

## Research Article

# Picking Path Optimization of *Agaricus bisporus* Picking Robot

Xiaomei Hu , Zhaoren Pan , and Shunke Lv 

The Key Laboratory of Intelligent Manufacturing and Robotics, School of Mechatronic Engineering and Automation, Shanghai University, No. 99 Shangda Road, Shanghai 200444, China

Correspondence should be addressed to Xiaomei Hu; blossomhxm@shu.edu.cn

Received 20 May 2019; Accepted 4 August 2019; Published 25 September 2019

Academic Editor: Nunzio Salerno

Copyright © 2019 Xiaomei Hu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The design and application of the mushroom picking robot will greatly reduce the labor cost, and it has become one of the research hotspots in the world. Therefore, we independently developed an *A. bisporus* (a kind of mushroom) picking robot and introduced its functional principle in this paper. At the same time, in order to improve the picking efficiency of the picking robot, a picking path optimization algorithm based on TSP model is proposed. Firstly, based on the TSP model, a picking route model for *A. bisporus* was established to determine the storage location of each *A. bisporus*. Then, an improved simulated annealing (I-SA) search algorithm is proposed to find the optimal path sequence. By improving the path initialization module, path generation module, and temperature drop module, the I-SA search algorithm can optimize the picking path in a short time. Finally, in order to improve the stability and reduce the running time of the I-SA search algorithm, a parallel optimization method for global search (“rough exploration”) and local search (“precision exploration”) is proposed. Through simulation experiments, the I-SA search algorithm can search stable and excellent path solution in a relatively short time. Through field experiments on mushroom base, the efficiency of picking *A. bisporus* can be improved by 14% to 18%, which verifies the effectiveness of the I-SA search algorithm.

## 1. Functional Principle

Worldwide, some leading *A. bisporus* cultivation bases have already formed industrialized production for *A. bisporus* cultivation, and automation has been applied in the aspect of fertilization, feeding, and humidity control. However, the *A. bisporus* cultivation bases still rely on manual picking at present, which cannot satisfy the need of industrialized production. With the aging of the population and the increase of labor costs, it is not only difficult for *A. bisporus* cultivation bases to recruit new employees but also the labor costs are greatly increased. Therefore, the research of *A. bisporus* picking robot has become one of the research hotspots worldwide.

Figure 1 is a 3D model of self-designed *A. bisporus* picking robot. In order to be able to run between the layers, the height of robot does not exceed 290 mm. The robot is mainly divided into visual area, picking area, and auxiliary area. Its main functions are as follows.

**1.1. Visual Area.** The size of the visual area is 1000 mm \* 150 mm, which is also the size of its coordinate system. The

visual area needs to perform the following tasks: first, a movable camera moves to each workstation for photo shooting. It is necessary to splice these taken photos together to obtain a complete and accurate visual area coordinate system. Then, image processing is performed to obtain coordinate points of each *A. bisporus* in the visual area coordinate system. Finally, each coordinate point is placed in the array F in order. However, in order to complete the appeal task, two major problems need to be solved. They are the effective area of the single photo and the way of “image stitching”.

The effective area of the single photo: in order to make the industrial camera get a larger field of view, this project chose a lens with a smaller focal length. Although the field of view can reach 240 mm \* 190 mm, the smaller the focal length, the larger the distortion. Even through image processing, distortion cannot be completely corrected. At the same time, as shown in Figure 2(b), *A. bisporus* at the edge of the photo is usually incomplete, which is the main reason. Due to two reasons for the appeal, the image processing could not correctly determine the actual central coordinate point (picking point) of *A. bisporus*, so it is necessary to

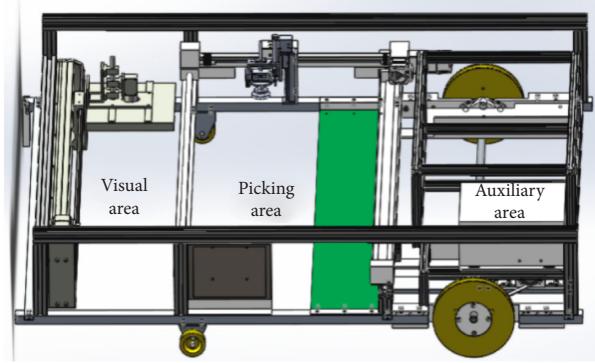


FIGURE 1: The 3D model of the *A. bisporus* picking robot.

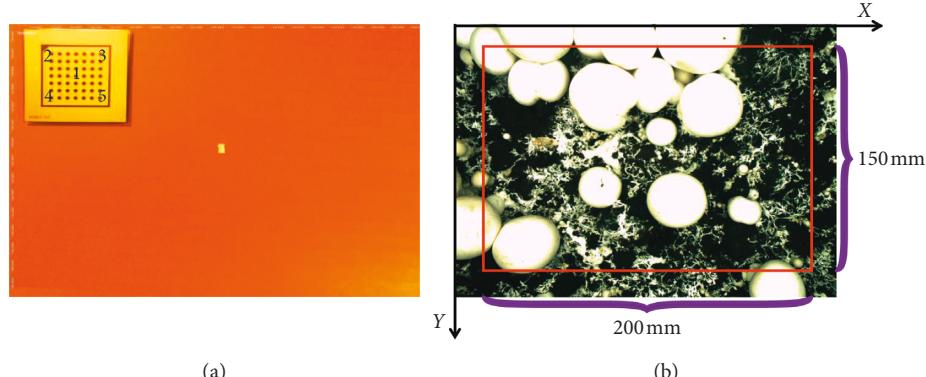


FIGURE 2: Image information. (a) Distortion test. (b) Effective area.

determine the effective area to eliminate the picking point of *A. bisporus* at the edge. The average diameter of the mature *A. bisporus* is about 40 mm, so the overlap of the two photos should be 40 mm to ensure that more *A. bisporus* can be completely photographed in one photo. At the same time, the image was calibrated using Halcon software, which obtained the camera internal parameters, camera external parameters, and the conversion value of pixel to millimeter ( $C_{pm} = 0.000119$ ). The distortion of the image is corrected by the camera's internal and external parameters. Then, the corrected distortion image is tested as shown in Figure 2(a). The four corners in the photo are tested using the calibration plate. The distance from 1 point to the other 4 points is calculated, and 5 sets of experiments are performed for each angle. Point distance (mm) is the pixel distance multiplied by  $C_{pm}$ . The true distance is 16.97 mm, and the results are shown in Table 1.

From the data in Table 1, the correction of distortion is quite successful, so the main consideration is the incompleteness of *A. bisporus*. Therefore, the width and height of the image are subtracted by 40 mm. At the same time, it also reduces the error caused by image distortion. As shown in Figure 2(b), the mushroom coordinate point is the effective picking point inside in the red border.

The way of “image stitching”: according to the effective area of the single photo and the area of the visual area, the camera has a total of 5 workstations to be reached and the

interval between each workstation should be 200 mm. Then, the establishment of the visual area coordinate system needs to be spliced by 5 photos. In the traditional way, feature point matching algorithm of image processing is used to make the photos spliced together [1]. This method is not only time consuming but also fails because the found feature points are too similar to match. To this end, this paper proposes a simple way to “splicing” pictures. First define the coordinate system of the single photo: the origin of the coordinate system of a single photo (240 mm \* 190 mm) is the upper left corner, the X-axis direction from left to right, and the Y-axis direction from top to bottom, as shown in Figure 2(b). The process of image processing is proposed by Wang scholar [2]. Through image processing, each *A. bisporus* in the photo has its corresponding coordinate point ( $X_m, Y_m$ ), which has been converted from pixels to millimetres by  $C_{pm}$ . Next,  $P_i$  is defined as the number of the current workstation, and the value of the number is from 0 to 4. Then, the position of *A. bisporus* in the visual area coordinate system ( $X_{pm}, Y_{pm}$ ) can be determined by the coordinate points in the photo and workstation, as follows:

$$X_{pm} = X_m + 200 * P_i, \quad Y_{pm} = Y_m. \quad (1)$$

From the appeal formula, the origin of the visual area coordinate system is the origin of the first photo coordinate system. Finally, the converted *A. bisporus* coordinate points are saved to the array F. The order of saving to the array F is

TABLE 1: Calibration test.

	Upper left corner	Lower left corner	Upper right corner	Lower right corner
Average value	16.37 mm	17.08 mm	16.44 mm	17.12 mm
Std. dev.	0.62	0.75	0.75	0.66

$Y_m$  value from small to large for a single photo and  $P_i$  value from small to large for the workstation. This save order is the picking order before the path optimization is performed.

In this paper, the workstations are used to “splicing” the pictures together, so the accuracy of the camera’s moving station is very high. For this purpose, a stepper motor with an encoder and a screw module is used to ensure that the movement error accuracy is within 0.5 mm.

**1.2. Picking Area.** The picking area needs to perform the following tasks: firstly, the picking point of each *A. bisporus* is extracted in order from the array  $F$ . Then, the picking mechanism moves to the picking point to pick the *A. bisporus*. When the picking is completed, *A. bisporus* is placed in the green area (storage area) shown in Figure 1. The storage area transports the *A. bisporus* through the track. Finally, when picking up all the *A. bisporus* in this area, the picking mechanism returns to the origin of the picking coordinate system.

