Acknowledgments:** We acknowledge the Shanghai Lianzhong edible fungus professional cooperative for the use of their mushrooms and facilities.

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

## Appendix A

**Table A1.** The detail data with the information of the fruits to be harvested in Figure 14.

Group	Number of Fruits	Detail Information of Fruits
a	40	(168.43, 136.53, 163.29, 0.0), (72.03, 129.46, 170.47, 0.0), (130.93, 121.4, 165.45, 0.0), (183.02, 162.07, 149.93, 0), (75.41, 10.54, 170.04, 0), (33.48, 35.3, 190.66, 0), (11.47, 98.3, 212.67, 0), (84.48, 168.46, 173.35, 0), (62.15, 75.78, 183.99, 0), (13.53, 133.29, 188.81, 0), (124.38, 8.55, 159.71, 0), (406.7, 39.63, 143.94, 0), (431.28, 77.04, 134.9, 0), (288.84, 157.6, 169.56, 0), (296.91, 43.28, 170.86, 0), (399.99, 87.99, 158.08, 0), (332.95, 7.87, 162.36, 0), (262.74, 170.84, 160.23, 0), (325.74, 175.71, 168.02, 0), (514.44, 43.76, 156.76, 0), (702.29, 35.59, 172.61, 0), (646.05, 34.69, 171.18, 0), (763.04, 101.5, 154.61, 0), (874.45, 137.36, 156.47, 0), (892.59, 65.13, 159.9, 0), (825.83, 162.81, 161.57, 0), (741.06, 76.6, 167.01, 0), (1087.35, 28.05, 152.2, 0), (1049.77, 102.33, 159.7, 0), (1128.99, 53.9, 166.76, 0), (1165.21, 13.61, 173.77, 0), (1013.76, 10.89, 175.49, 0), (1167.15, 171.3, 161.97, 1), (1054.98, 64.92, 169.91, 1), (1385.3, 155.13, 173.38, 2), (1312.73, 160.07, 175.86, 2.0), (1239.82, 172.1, 176.6, 3.0), (1274.88, 117.37, 179.35, 3.0), (1236.71, 9.1, 168.9, 4.0), (1347.87, 167.54, 160.21, 4.0) (56.71, 93.06, 163.55, 0.0), (167.53, 97.5, 155.25, 0.0), (132.8, 114.33, 156.23, 0.0), (134.06, 77.44, 135.51, 0.0), (229.94, 9.72, 159.53, 0), (111.97, 80.42, 142.6, 0), (172.3, 6.49, 160.79, 0), (186.14, 36.68, 153.34, 0), (309.16, 127.35, 170.77, 0), (257.31, 124.79, 170.59, 0), (258.59, 34.41, 172.91, 0), (294.92, 20.13, 170.75, 0), (293.61, 58.17, 178.57, 0), (387.83, 78.1, 139.3, 0), (326.2, 142.62, 171.57, 0), (332.02, 172.98, 173.62, 0), (331.68, 4.6, 174.6, 0), (449.56, 176.44, 179.32, 0), (260.4, 6.76, 165.98, 0), (256.9, 106.31, 250.58, 0), (330.21, 51.61, 159.84, 0), (558.02, 96.99, 154.04, 0), (675.93, 90.7, 158.23, 0), (538.99, 162.97, 171.9, 0), (590.67, 158.46, 147.43, 2), (587.62, 143.37, 160.68, 2), (846.49, 100.02, 145.07, 0), (813.59, 151.48, 152.09, 0), (823.82, 67.78, 155.26, 0), (807.23, 124.87, 148.15, 0), (877.89, 104.9, 155.6, 4), (878.81, 86.16, 155.87, 4), (832.51, 29.5, 159.6, 0), (908.39, 50.74, 169.97, 0), (879.47, 8.38, 160.7, 0), (741.88, 81.64, 184.95, 0)
b	55	

Table A1. Cont.