The picking mechanism moves in the range of 1000 mm \* 240 mm. At the same time, it should be noted that the coordinate system of the visual area and the coordinate system of the picking area must overlap, so that the picking mechanism can correctly pick *A. bisporus* according to the coordinate point. The picking mechanism picks *A. bisporus* by means of pressing, rotating, and sucking. The purpose of pressing the *A. bisporus* is to prevent the stem from breaking off when the mushroom is rotated. The purpose of the rotation is to break the root of the *A. bisporus* from the rhizome, so that they can be easily sucked up. Finally, *A. bisporus* was picked by proper suction. Some *A. bisporus* are not vertical but are inclined. For this reason, the picking mechanism uses an organ suction cup, which can be adaptively attached to the mushroom surface. At the same time, improper suction or rotation may damage the umbrella surface of *A. bisporus*. For this reason, a sponge is added to the head of the suction cup as shown in Figure 3(a). In addition, in order to improve the picking efficiency, the air pressure sensor is used to detect whether or not the mushroom is sucked, as shown in Figure 3(b). If the mushroom is not sucked, the picking mechanism will move directly to the next picking point instead of the storage area.

**1.3. Auxiliary Area.** This area is used to store various electronic components, such as sensors, industrial computers, vacuum pumps, and electrical cabinets. At the same time, the driving wheel of the picking robot is also installed here.

In order to speed up the picking efficiency of the *A. bisporus* picking robot, it is one of the best ways to optimize the picking path. Its advantage is that it can improve the picking efficiency without additional requirements on the

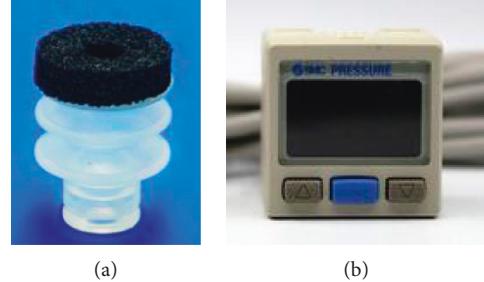


FIGURE 3: Components. (a) Organ suction cup. (b) Air pressure sensor.

picking precision and burdens on the mechanical structure. For this reason, the path optimization module needs to be put into the whole control process of *A. bisporus* picking robot. Its overall control flow and image processing flow are described in detail in the literature [2, 3]. In this paper, the process is simplified, as shown in Figure 4.

The brief process is as follows: firstly, after the shooting and image processing, the array  $F$  will obtain the coordinates of the *A. bisporus*; secondly, array  $F$  is passed to the path optimization module. After path optimization, array  $F$  is transformed into array  $S$ . In the array  $S$ , not only the picking path is optimized but also the storage location of each *A. bisporus* is added. Thirdly, the array  $S$  is passed to the picking area for saving. Finally, the corresponding *A. bisporus* is picked by the picking mechanism the next time. At the same time, the visual area and the picking area can work at the same time by parallel control so as to improve the work efficiency.

## 2. Introduction

The picking path of *A. bisporus* can be optimized by the searching algorithm. Search algorithms are applied to various occasions [4–7], and one of them is applied to TSP (traveling salesman problem) model [8]. The TSP model is a typical combinatorial optimization problem, which can be expressed as a person needs to go through several cities, he needs to start from a city (the city can be fixed or not fixed), go through all cities and only once, and finally return to the starting position. The picking path of our picking robot is very similar to the TSP model, so the model in this paper will be improved based on the TSP model.

The most classical search algorithms based on the TSP model can be generally divided into two categories: the first category is the heuristic search algorithm, which mainly includes linear programming, branch and bound [9], dynamic programming [10], and so on. The second category is the intelligent optimization algorithm, which is the main research direction of scholars at present. The typical ones are

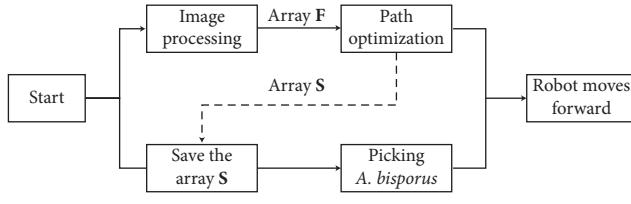


FIGURE 4: Control flow diagram.

ant system (AS), simulated annealing algorithm (SA), genetic algorithm (GA), particle swarm optimization (PSO), artificial neural networks (ANNs), and so on. The purpose of these algorithms is to obtain better solutions rather than exact solutions in a relatively short period of time. Among them, the typical ACO and PSO are faced with the problem of falling into local optimal or its convergence precision is not high. For this reason, scholars also optimize and improve these search algorithms. The novel quantum ant colony algorithm [11] improves the population information structure through the quantum evolution algorithm (QEA), which improves the convergence speed and global search ability. The max-min ant system (MMAS) [12] improves the pheromone trail updating method on the basis of AS and limits the value of pheromone trail to avoid stagnation of the search. Through the above methods, MMSA can find excellent path solutions.

Among them, simulated annealing also is one of the most popular algorithms because of its ability to jump out of the poor local optimum. But its disadvantage is that the parameters such as annealing temperature and initial temperature have great influence on the optimization results. Moreover, Metropolis criterion has a certain probability of accepting poor solutions at high temperature, which may lead to the loss of better solutions in the process of solving. For this reason, many scholars propose improved schemes for SA and introduced them into various projects [13–16]. The improvement of each module of SA is as follows.

For the path initialization module, Wei et al. initializes the path through the improved particle swarm optimization algorithm to increase the global search ability of SA [17]. Yao et al. uses the improved genetic algorithm to obtain a better initial path order, so as to reduce the possibility of SA falling into a poor local optimal path [18].

For the path generation module, García [19] uses SA as an acceptance device in GA. Through the acceptance device, the inferior parent has a chance not to be replaced by the superior offspring after cross mutation. At the same time,  $X^C$  is introduced. If the solution of the offspring is better than  $X^C$ , the old  $X^C$  is replaced and  $X^C$  is regarded as the father of the next mating. In this way, this search algorithm can finally get a better solution. Yao et al. [18] mainly uses two algorithms in path generation. The first algorithm divides the whole path into three regions randomly and then inverts three regions to form new three regions. Finally, the six path regions are recombined, and the shortest path solution is selected as the newly generated path. The second algorithm uses DFS to find the optimal path for a region of the entire path. The commonly used path generation algorithm is to invert a section of the entire path to generate a new path [20].

For the temperature drop module, many scholars studied the way of temperature drop [21, 22]. Many scholars [23, 24] combine SA with tabu search algorithm and propose an adaptive temperature drop module to prevent the algorithm from rapidly converging to a poor local optimal solution. Among them, Ingber proposes a very fast simulated annealing algorithm (VFSA) [25], which accelerates the optimization process, and at the same time, VFSA became a typical SA search algorithm used in practice [26–28]. The VFSA has also been improved by scholars; Chen et al. scholar modifies the disturbance model of VFSA to reduce the disturbance at low temperature, so as to improve the accuracy of the model [29].

In order to improve the efficiency and stability of the search algorithm, parallel search algorithms are proposed and applied in various scenarios [13, 30, 31], and remarkable results are obtained at the same time. Gehring and Bortfeldt propose an improved parallel genetic algorithm (PGA) [32] to solve the container loading problem. When the crossover and mutation of each subpopulation are completed,  $N$  individuals that are excellent themselves will be moved to another subpopulation and new  $N$  individuals will be accepted from other subpopulations, so as to keep the number of each subpopulation unchanged, as shown in Figure 5.

Gelfand firstly reduces the number of iterations of simulated annealing to more than half and carries out multiple SA [33]. The new SA regards the previous SA solution as the initial solution and resets the annealing parameters. If this solution is better than the previous solution, the next SA will be carried out. If this solution is not better than the previous solution, this thread will stop running. Finally, when all threads stop, the best solution in all threads is selected as the final solution, as shown in Figure 6.

According to the analysis of existing search algorithms, the SA algorithm can be applied for picking path optimization of *A. bisporus*. In order to solve the problem that the traditional SA converges slowly and takes a long time, an improved simulated annealing (I-SA) search algorithm is proposed based on traditional SA: add memory function to I-SA search algorithm, so that it can memorize the optimal path sequence of the current program; a new path initialization assignment method is proposed in the path initialization module; two path generation algorithms are proposed in the path generation module, which are used for the path generation algorithm of “rough exploration” and “precision exploration”; based on the VFSA temperature drop method, the T-VFSA temperature drop method is proposed, which makes the search algorithm to jump out of the poor local solution. A new parallel optimization method is proposed to improve the stability and efficiency of the I-SA search algorithm.

### 3. Picking Path Model for *A. bisporus*

In the traditional TSP model, the position of each coordinate point is known. When calculating the total length of the path, it is only necessary to calculate the distance between the two positions in sequence and add them in turn.

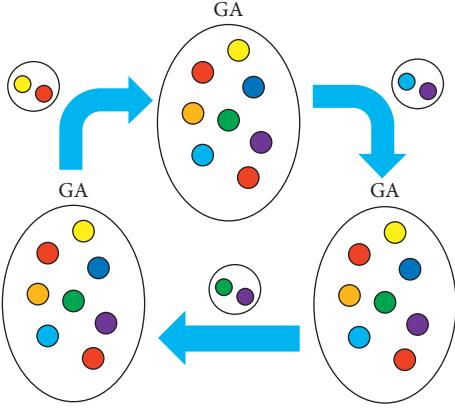


FIGURE 5: Multiple EAs running in parallel.

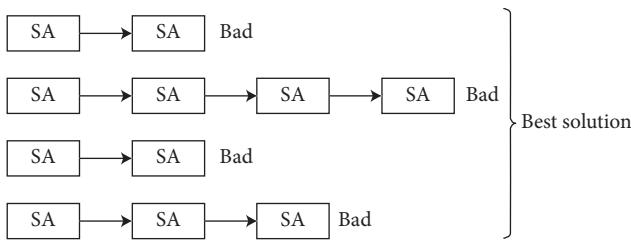


FIGURE 6: Multiple SAs running in parallel.

However, this paper needs to solve the problem of unknown storage positions, so not all positions are known. Therefore, this chapter builds a picking route model based on the TSP model.