Group	Number of Fruits	Detail Information of Fruits
c	70	(1186.72,90.04,162.32,3),(1140.91,100.87,159.13,3),(1172.68,40.46,169.81,3),(1150.34,71.37,160.56,3), (1347,148.08,151.3,0),(1315.87,126.89,158.6,0),(1282.68,61.11,149.35,0),(1278.79,148.58,163.01,0), (1312.53,159.13,147.72,0),(1316.57,45.91,154.21,0),(1231.16,60.71,167.48,0),(1350.77,36.87,165.88,0), (1308.44,78.13,163.01,0),(1245.3,141.76,174.2,0),(1373.8,166.04,151.64,0),(1384.82, 18.79, 172.77, 4.0), (1378.14,137.58,162.09,4.0),(1329.02, 93.26, 136.63, 5.0), (1309.99, 9.79, 173.65, 5.0) (36.54,51.33,173.4,0.0),(34.66,18.28,172.73,0)(38.94,163.27,161.22,0.0),(39.91,156.28,169.74,0.0), (43.31,81.47,159.44,0.0),(43.32,12.06,168.63,0.0),(52.59,40.87,167.5,0),(61.23,155.52,153.22,0), (62.45,9.62,183.5,0),(181.64,103.87,144.54,0),(230.97,80.64,171.97,0),(268.83,153.14,161.5,0), (270.25,23.83,174.2,0),(273.84,146.9,172.97,0),(284.45,49.35,157.29,0),(291.88,60.67,169.3,0), (299.04,74.55,181.48,0),(351.08,113.39,162.1,0),(372.31,9.62,153.22,0),(386.24,135.54,167.47,0), (402.12,68.05,207.25,0),(574.22,101.72,166.15,0),(586.16,114.01,158.9,0),(590.18,165.77,157.53,0), (611.49,163.4,160.1,0),(626.03,69.38,172.7,0),(628.9,19.62,175.28,0),(631.23,105.9,183.47,0), (633.71,59.05,175.13,0),(635.86,18.45,164.45,0),(640.67,139.46,150.91,0),(641.93,113.39,152.45,0), (652.2,105.64,172.64,0),(661.64,155.52,169.74,0),(669.45,9.62,178.28,0),(699.63,109.9,172.31,0), (705.25,69.05,267.25,0),(765.4,8.45,166.08,0),(878.4,45.15,168.06,0),(1018.96,145.14,172.04,0), (1114.33,12.86,174.04,0),(1115.94,179,177.31,0),(1116.67,32.84,154.18,0),(1117.28,105.43,159.69,0), (1118.38,66.59,162.1,0),(1122.56,18.75,153.22,0),(1122.93,163.7,183.5,0),(1130.49,162.3,239.41,0), (1274.22,60.15,153.1,0),(1280.38,60.58,168.15,0),(1283.99,159.01,147.51,0),(1320.04,45.58,155.63,0), (1331.13,126.89,158.6,0),(1334.07,148.09,151.35,0),(1341.03,78.13,163.01,0),(1363.14,166.09,151.68,0), (1378.89,36.51,167.16,0),(1380.81,94.34,174.28,0),(1397.66,137.94,163.47,0),(277.82,174.17,165.13,1), (318.43,169.18,164.42,1),(1127.94,37.02,146.94,2),(1115.54,18.48,164.17,2),(1311.72,148.48,163.4,3), (1317.97, 141.62, 173.72, 3.0), (1325.74, 8.28, 177.16, 3.0), (336.15, 131.55, 249.33, 4.0), (305.25, 88.58, 157.51, 4.0), (611.4, 128.39, 144.99, 5.0), (612.86, 94.02, 152.85, 5.0)

## References

- Guo, H.; Ren, P.; Huang, C. Green prevention and control technology of straw rotting fungus. *Edible Med. Mushrooms* **2019**, *3*, 218–221.
- Noble, R.; Reed, J.N.; Miles, S.; Jackson, A.F.; Butler, J. Influence of mushroom strains and population density on the performance of a robotic harvester. *J. Agric. Eng. Res.* **1997**, *52*, 215–222. [[CrossRef](#)]
- Schiau, H.G. Considerations on the evolution of mushrooms harvesting systems. *Res. Sci. Today* **2013**, *7*, 170–182.
- Reed, J.N.; Tillett, R.D. Initial experiments in robotic mushroom harvesting. *Mechatronics* **1994**, *4*, 265–279. [[CrossRef](#)]
- Reed, J.N.; Mrilles, S.J.; Butler, J.; Baldwin, M.; Noble, R. Automatic mushroom harvester development. *J. Agric. Eng. Res.* **2001**, *78*, 15–23. [[CrossRef](#)]
- Yu, G.H.; Luo, J.M.; Zhao, Y. Region labeling technology and mushroom image segmentation method based on sequential scanning algorithm. *J. Agric. Eng.* **2006**, *4*, 139–142.
- Lu, C.P.; Liaw, J.J.; Wu, T.C.; Hung, T.F. Development of a Mushroom Growth Measurement System Applying Deep Learning for Image Recognition. *Agronomy* **2019**, *9*, 32. [[CrossRef](#)]
- Kuchinskiy, N.A. Development of an Autonomous Robotic Mushroom Harvester. Master's Thesis, The University of Western Ontario, London, ON, Canada, 2016.
- Huang, M.; He, L.; Choi, D.; Pecchia, J.; Li, Y. Picking dynamic analysis for robotic harvesting of *Agaricus bisporus* mushrooms. *Comput. Electron. Agric.* **2021**, *32*, 185–256. [[CrossRef](#)]
- Lu, W.; Wang, P.; Wang, L.; Deng, Y. Design and experiment of flexible gripper for mushroom non-destructive picking. *Trans. Chin. Soc. Agric. Mach.* **2020**, *51*, 28–36.
- Yang, S.; Ni, B.; Du, W.; Yu, T. Research on an improved segmentation recognition algorithm of overlapping *Agaricus bisporus*. *Sensors* **2022**, *22*, 3946. [[CrossRef](#)]
- Huang, M.; Jiang, X.; He, L.; Choi, D.; Pecchia, J.; Li, Y. Development of a robotic harvesting mechanism for button mushroom. *Trans. ASABE* **2021**, *64*, 565–575. [[CrossRef](#)]
- Mohanani, M.G.; Salgaonkar, A. Robotic mushroom harvesting by employing probabilistic road map and inverse kinematics. *BOHR Int. J. Future Robot. Artif. Intell.* **2021**, *1*, 1–10. [[CrossRef](#)]
- Yang, Y. Key Techniques for Automatic Mushroom Picking System Based on Machine Vision. Master's Thesis, Nanjing University of Aeronautics and Astronautics, Nanjing, China, 2019.
- Hu, X.M.; Wang, C.; Yu, T. Design and application of visual system in the *Agaricus bisporus* picking robot. *JOP Conf. Ser.* **2019**, *1187*, 032034.
- Tinoco, V.; Silva, M.F.; Santos, F.N.; Rocha, L.F.; Santos, L.C. A review of pruning and harvesting manipulators. In Proceedings of the 2021 IEEE International Conference on Autonomous Robot Systems and Competitions, ICARSC 2021, Santa Maria da Feira, Beja, Portugal, 28–29 April 2021; pp. 155–160.