**3.1. Model Assumption.** The picking model of *A. bisporus* is established based on the following hypotheses:

- (1) It is known that  $n$  *A. bisporus* need to be caught (excluding the starting position)
- (2) The picking mechanism starts from the starting position and finally returns to the starting position
- (3) Each *A. bisporus* can be picked by the picking mechanism and put into storage area
- (4) The storage area is defined as a line segment
- (5) If the optimal storage position of an *A. bisporus* is not within the storage area, the end of the storage area near the optimal storage position is used as the storage position

**3.2. Determination of Storage Positions.** As shown in Figure 7,  $A$  and  $B$  are successively picked *A. bisporus*, whose coordinate points are  $(X_1, Y_1)$  and  $(X_2, Y_2)$ ;  $C$  represents the storage position of  $A$ , whose coordinate point is  $(X_{s1}, Y_{s1})$ ; the line segment of the storage area is  $y = a_s x + b_s$ ; the position  $C$  can be determined by position  $A$  and position  $B$ . The steps are as follows:

- (1) Judge whether positions  $A$  and  $B$  are on the same side of the line segment of the storage area. If they are on the same side, we need to find the symmetry position

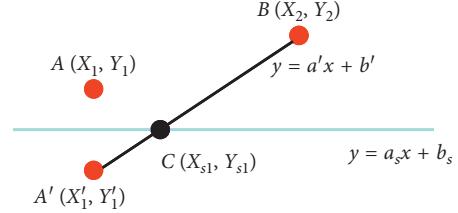


FIGURE 7: Optimal storage position.

$A'(X'_1, Y'_1)$ , as shown in Figure 7. The coordinate point of  $A'$  is obtained by formulas (4) and (5). If positions  $A$  and  $B$  are on the different side, the  $A$  coordinate point is passed directly to the  $A'$  coordinate point, as shown in formula (6):

$$\frac{Y_1 + Y'_1}{2} = a_s \left( \frac{X_1 + X'_1}{2} \right) + b_s, \quad (2)$$

$$\frac{Y_1 + Y'_1}{X_1 - X'_1} = -\frac{1}{a_s}, \quad (3)$$

$$X'_1 = \frac{X_1 - a_s^2 - 2a_s b_s + 2a_s Y_1}{a_s^2 + 1}, \quad (4)$$

$$Y'_1 = \frac{2a_s X_1 + 2b_s + a_s^2 Y_1 - Y_1}{a_s^2 + 1}, \quad (5)$$

$$\begin{aligned} X'_1 &= X_1, \\ Y'_1 &= Y_1. \end{aligned} \quad (6)$$

- (2) Now,  $A'$  and  $B$  are not on the same side. The intersection position between line  $A'B(y = a'x + b')$  and line segment of storage area ( $y = a_s x + b_s$ ) is position  $C(X_{s1}, Y_{s1})$ . When  $X_2 = X'_1$ , position  $C$  is shown in formula (7). When  $X_2 \neq X'_1$ , position  $C$  is shown in formula (8):

$$X_{s1} = X_2, Y_{s1} = a_s X_2 + b_s, \quad (7)$$

$$X_{s1} = -\frac{b_s - b'}{a_s - a'}, Y_{s1} = -a_s \left( \frac{b_s - b'}{a_s - a'} \right) + b_s. \quad (8)$$

where  $a' = (Y_2 - Y'_1)/(X_2 - X'_1)$  and  $b' = Y_2 - a' X_2$ .

- (3) Through the appeal procedure, the position of the optimal storage position  $C$  is calculated. The old picking route is shown in Figure 8(a). Now, the length of improved picking route is shorter than the length of old route, as shown in Figure 8(b).

**3.3. Picking Path Model Establishment.** In the establishment of the picking path model, the following parameters need to be defined.

$N = \{1, 2, 3, \dots, n\}$  indicates the set of *A. bisporus* nodes. The picking path sequence of *A. bisporus* can be defined as

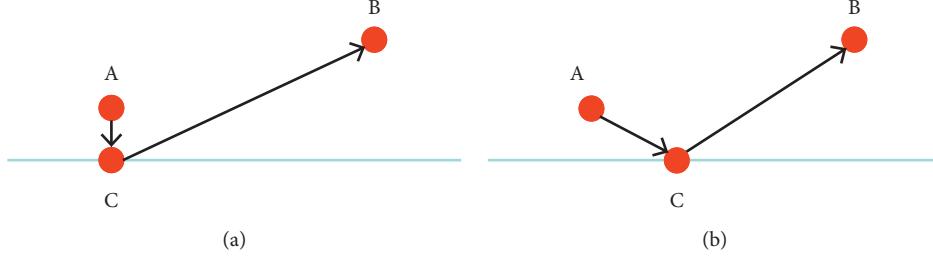


FIGURE 8: Picking and storage route. (a) Old. (b) New.

$D = \{D_1, D_2, \dots, D_n; D_m \in N\}$ , where  $D_m$  represents the *A. bisporus* node of  $m$ -th position. Each *A. bisporus* has its corresponding storage, which can be defined as  $D_s = \{D_{s1}, D_{s2}, \dots, D_{sn}; D_{sm} \in N\}$ , and the starting node is defined as  $D_0$ .

$E = D_0 \cup N$  indicates that  $E$  contains the set of the starting node and all nodes of *A. bisporus*.

$d(D_m, D_{sm})$  represents the distance from the *A. bisporus* node of  $m$ -th position to its storage.

When the  $i$  and  $j$  *A. bisporus* nodes are picked successively ( $j \neq i, i, j \in E$ ),  $y_{ij}$  is set to 1; otherwise,  $y_{ij}$  is set to 0.

When the picking mechanism has pick the  $j$  *A. bisporus*,  $Z_j$  is set to 1. If this *A. bisporus* has not been picked,  $Z_j$  is set to 0.

The following mathematical model can be established through the above model parameters:

$$\begin{aligned} \text{minimize: } T_d &= d(D_0, D_1) + d(D_n, D_{sn}) + d(D_{sn}, D_0) \\ &+ \sum_{m=1}^{n-1} [d(D_m, D_{sm}) + d(D_{si}, D_{m+1})], \end{aligned} \quad (9)$$

$$\begin{aligned} \text{subject to: } \sum_{i \in E} y_{ij} &= 1, \quad \forall j \in E \& i \neq j, \\ \sum_{j \in N} Z_j &= n, \\ y_{ij}, Z_j &= \{0, 1\}. \end{aligned} \quad (10)$$

Formula (9) is used to calculate the length of the path according to the picking sequence of *A. bisporus*, and the calculation result  $T_d$  is called path solution. The goal of this formula is to make the path solution minimum by the searching algorithm.

#### 4. Optimizing the Picking Path of *A. bisporus* Based on I-SA

I-SA is proposed to optimize the picking path of *A. bisporus*. Compared with the traditional SA, I-SA improves the path initialization module, path generation module, and temperature drop module in order to search a better path sequence in a relatively short time and to improve the working efficiency of the *A. bisporus* picking robot. Each improved module will be grouped together in Section 4.4 to become a complete I-SA (Figure 9).

**4.1. Path Initialization Modules.** This module corresponds to **Initialize** ( $W_c$  and  $W_{\min}$  form Array F) in Figure 9.

```

Input: Array F
Output:  $W_{\min}$ 
Initialize ( $T_0, T_{\max}, \alpha, K, \beta$ );
Initialize ( $W_c, W_{\min}$  from Array F);
 $T = T_{\max}$ ;
 $\Delta f = |f(W_c) - f(W_{\min})|$ ;
While ( $T < T_0$ )
    For ( $k = 0; k < K; k++$ )
        Generate ( $W'$  from  $W_c$  and  $W_{\min}$ );
         $\Delta f = f(W_c) - f(W')$ ;
        If ( $\Delta f > 0$ )
             $W_c \leftarrow W'$ ;
            If ( $f(W') < f(W_{\min})$ )
                 $W_{\min} \leftarrow W'$ ;
        Else if ( $\exp \{\Delta f/T\} > \text{random} ([0, 1])$ )
             $W_c \leftarrow W'$ ;
    End
    Generate ( $T$  from  $\alpha$  and  $T_0$ );
End

```

FIGURE 9: I-SA pseudocode.

Two parameters,  $W_{\min}$  and  $W_c$  are introduced in the path initialization module.  $W_{\min}$  represents the global-best path sequence that I-SA search algorithm has searched so far. Through  $W_{\min}$  parameter, the I-SA search algorithm has the memory function to ensure that the search algorithm does not lose the current global-best path sequence. And,  $W_{\min}$  will be used for “rough exploration” path generation algorithm in the path generation module which will be introduced in Section 4.2.  $W_c$  represents the initial path sequence which is assigned by array F. At the same time,  $W_c$  is also the path sequence selected by the Metropolis criterion in the next iteration which will be introduced in Section 4.4.

The Metropolis criterion in SA makes the search algorithm accept the bad path sequence with a certain probability at high temperature and thus may lose the original better path sequence. In order to solve the problem

that initialization may have little effect on the final path solution, a new path initialization assignment method is proposed. Firstly, MMSA is used in the path initialization in order to get a good initial path sequence and its main formula is as follows. In formula (11),  $p_{ij}^{k_A}(t)$  represents the probability that ant  $k_A$  chooses from  $i$  node to  $j$  node;  $\tau_{ij}(t)$  represents the amount of pheromone between  $i$  node and  $j$  node (pheromone trail);  $\eta_{ij}$  represents the reciprocal of the distance from  $i$  node to  $j$  node.  $N_i^{k_A}$  represents the set of cities which ant  $k_A$  has not visited yet.

$$p_{ij}^{k_A}(t) = \frac{[\tau_{ij}(t)]^{\alpha_A} [\eta_{ij}]^{\beta_A}}{\sum_{l \in N_i^{k_A}} [\tau_{il}(t)]^{\alpha_A} [\eta_{il}]^{\beta_A}}, \quad \text{if } j \in N_i^{k_A}, \quad (11)$$

$$\tau_{ij}(t+1) = \rho \tau_{ij}(t) + \Delta \tau_{ij}^{\text{best}}, \quad (12)$$

$$\Delta \tau_{ij}^{\text{best}} = \frac{1}{f(s^{\text{best}})}. \quad (13)$$

In formula (12),  $\rho$  represents the evaporation rate. The evaporation mechanism helps to avoid the unlimited accumulation of the pheromone trails. In formula (13),  $f(s^{\text{best}})$  represents path solution of either the iteration-best ant or the global-best ant. In addition,  $M_A$  represents the number of ants in the path search and  $K_A$  represents the number of iterations.

Then, unlike the traditional path initialization assignment method, the path sequence obtained by the MMSA is assigned to  $W_{\min}$  instead of  $W_c$ . In this way,  $W_{\min}$  can guide the  $W_c$  to a certain extent and it does not lose the better path sequence at high temperature. Meanwhile, the  $W_c$  path sequence is randomly disrupted in the initialization module. The pseudocode is shown in Figure 10.

**4.2. Path Generation Module.** This module corresponds to **Generate** ( $W'$  from  $W_c$  and  $W_{\min}$ ) in Figure 9. The following parameters are introduced in the path generation module.

$f(W)$  represents the length of the  $W$  path sequence, which is called  $W$  path solution.

$W'$  represents the new path sequence obtained by the path generation algorithm,  $W'$  path solution will be compared with  $W_c$  path solution, and then the Metropolis criterion will be used to determine whether  $W'$  can replace the  $W_c$ ;

$\Delta f$  represents the difference between the  $W_c$  path solution and the  $W'$  path solution, and the formula is expressed as  $\Delta f = f(W_c) - f(W')$ .

In the path generation module, the first step is initialization.  $i$  and  $j$  nodes are selected randomly in the path, and they need to satisfy  $0 < i < j \leq n$  ( $n$  is the number of *A. bisporus*).

Next, two ways of path generation algorithm, “rough exploration” and “precision exploration”, are designed in order to improve the search quality of path solution. At high temperature, more use of “rough exploration” path generation algorithm can improve the global search ability of the search algorithm, and at low temperature, more use of

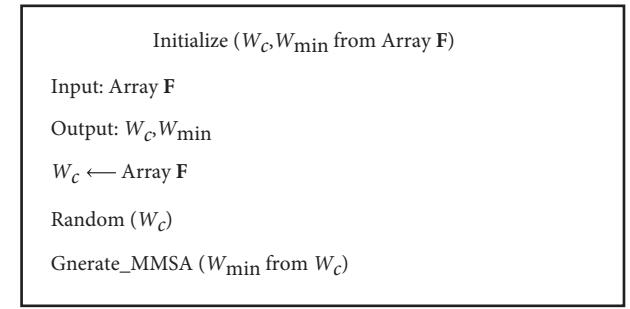


FIGURE 10: Pseudocode for path initialization module in I-SA.

“precision exploration” path generation algorithm can improve the local search ability of the search algorithm. To meet the above requirements, an I-Metropolis formula is defined as follows:

$$\tau = e^{-(|\Delta f|/T)} - \beta. \quad (14)$$

According to formula (14), the introduction of  $|\Delta f|$  makes the  $\tau$  range between [0, 1]. The  $\tau$  value will be compared with random numbers between 0 and 1. When the  $\tau$  value is greater than the random number, the “rough exploration” path generation algorithm is used; otherwise, the “precision exploration” path generation algorithm is used. The  $T$  parameter indicates the current temperature. In addition,  $\beta$  parameter guarantees that the “rough exploration” path generation algorithm is no longer used or less used at low temperature. The experimental results show that, when the  $\beta$  parameter is between 0.1 and 0.2, the search effect is the best.

Two different path generation algorithms are described as follows.

**4.2.1. “Rough Exploration” Path Generation Algorithm.** It greatly disrupts  $W_c$  path sequence, the purpose of which is to enable the search algorithm to explore more path sequence and select better path sequence for the next “precision exploration.” However, in the experiment, it is found that only random disruption of the path cannot find a better path sequence. This way of “rough exploration” ability is poor and inefficient. For this reason, a “rough exploration” path generation algorithm based on the improved regional crossover algorithm (IRCA) is proposed. The region of  $W_{\min}$  path sequence in randomly disrupted path sequence is retained, so as to explore whether there is a more appropriate optimal path. The specific steps are as follows:

- (1) Random disruption of the current  $W_c$  path sequence
- (2) Judge whether the disrupted  $W_c$  path sequence is the same as the current  $W_{\min}$  path sequence, jump to step 1 if the path sequence is the same, and jump to step 3 if it is not the same
- (3) The  $i$ -to- $j$  region of  $W_{\min}$  is assigned to the  $i$ -to- $j$  region of the  $W_c$
- (4) As shown in Figure 11(c), some nodes appear to be duplicated in the path sequence after the region is

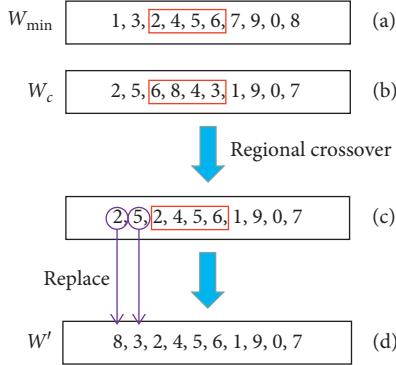


FIGURE 11: The improved regional crossover algorithm.

- assigned; the duplicate nodes are replaced by the removed nodes in the regional crossover process
- (5) If there are duplicate nodes in the path sequence, jump to step 4; otherwise, generate a new path  $W'$

The improved regional crossover algorithm is shown in Figure 7.

**4.2.2. “Precision Exploration” Path Generation Algorithm.** It makes small changes to the path sequence of a section in the  $W_c$ . Its purpose is to explore the path sequence found by “rough exploration” and to explore whether better path solutions can be obtained. When the “precision exploration” path generation algorithm contains multiple path generation algorithms, it can not only increase the probability that the search algorithm can find a better path solution but also can find a better path sequence for different data samples. Therefore, the “precision exploration” path generation algorithm is composed of four methods: region greedy method; two-node swapping method; region rotation method; and region inversion method.

The region greedy (RG) method performs a greedy algorithm in the region from the  $i$  node to the  $j$  node. The  $i$  node is used as a starting node to find its nearest coordinate point in the region as the next node and sequentially searches for the closest coordinate point to the current node as the next node.

The 3-OPT method is to exchange three points ( $i$  node to  $i+2$ -th node) for each other and select the smallest path solution as a result.

The region rotation (RR) method is used in the path region from the  $i$  node to the  $j$  node. In this region, the  $j$  node is inserted before the  $i$  node, which is shown in Figure 12.

The region inversion (RI) method exchanges nodes in the path region from the  $i$  node to the  $j$  node. The  $i$  and  $j$  nodes are interchanged, and then the  $(i+1)$ -th and  $(j-1)$ -th are interchanged and sequentially executed. The RI is shown in Figure 13.

The pseudocode of the path generation module is shown in Figure 14. Each iteration in the I-SA search algorithm generates a new  $W'$  in the path generation module. New  $W_{\min}$ ,  $W_c$ , and  $\Delta f$  can be obtained by  $W'$ . These parameters in **Input** come from the result of the last iteration.



FIGURE 12: Region rotation algorithm.



FIGURE 13: Region inversion algorithm.

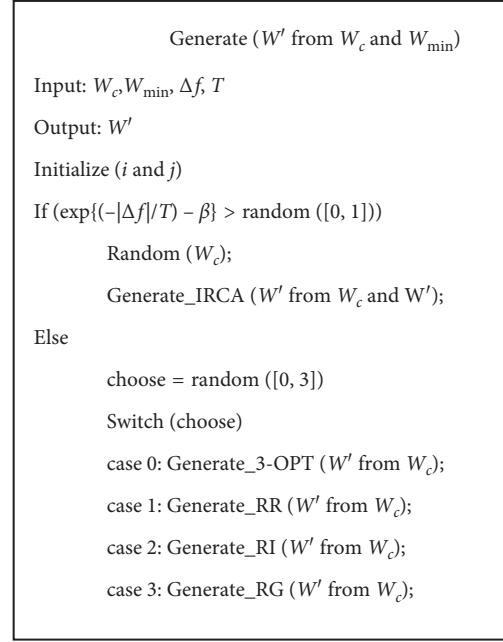


FIGURE 14: Path generated pseudocode.

I-Metropolis criterion is used to choose whether to use “rough exploration” or “precision exploration” path generation algorithms. The path generation algorithms are used to obtain a new path sequence ( $W'$ ). Among them, IRCA is used more frequently in “rough exploration”, and 3-OPT, RR, RI, and RG methods are used more frequently in “precision exploration.”

**4.3. Temperature Drop Module.** This module corresponds to **Generate** ( $T$  from  $\alpha$  and  $T_0$ ) in Figure 9.

VFSA as a typical SA search algorithm is widely used in industry, military, scheduling, and so on. However, the VFSA temperature drop method is not used in the TSP model at present. In this paper, the VFSA temperature drop method is introduced into the model and improved.