17. Nas, B.; Nrv, A.; Vsh, A.; Vd, A. Cotton harvester through the application of machine learning and image processing techniques. *Mater. Today Proc.* **2021**, *47*, 2200–2205.
18. Zion, B.; Mann, M.; Levin, D.; Shilo, A.; Rubinstein, D.; Shmulevich, I. Harvest-order planning for a multiarm robotic harvester. *Comput. Electron. Agric.* **2014**, *103*, 75–80. [[CrossRef](#)]
19. Xiong, Y.; Ge, Y.; Grimstad, L.; From, P.J. An autonomous strawberry-harvesting robot: Design, development, integration, and field evaluation. *J. Field Robot.* **2020**, *37*, 202–224. [[CrossRef](#)]
20. Rong, J.; Wang, P.; Yang, Q.; Huang, F. A field-tested harvesting robot for oyster mushroom in greenhouse. *Agronomy* **2021**, *11*, 1210. [[CrossRef](#)]
21. Barnett, J.; Duke, M.; Chi, K.A.; Shen, H.L. Work distribution of multiple cartesian robot arms for kiwifruit harvesting. *Comput. Electron. Agric.* **2020**, *169*, 105202. [[CrossRef](#)]
22. Jia, B.C.; Yang, S.Z.; Yu, T. Research on three picking arm avoidance algorithms for agaricus mushroom picking robot. In Proceedings of the 2020 IEEE International Conference on Advances in Electrical Engineering and Compute Applications, AEECA 2020, Dalian, China, 25–27 August 2020; pp. 325–328.
23. Kurtser, P.; Edan, Y. Planning the sequence of tasks for harvesting robots. *Robot. Auton. Syst.* **2020**, *131*, 103591. [[CrossRef](#)]
24. Venkatesh, P.; Singh, A. Two metaheuristic approaches for the multiple traveling salesperson problem. *Appl. Soft Comput.* **2015**, *26*, 74–89. [[CrossRef](#)]
25. Zhang, H. The Area Divison of Express Delivery Terminal and Vehicle Routing Problem with Balance. Master's Thesis, Beijing Jiaotong University, Beijing, China, 2019.
26. Yin, Y. Research on Path Optimization of Warehouse Robot Based on Improved Genetic Algorithm. Master's Thesis, Chengdu University of Technology, Chengdu, China, 2019.
27. Xu, X.; Yuan, H.; Mark, L.; Marcello, T. Two phase heuristic algorithm for the multiple-travelling salesman problem. *Soft Comput.* **2018**, *22*, 6567–6581. [[CrossRef](#)]
28. Liu, W.; Wang, Y. Path planning of multi-UAV cooperative search for multiple targets. *Electron. Opt. Control.* **2019**, *26*, 35–38.
29. Dong, X.; Cai, Y. A novel genetic algorithm for large scale colored balanced traveling salesman problem. *Future Gener. Comput. Syst.* **2019**, *95*, 727–742. [[CrossRef](#)]
30. Luo, Z.; Feng, S.; Liu, X.; Che, J.; Wang, R. Method of area coverage path planning of multi-unmanned cleaning vehicles based on step by step genetic algorithm. *J. Electr. Measur. Instr.* **2020**, *34*, 43–50.
31. Wang, M. Cooperative Task Assignment and Path Planning for Multi-Uavs Bird-Driving System at the Airport. Master's Thesis, Nanjing University of Aeronautics and Astronautics, Nanjing, China, 2019.
32. Lu, Z.; Zhang, K.; He, J.; Niu, Y. Applying K-means clustering and genetic algorithm for solving MTSP. Proceedings of Bio-inspired Computing—Theories and Applications-11th International Conference, BIC-TA 2016, Xi'an, China, 28–30 October 2016; pp. 278–284.
33. Kencana, E.N.; Harini, I.; Mayuliana, K. The performance of ant system in solving multi traveling salesmen problem. *Proc. Comput. Sci.* **2017**, *124*, 46–52. [[CrossRef](#)]
34. Necula, R.; Breaban, M.; Raschip, M. Tackling the bi-criteria facet of multiple traveling salesman problem with ant colony systems. In Proceedings of the IEEE 27th International Conference on Tools with Artificial Intelligence, Vietrisul Mare, Italy, 9–11 November 2015; pp. 873–880.
35. Changdar, C.; Pal, R.K.; Mahapatra, G.S. A genetic ant colony optimization based algorithm for solid multiple travelling salesmen problem in fuzzy rough environment. *Soft Comput.* **2017**, *21*, 4661–4675. [[CrossRef](#)]
36. Yang, S.Z.; Jia, B.C.; Yu, T. Research on Path Planning Algorithm of Multi-Arm Collaborative Picking for *Agaricus Bisporus* Intelligent Picking Robot. In Proceedings of the 5th IEEE/IFToMM International Conference on Reconfigurable Mechanisms and Robots, Toronto, ON, Canada, 12–14 August 2021; pp. 776–796.
37. Cao, X.; Yan, H.; Huang, Z.; Ai, S.; Xu, Y.; Fu, R.; Zou, X. A Multi-Objective Particle Swarm Optimization for Trajectory Planning of Fruit Picking Manipulator. *Agronomy* **2021**, *11*, 2286. [[CrossRef](#)]
38. Xu, H.; Hu, Z.; Zhang, P.; Gu, F.; Wu, F.; Song, W.; Wang, C. Optimization and Experiment of Straw Back-Throwing Device of No-Tillage Drill Using Multi-Objective QPSO Algorithm. *Agriculture* **2021**, *11*, 986. [[CrossRef](#)]
39. Saxena, A.; Prasad, M.; Gupta, A.; Bharill, N.; Patel, O.P.; Tiwari, A.; Er, M.J.; Ding, W.; Lin, C.T. A review of clustering techniques and developments. *Neurocomputing* **2017**, *267*, 664–681. [[CrossRef](#)]
40. Sisodia, D.; Sisodia, S.; Saxena, K. Clustering techniques: A brief survey of different clustering algorithms. *Int. J. Latest Trends Eng. Technol.* **2012**, *1*, 82–87.
41. Karna, A.; Gibert, K. Automatic identification of the number of clusters in hierarchical clustering. *Neural. Comput. Appl.* **2022**, *34*, 119–134. [[CrossRef](#)]
42. Boonchoo, T.; Ao, X.; Liu, Y.; Zhao, W.; He, Q. Grid-based dbscan: Indexing and inference. *Pattern Recognit.* **2019**, *90*, 271–284. [[CrossRef](#)]
43. Liu, F.; Wen, P.; Zhu, E. Efficient grid-based clustering algorithm with leaping search and merge neighbors method. *JOP Conf. Ser.* **2017**, *1*, 12122–12132. [[CrossRef](#)]
44. Melnykov, V. Challenges in model-based clustering. *WIREs Comput. Stat.* **2013**, *5*, 135–148. [[CrossRef](#)]

- 
45. Bechini, A.; Marcelloni, F.; Renda, A. Tsf-dbscan: A novel fuzzy density-based approach for clustering unbounded data streams. *IEEE Trans. Fuzzy Syst.* **2022**, *30*, 623–637. [[CrossRef](#)]
  46. Zhu, Y.; Ting, K.M.; Jin, Y.; Angelova, M. Hierarchical clustering that takes advantage of both density-peak and density-connectivity. *Inform. Syst.* **2022**, *103*, 101871. [[CrossRef](#)]