The VFSA temperature drop method is as shown in formula (15).  $T$  and  $T_0$  are the current temperature and initial temperature;  $\alpha$  is the temperature drop coefficient;  $p$  is the current number of temperature drops. When the temperature goes down once,  $p$  goes up by 1 and  $U$  is the number of parameters:

$$T = T_0 \exp(-\alpha p^{1/U}). \quad (15)$$

The VFSA temperature drop method can find the optimal path solution after fewer iterations. However,

because the VFSA temperature drop method drops too fast at high temperature, the global search ability is not good when there are more data (more than 80 *A. bisporus* coordinate points), which may lead to search algorithm falling into poor local optimum. Therefore, this paper improves the VFSA temperature drop method by increasing its tempering function, which is called tempering VFSA (T-VFSA). When the temperature reaches the low temperature, it returns to the initial temperature  $T_{\max}$ . By means of tempering, the “rough exploration” is carried out again so as to jump out of the bad local optimal path solution with a certain probability. The curve of Figure 12 shows the temperature corresponding to the number of iterations. Figure 15(a) shows the temperature drop curve of VFSA, and Figure 15(b) shows the temperature drop curve of T-VFSA. It can be seen from the comparison that T-VFSA increases the number of iterations at high temperature to improve “rough exploration” ability.

**4.4. The Process of I-SA.**  $K$  parameters are introduced in the process of the I-SA search algorithm, which represents the number of iterations when the parameter  $T$  reaches each temperature value. The pseudocode of I-SA search algorithm is shown in Figure 9. The specific steps are as follows:

- (1) Firstly, the parameters ( $T_0$ ,  $T_{\max}$ ,  $\alpha$ ,  $K$ , and  $\beta$ ) are assigned and the MMSA is used to initialize  $W_{\min}$ , so that it can obtain better path sequence in the initial stage.  $\Delta f = |f(W_c) - f(W_{\min})|$ , which is used in the path generation module for the first time.
- (2) In order to ensure that each temperature node can perform enough exploration to obtain a better path solution, the  $K$  parameter is required. When the  $K$  value cycle is completed, it drops to the next temperature node.
- (3) In the path generation module (Generate ( $W'$  from  $W_c$  and  $W_{\min}$ )), select the path generation algorithm of “rough exploration” or “precision exploration” through the I-Metropolis criterion to generate the new path sequence ( $W'$ ).
- (4) If  $W'$  path solution is better than  $W_c$  path solution,  $W_c$  will be replaced. If  $W'$  path solution is not better than  $W_c$  path solution, the Metropolis criterion will be used to judge whether  $W'$  is to replace  $W_c$ . At the same time, if  $W'$  path solution is better than  $W_{\min}$  path solution,  $W'$  will become the new  $W_{\min}$ .
- (5) When the current temperature has been iterated for  $K$  times, the temperature shall be decreased. The way to drop the temperature is to use the T-VFSA method, when the temperature reaches  $10^\circ$  and return to  $T_{\max}$  temperature.
- (6) Finally, when the generated temperature  $T$  is less than  $T_0$ , jump out of the loop and output the  $W_{\min}$  path sequence as the final path solution of the search algorithm.

## 5. Parallel Optimization Method for Picking Path of *A. bisporus*

Because the multireading parallel method can improve the stability and efficiency of search algorithms [31, 34–40], the parallel optimization method for picking path of *A. bisporus* is proposed. The parallel optimization method at high temperature runs multiple I-SAs at the same time. First, each I-SA has the same initial parameters. Then, starting from the path initialization, each SA performs its own operation, the purpose of which is to ensure that the poor I-SA does not affect other I-SAs. Finally, when the low temperature is reached, the path solutions of I-SA are compared, and then the best of these solutions is used as the path sequence in “precision exploration”. In this way, the stability of the I-SA search algorithm is improved.

In the low temperature, the “precision exploration” parallel optimization method should search for the local optimal path solution in a relatively short time. Therefore, parallel operations should be performed within I-SA, rather than parallel operations as “rough exploration”. In the I-SA search algorithm, the parallel optimization method is to run multiple path generation modules at the same time, and the best path solution is selected as the new generation path solution. In this way, the number of iterations can be reduced and the efficiency of the I-SA search algorithm is improved.

In this paper, the parallel optimization method of “rough exploration” and “precision exploration” not only improves the stability of the search algorithm but also shortens the running time and improves the efficiency. The parallel optimization method is shown in Figure 16.

## 6. Experimental Tests

The experiment is divided into two parts. The first part is the comparative test of algorithm performance. First, the path generation algorithm in this paper is compared with other path generation algorithms, which proves that the path generation algorithm in this paper can provide better path solutions. Then three methods of temperature drop are compared, which are Markov chain temperature drop mode, VFSA temperature drop mode, and T-VFSA temperature drop mode in this paper, which proves that T-VFSA mode can improve the efficiency of the search algorithm without affecting the accuracy and stability of the search algorithm. Finally, the I-SA search algorithm is compared with other search algorithms through TSP experimental samples, which proves that the search algorithm can search for better path solutions in a shorter time.

The second part is the practical application of the I-SA search algorithm. Firstly, the coordinate points of *A. bisporus* are extracted through image processing. And, the experiment is carried out with these data of coordinate points to test whether the I-SA search algorithm can find better path solutions. Then, the I-SA search algorithm is applied to the *A. bisporus* picking robot to test how much improvement of picking efficiency can be achieved through path optimization.

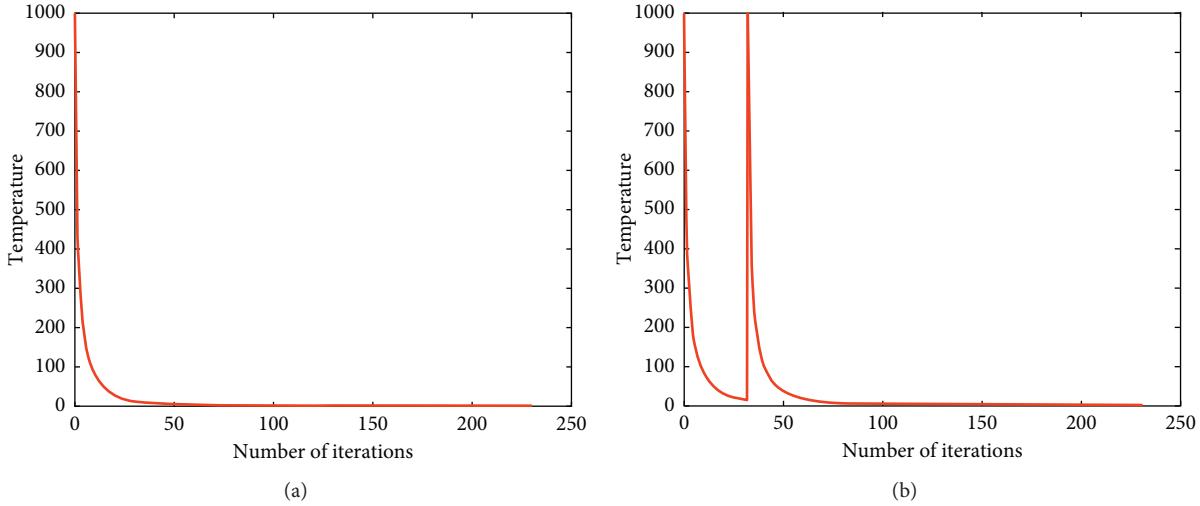


FIGURE 15: Comparison of VFSA and T-VFSA temperature curve.

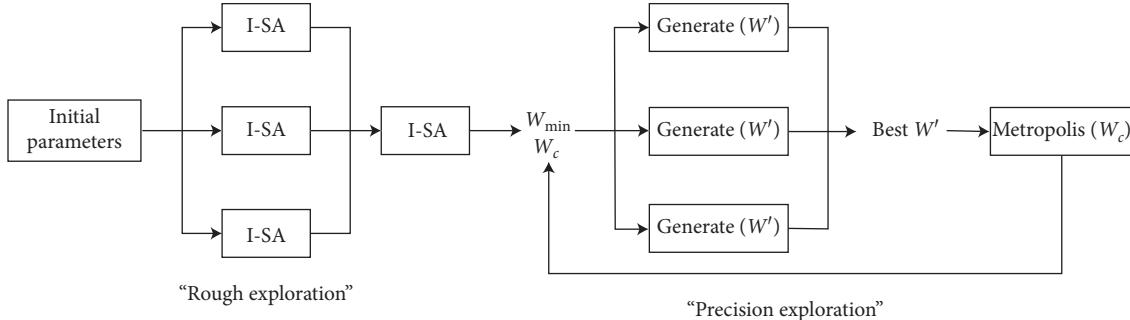


FIGURE 16: Parallel optimization method.

In order to verify the efficiency and practicability of the I-SA search algorithm, field experiments are conducted at the mushroom base. The host computer of this *A. bisporus* picking robot is configured as CPU: i5-7300HQ @ 2.50 GHz; RAM: 8.00 GB; operating system: Win10; programming language: C++, because the advantage of C++ language lies in its fast running speed.

**6.1. Comparative Test of Algorithm Performance.** In the comparison test of algorithm performance, standard TSP experimental samples are used for testing. The coordinate distribution of each experimental sample is different from that of other samples, which can better test the comprehensiveness and practicability of the search algorithm.

**6.1.1. Comparison of Algorithms for Path Generation.** The following two scholars use their improved path generation algorithm in SA. Path improvement algorithms (PIA) proposed by Ziqiang et al. scholars is also a common algorithm at present. On the basis of the traditional path generation algorithm, the region rotation algorithm and the region inversion algorithm are added [20]. Yao Minghai scholars combined genetic with simulated annealing search algorithms (SAGAs), and a three-region exchange method, a

region DFS method, and a single-node optimal method are introduced in the path generation module.

This section is just to compare the path generation algorithms. Therefore, this experiment will not use the path initialization module and parallel optimization method. At the same time, the path generation algorithms of the two papers are separately extracted from the model and put into the simulated annealing model. Considering the time consumption of region DFS algorithm in SAGA, this algorithm only operates 9 adjacent nodes in the path at a time. Table 1 shows the optimal path solution, the worst path solution, the average value, population standard deviation (STDEVP), and the time-consuming algorithm of each experimental object. Each path generation algorithm performs 30 groups of experiment for each TSP experimental sample. The “optimal solution” column represents the optimal path solution in the 30 groups of experiment. And, the “worst solution” column represents the worst path solution in the 30 groups of experiment. The optimal path solution and the worst path solution are removed in each experiment when calculating the “average value” and “Std. dev.”

As shown in Table 2, the path generation algorithm in SAGA is not ideal and takes a long time, which is not suitable for the fast speed of the industry. The I-SA path

TABLE 2: Comparison of path generation algorithms.

Experimental samples	Path generation algorithms	Optimal solution	Worst solution	Average value	Std. dev.	Time consumption (s)
eil51	SAGA	448	482	460.86	10.44	572.7
	PIA	432	460	446.29	6.18	1.7
	I-SA	<b>426</b>	<b>445</b>	<b>436.07</b>	<b>5.22</b>	<b>1.8</b>
ei76	SAGA	581	626	597.46	14.73	1194.2
	PIA	558	592	570.86	6.98	5.6
	I-SA	<b>540</b>	<b>575</b>	<b>551.86</b>	<b>5.11</b>	<b>5.8</b>
ch150	SAGA	7408	7998	7696.79	161.26	3496.6
	PIA	6845	7416	7042.83	130.17	13.5
	I-SA	<b>6547</b>	<b>6710</b>	<b>6610.93</b>	<b>35.13</b>	<b>16.8</b>

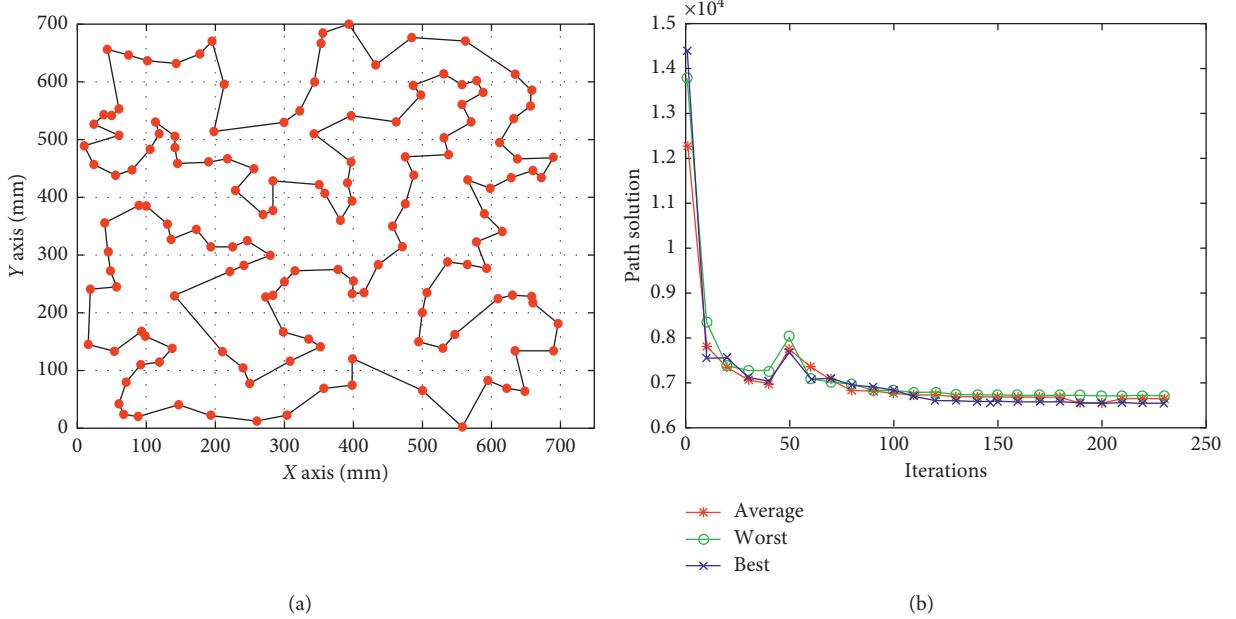


FIGURE 17: (a) Optimal path and (b) convergence graph of Ch150.

generation algorithm takes almost the same time with the PIA algorithm, but the final path solution is better than the PIA algorithm in all aspects. As can be seen from Table 1, the path generation algorithm in this paper can not only search better path solutions but also satisfy the industrial rapidity. At the same time, the optimal path and convergence graph of I-SA for Ch150 samples are shown in Figure 17.

**6.1.2. Comparison of Temperature Drop Modes.** The ch150 TSP experimental sample (the optimal path solution is 6528) is used in the comparison of Markov chain, VFSA, and T-VFSA methods. 30 groups of experiment are conducted for comparison. For the T-VFSA temperature drop method, the experimental results show that the tempering effect is the best when the temperature reaches  $10^\circ$  to  $1^\circ$ . Therefore, in the T-VFSA method, when the temperature reaches  $10^\circ$ , the temperature returns to the initial temperature ( $T_{\max}$ ). The corresponding parameter values of each temperature drop method are shown in Table 3. The number of iterations per temperature  $K$  is closely related to the final optimal path

solution, so  $K$  is added in this experiment. As with previous experiments, this experiment will not use the path initialization module and parallel optimization. The experimental results are shown in Table 3.

As can be seen from Table 4, the average value and variance of VFSA temperature drop method are basically stable after  $K$  exceeds 700, and there is no large fluctuation. When  $K=600$ , the traditional Markov chain temperature drop method has reached a plateau with excellent performance in all aspects, but the total number of iterations is very high, which is not suitable for the rapidity in the industry. Although the T-VFSA method is slightly worse than the Markov chain method in all aspects, the total number of iterations is reduced by a lot. In general, T-VFSA increases the stability of search algorithm and improves the operation speed to a certain extent, but if more stable path solutions are needed, the traditional Markov chain temperature drop method is still the best choice.

**6.1.3. Comparison of Search Algorithms.** Optimal path solutions for each TSP experimental sample are from the

TABLE 3: Parameter values.

Temperature drop method value	$\alpha$	$U$	Temperature	$T_{\max}$	$T_0$
VFSA	0.8	2	—	1000°	0.005°
Markov chain	0.98	—	10°	1000°	0.01°
T-VFSA	0.8	2	—	1000°	0.005°

TABLE 4: Comparison of temperature drop methods.

Temperature drop method	$K$	Optimum solution	Worst solution	Average value	Std. dev.	Total number of iterations
VFSA	600	6554	6818	6630.18	35.28	120000
	700	6554	6807	6621.04	45.08	140000
	800	6549	6732	6627.75	48.09	160000
	900	6549	6851	6614.75	40.48	180000
Markov chain	500	6549	6705	6602.46	42.70	285000
	600	6549	6740	6588.50	25.27	330000
T-VFSA	700	6549	6710	6610.93	35.13	154000
	800	6551	6711	6617.14	41.11	176000

TABLE 5: Comparison of search algorithms.

Experimental samples	Current optimal solution [31]	Method	Best	Average value	Std. dev.
eil51	427	I-SA	<b>426</b>	<b>426.64</b>	<b>0.78</b>
		GSAACS	427	427.27	0.45
		Mahi et al.	N/A	426.45	0.61
		SAGA	426	428	N/A
eil76	538	I-SA	<b>538</b>	<b>538.64</b>	<b>0.95538</b>
		GSAACS	538	540.2	2.94
		Mahi et al.	N/A	538.3	0.47
		SAGA	538	544	N/A
rat99	1211	I-SA	<b>1211</b>	<b>1211.93</b>	<b>1.12</b>
		I-SA	<b>21282</b>	<b>21303.96</b>	<b>39.54</b>
KroA100	21282	GSAACS	21282	21370.47	123.36
		Mahi et al.	N/A	21445.1	78.24
		I-SA	<b>6528</b>	<b>6552.18</b>	<b>7.51</b>
		GSAACS	6528	6563.70	22.45
ch150	6528	Mahi et al.	N/A	6563.95	27.58
		SAGA	6528	6584	N/A
		I-SA	<b>29375</b>	<b>29583.39</b>	<b>118.96</b>
		GSAACS	29383	29738.73	356.07
KroA200	29368	Mahi et al.	N/A	29646.05	114.71

standard database TSPLIB [31, 41]. Because the I-SA search algorithm is better than traditional SA and GA, so I-SA search algorithm is compared with GSAACS [42] and Mahi scholar's search algorithm [43].

The values of the parameters used by I-SA search algorithm are as follows:  $\beta = 0.2$  is used when the number of coordinate points is less than 100 and  $\beta = 0.1$  when the number of coordinate points is greater than 100; the number of iterations  $K$  corresponding to the experimental sample is 1, 10, 100, 100, 300, and 500 in sequence; the remaining parameters are the same as the T-VFSA parameters in Table 2. At the same time, the three-thread parallel optimization method is adopted. The high-temperature and low-temperature boundary values are set as 1° according to the experimental data. Therefore, the “rough exploration” parallel method is used when temperature is greater

than 1°, and the “precision exploration” parallel method is used when temperature is less than 1°.

The parameter values of MMSA used in the path initialization module are as follows: the value of  $M_A$  is the number of samples; the number of iterations  $K_A$  corresponding to the experimental sample is 500, 500, 600, 600, 800, and 1000 in sequence;  $\beta_A = 2$ ;  $\alpha_A = 1$ ;  $\rho = 0.95$ ;

From the results of Table 5, when the amount of data exceeds 100, the average value of the I-SA search algorithm and the sample standard deviation is superior to the other two algorithms. For Ch150 and KroA200 samples, the optimal path obtained by I-SA is shown in Figure 18. At the same time, the convergence graphs of these two samples are given, as shown in Figure 19. In the experiment, except for the worse path solution or the best path solution, when the

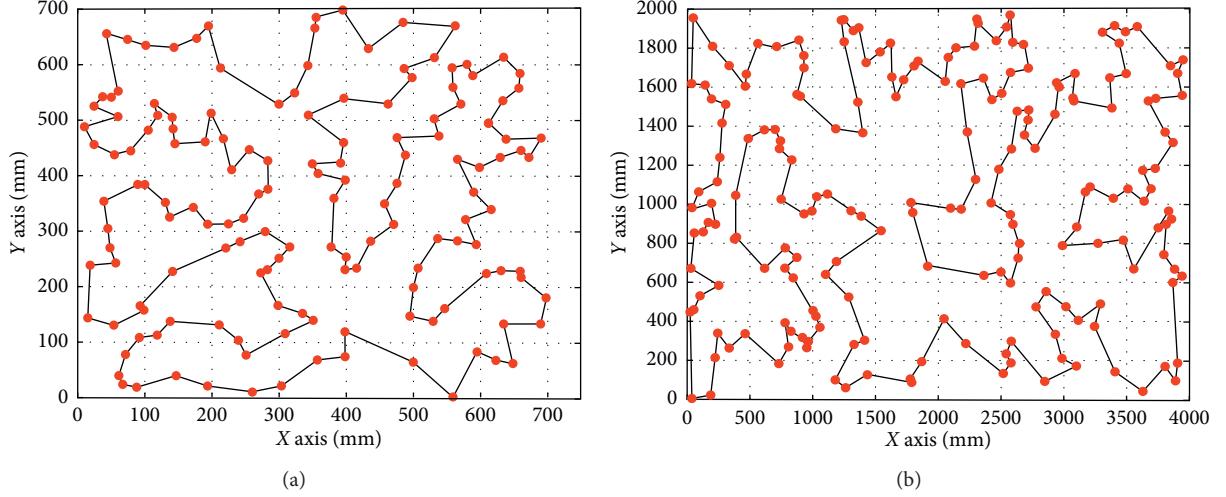


FIGURE 18: (a) Optimal paths of Ch150 and (b) KroA200.

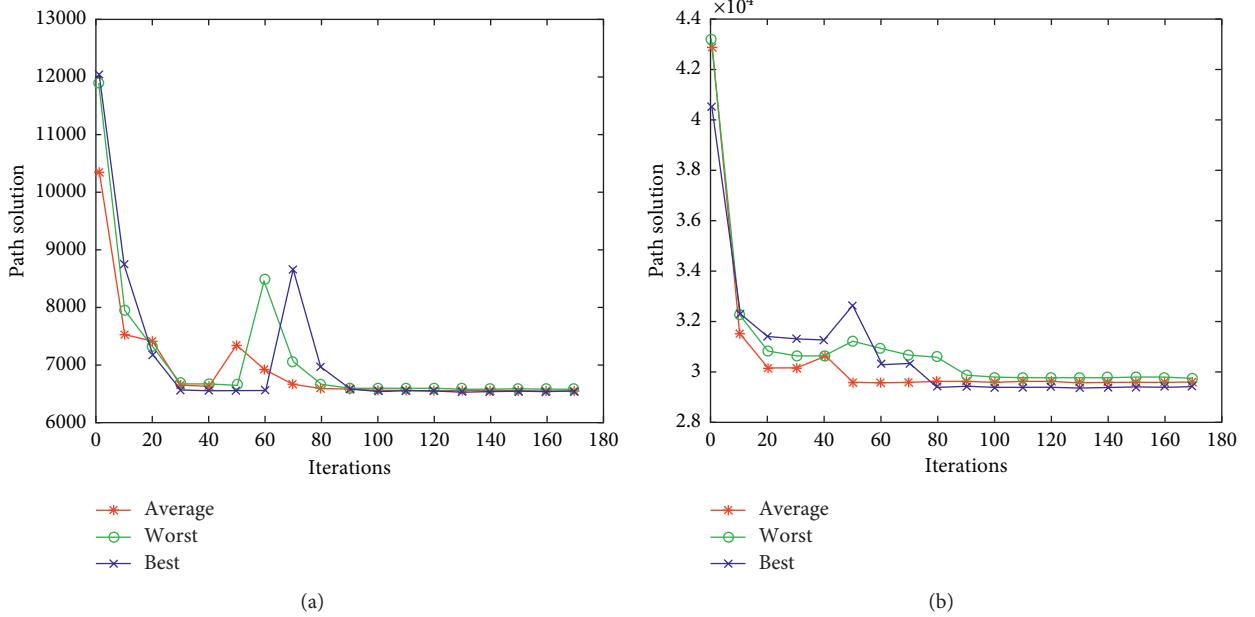


FIGURE 19: (a) Convergence graphs of Ch150 and (b) KroA200.

number of iterations reaches 100, the convergence is almost complete.

**6.2. Practical Application Stage.** At present, the range of the visual area is only 150 mm \* 1000 mm, which is still not big enough, and the image processing cannot distinguish the overlapping and interspersed *A. bisporus*. Therefore, this picking robot can only play an auxiliary role at present, but we are making more advanced optimizations for image processing and picking method. At present, the first batch of the grown *A. bisporus* is artificially picked, and the latter batches are picked by the robot. The number of mature *A. bisporus* (diameter greater than 35) that can be picked in the visual area is currently around 30–50.

**6.2.1. Picking Path of I-SA Search Algorithm.** The experimental data were obtained by testing at the mushroom breeding base. The number of *A. bisporus* is  $n = 35$ ;  $K = 1$ ;  $K_A = 300$ ; other parameters are the same as those in the appeal experiment; the two-thread parallel optimization method is adopted. The optimal path search time based on I-SA is 1.307 seconds, and the path solution of picking mechanism movement is 10398 mm. The picking path is shown in Figure 20. The sequence of the path for picking the *A. bisporus* is given below.

0--(108.84, 220)--15--(193.86, 220)--21--(265.21, 220)--19--(356.22, 220)--22--(456.74, 220)--24--(511.15, 220)--11--(536.57, 220)--25--(581.56, 220)--5--(620.51, 220)--7--(657.45, 220)--28--(740.11, 220)--3--(755.96, 220)--29--(804.20,

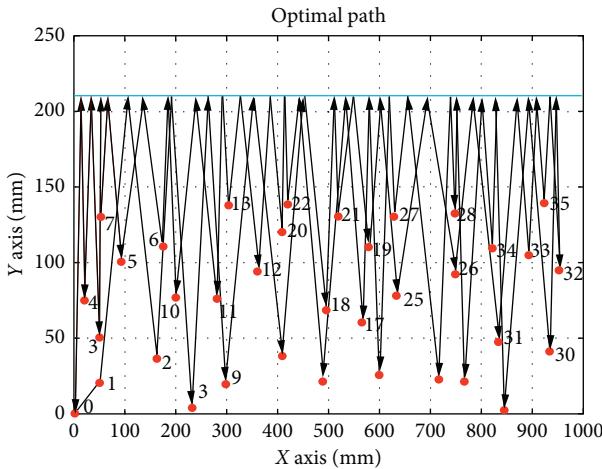


FIGURE 20: The picking path.

FIGURE 21: Picking experiment of *A. bisporus* picking robot.

TABLE 6: Comparison of picking experiments.

Group number	Vision	Before (s)	After (s)	Save (s)
1	33	531.6	441.9	89.7
2	30	474.3	398.4	75.9
3	35	564.2	465.7	98.5
4	44	673.2	565.5	107.7

220)--32--(831.64, 220)--34--(896.25, 220)--35--(947.71, 220)--30--(936.96, 220)--4--(912.74, 220)--33--(871.14, 220)--31--(785.48, 220)--27--(694.87, 220)--26--(603.95, 220)--6--(548.00, 220)--8--(440.43, 220)--23--(415.46, 220)--10--(388.01, 220)--18--(325.38, 220)--17--(294.037, 220)--20--(239.605, 220)--14--(185.251, 220)--16--(133.985, 220)--13--(68.34, 220)--2--(51.32, 220)--12--(33.80, 220)--9--(12.06, 220)--0.

**6.2.2. Picking Efficiency Test.** The experimental test is shown in Figure 21. The picking path optimization algorithm is put into the program of the picking robot for actual picking.

After completing the picking test, four groups of comparative experiments are performed. The coordinate points of *A. bisporus* in a visual region were obtained by image processing. According to this coordinate point, the picking time of the non-path optimization and the path optimization is timed separately. The “vision” column indicates the number of *A. bisporus* identified in this vision area; the “before” column indicates the time required to pick the *A.*

*bisporus* without using the picking path optimization algorithm. The “after” column indicates the time required to pick the *A. bisporus* using the picking path optimization algorithm; the “save” column indicates the time saved after using the path optimization. From the results in Table 6, the improvement of efficiency through path optimization is 14%–18%. The formula for improvement of efficiency is  $(\text{Save}/\text{Before}) \times 100\%$ .

## 7. Conclusion

The design and application of *A. bisporus* picking robot is one of the current research hotspots. Because the working efficiency of *A. bisporus* picking robot is critical, based on the functional structure of the self-developed *A. bisporus* picking robot, the control flow is designed and the path optimization module is introduced into the control flow. At the same time, in order to further improve the efficiency, the parallel control mode is used to work simultaneously in the picking area and the visual area. According to the control flow, the traditional TSP model is modified to optimize the path of the picking positions and the unknown storage positions. Based on the traditional simulated annealing, the I-SA search algorithm is proposed. Moreover, a parallel optimization method of “rough exploration” and “fine exploration” is proposed, which makes the I-SA search algorithm more stable and can obtain better path solution results and at the same time speeds up the operation to a certain extent. From the results of TSP experiments, when the number of data is more than 100, better path solution can be obtained in a shorter time. Therefore, this I-SA search algorithm is suitable for picking path optimization. In addition, in the mushroom base test, the picking efficiency increased by 14%–18% after path optimization, which shows the feasibility and effectiveness of the algorithm.

## Data Availability

Anyone can get our data by sending an email to panzhaoren12@163.com.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

## Acknowledgments

This work was supported by the Science and Technology Agriculture Project of the Agricultural Committee of Shanghai, China (no. 19111101303).

## Supplementary Materials

In the “photo” file, there are pictures of our self-developed robot and pictures used for image processing. So that readers can better understand the content and value of this paper. In the “supplementary information” file, the corresponding experimental data in our paper are given. “Path generation algorithms. text” corresponds to Table 1, “Comparisons of

temperature drop modes. txt” corresponds to Table 3, “I-SA three parallel. txt” corresponds to Table 4, and “Coordinates of Agaricus bisporus. txt” corresponds to mushroom coordinate point data. (*Supplementary Materials*)

## References

- [1] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [2] H. Xiaomei, W. Chuan, and Y. Tao, “Design and application of visual system in the Agaricus bisporus picking robot,” *Journal of Physics: Conference Series*, vol. 1187, no. 3, 2019.
- [3] H. Xiaomei, P. Zhaoren, Y. Shuzhen, and Y. Tao, “Study on the control system of Agaricus bisporus picking robot,” *Journal of Physics: Conference Series*, vol. 1187, no. 3, 2019.
- [4] M. K. M. Ali and F. Kamoun, “Neural networks for shortest path computation and routing in computer networks,” *IEEE Transactions on Neural Networks*, vol. 4, no. 6, pp. 941–954, 1993.
- [5] Y. Marinakis, M. Marinaki, and G. Dounias, “A hybrid particle swarm optimization algorithm for the vehicle routing problem,” *Engineering Applications of Artificial Intelligence*, vol. 23, no. 4, pp. 463–472, 2010.
- [6] K. Savla, E. Frazzoli, and F. Bullo, “Traveling salesperson problems for the Dubins vehicle,” *IEEE Transactions on Automatic Control*, vol. 53, no. 6, pp. 1378–1391, 2008.
- [7] I. R. Obregon and A. P. Pavlovsky, “A hybrid SA-EA method for finding the maximum number of switching gates in a combinational circuit,” *IEICE Electronics Express*, vol. 5, no. 18, pp. 756–761, 2008.
- [8] R. Gan, Q. Guo, H. Chang, and Y. Yi, “Improved ant colony optimization algorithm for the traveling salesman problems,” *Journal of Systems Engineering and Electronics*, vol. 21, no. 2, pp. 329–333, 2010.
- [9] S. Yang and P.-F. Shi, “Bidirectional automated branch and bound algorithm for feature selection,” *Journal of Shanghai University*, vol. 9, no. 3, pp. 244–248, 2005.
- [10] H. Sakoe and S. Chiba, “Dynamic programming algorithm optimization for spoken word recognition,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 43–49, 1978.
- [11] L. I. Xu and Z. Liu, “Novel quantum ant colony algorithm for TSP,” *Computer Engineering & Applications*, 2011.
- [12] T. Stützle and H. H. Hoos, “Max-min ant System,” *Future Generation Computer Systems*, vol. 16, no. 8, pp. 889–914, 2000.
- [13] C. C. Queirolo, L. Silva, O. R. P. Bellon, and M. P. Segundo, “3D face recognition using simulated annealing and the surface interpenetration measure,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 2, pp. 206–219, 2010.
- [14] Y. Segundo, H. Yan, X. Zou, F. Tao, and L. Zhang, “Image threshold processing based on simulated annealing and OTSU method,” in *Proceedings of the 2015 Chinese Intelligent systems Conference*, Springer, Berlin, Germany, November 2015.
- [15] L. Kang and T. Yang, “Study of hybrid genetic algorithm based on simulated annealing in electronic image processing,” in *Advances in Mechanical and Electronic Engineering*, Springer, Berlin, Germany, 2013.
- [16] B. Haiying, “Application of simulated annealing genetic algorithm in balance optimization of construction resources of hydropower projects,” *Technical Supervision in Water Resources*, vol. 5, pp. 108–112, 2018.
- [17] J. Wei, G. Liu, and Y. Zhang, “Particle swarm optimization based on simulated annealing for solving constrained optimization problems,” *Systems Engineering and Electronics*, vol. 32, no. 7, pp. 1532–1536, 2010.
- [18] M. Yao, N. Wang, and L. Zhao, “Improved simulated annealing algorithm and genetic algorithm for TSP,” *Computer Engineering and Applications*, vol. 49, no. 14, pp. 60–65, 2013.
- [19] C. García, *A Simulated Annealing Method Based on a Specialised Evolutionary Algorithm*, Elsevier Science Publishers B. V., Amsterdam, Netherlands, 2012.
- [20] L. Ziqiang, Y. Zhang, B. Wen, and X. Guo, “An evolution algorithm based on greedy idea and cloud model for TSP,” *Journal of PLA University of Science and Technology*, pp. 1–3, 2018.
- [21] A. H. Alavi and A. H. Gandomi, “Prediction of principal ground-motion parameters using a hybrid method coupling artificial neural networks and simulated annealing,” *Computers & Structures*, vol. 89, no. 23–24, pp. 2176–2194, 2011.
- [22] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [23] P.-F. Xu, Q.-G. Miao, W.-S. Li, and J.-Y. Zhang, “Adaptive simulated annealing algorithm and tabu search algorithm based on the function complexity,” *Acta Electronical Sinica*, vol. 40, no. 6, pp. 1218–1222, 2012, in Chinese.
- [24] X. Geng, Z. Chen, W. Yang, D. Shi, and K. Zhao, “Solving the traveling salesman problem based on an adaptive simulated annealing algorithm with greedy search,” *Applied Soft Computing*, vol. 11, no. 4, pp. 3680–3689, 2011.
- [25] L. Ingber, “Very fast simulated annealing,” *Mathematical and Computer Modelling*, vol. 12, no. 8, pp. 967–973, 1989.
- [26] L.-S. Zhao, M. K. Sen, P. Stoffa, and C. Frohlich, “Application of very fast simulated annealing to the determination of the crustal structure beneath Tibet,” *Geophysical Journal of the Royal Astronomical Society*, vol. 125, no. 2, pp. 355–370, 2010.
- [27] C. Ortiz-Aleman, R. Martin, and J. C. Gamio, “Reconstruction of permittivity images from capacitance tomography data by using very fast simulated annealing,” *Measurement Science and Technology*, vol. 15, no. 7, pp. 1382–1390, 2004.
- [28] A. Biswas, “Interpretation of residual gravity anomaly caused by simple shaped bodies using very fast simulated annealing global optimization,” *Geoscience Frontiers*, vol. 6, no. 6, pp. 875–893, 2015.
- [29] H. Chen, L. Li, H. Xu, and B. Chen, “Modified very fast simulated annealing algorithm,” *Journal of Tongji University*, vol. 34, no. 8, pp. 1121–1125, 2006.
- [30] H. Cheng, X. Wang, S. Yang, and M. Huang, “A multi-population parallel genetic simulated annealing-based QoS routing and wavelength assignment integration algorithm for multicast in optical networks,” *Applied Soft Computing*, vol. 9, no. 2, pp. 677–684, 2009.
- [31] S. Wanneng and Z. Shijue, “A parallel genetic simulated annealing hybrid algorithm for task scheduling,” *Wuhan University Journal of Natural sciences*, vol. 11, no. 5, pp. 1378–1382, 2006.
- [32] H. Gehring and A. Bortfeldt, “A parallel genetic algorithm for solving the container loading problem,” *International Transactions in Operational Research*, vol. 9, no. 4, pp. 497–511, 2010.
- [33] S. B. Gelfand, *Analysis of Simulated Annealing for Optimization*, Massachusetts Institute of Technology, Cambridge, MA, USA, 1987.

- [34] H. Adeli and P. Vishnubhotla, "Parallel processing," *Computer-Aided Civil and Infrastructure Engineering*, vol. 2, no. 3, pp. 257–269, 2010.
- [35] M. F. Boston, J. T. Hale, S. Vasishth, and R. Kliegl, "Parallel processing and sentence comprehension difficulty," *Language and Cognitive Processes*, vol. 26, no. 3, pp. 301–349, 2011.
- [36] G. V. D. Engh and W. Stokdijk, "Parallel processing data acquisition system for multilaser flow cytometry and cell sorting," *Cytometry*, vol. 10, no. 3, pp. 282–293, 2010.
- [37] A. T. Zimmerman and J. P. Lynch, "A parallel simulated annealing architecture for model updating in wireless sensor networks," *IEEE Sensors Journal*, vol. 9, no. 11, pp. 1503–1510, 2009.
- [38] M. Kovac, "Solving quadratic assignment problem in parallel using local search with simulated annealing elements," in *Proceedings of the International Conference on Digital Technologies*, May 2013.
- [39] L.-L. Dong, N. Li, and G.-H. Gong, "Adaptive & parallel simulated annealing genetic algorithm based on cloud model," in *Proceedings of the International Conference on Intelligent Computing & Integrated Systems*, October 2010.
- [40] S. Hua, G. Liangxian, and G. Chunlin, "Hybrid simulated annealing algorithm based on the parallel strategy," in *Proceedings of the International Symposium on Computational Intelligence & Design*, October 2011.
- [41] Standard, TSP Test Library [EB/OL]: 2012, <http://www.iwr.uniheidelberg.de/groups/comopt/software-TSPLIB95/tsp/>.
- [42] S.-M. Hoos and C.-Y. Chien, "Solving the traveling salesman problem based on the genetic simulated annealing ant colony system with particle swarm optimization techniques," *Expert Systems with Applications*, vol. 38, no. 12, pp. 14439–14450, 2011.
- [43] M. Mahi, Ö. K. Baykan, and H. Kodaz, "A new hybrid method based on particle swarm optimization, ant colony optimization and 3-opt algorithms for traveling salesman problem," *Applied Soft Computing*, vol. 30, pp. 484–490, 2015.

