PATH PLANNING OF FRUIT AND VEGETABLE PICKING ROBOTS BASED ON IMPROVED A* ALGORITHM AND PARTICLE SWARM OPTIMIZATION ALGORITHM /

*基于改进 A*算法与粒子群算法的果蔬采摘机器人路径规划*

Chen LI *
School of Applied Technology, Zhumadian Preschool Education College, Zhumadian, Henan, China
**E-mail: chenli6780@126.com*
Corresponding author: Li Chen

DOI: https://doi.org/10.35633/inmateh-71-41

Keywords: *agricultural robot, fruit and vegetable picking, path planning, A* algorithm, particle swarm optimization*

ABSTRACT

Aiming at the suboptimal local path, slow convergence speed, and many inflection points in the path planning of fruit and vegetable picking robots in complex environments, a global planning method combining particle swarm optimization (PSO) algorithm and A algorithm was proposed. Firstly, Manhattan distance was taken as a heuristic function of global programming based on the A* algorithm. Secondly, the step size of PSO was adjusted to optimize the path, shorten the path length, and reduce the number of inflection points. Finally, the planned path of the fruit and vegetable picking robot was smoothed so that it could steadily move along a smoother driving path in real scenarios. The experimental results show that compared with the traditional PSO algorithm, the hybrid algorithm based on the improved A* algorithm and PSO algorithm achieves a smoother path and fewer folding points. In comparison with the PSO algorithm, moreover, this algorithm can guarantee the path generation efficiency and the global optimum. In the end, the effectiveness of the proposed method was verified by shortening the path length and reducing the accumulative number of inflection points.*

摘要

*针对复杂环境下果蔬采摘机器人路径规划中存在的局部路径次优、收敛速度慢、拐点多等问题，提出了一种基于粒子群优化算法和 A*算法相结合的全局规划方法。首先，基于 A*算法，将曼哈顿距离作为全局规划的启发式函数。其次，通过调整粒子群算法的步长来优化路径，缩短路径长度，减少拐点数量。最后，对果蔬采摘机器人规划路径进行平滑处理，使机器人能够在真实场景中平稳移动，使采摘机器人行驶路径更加平滑。实验结果表明，与传统的粒子群优化算法相比，基于改进 A*算法和粒子群算法的混合算法具有更平滑的路径和更少的折叠点。与粒子群优化算法相比，该算法能够保证生成路径的效率和全局最优性。通过缩短路径长度和减少累积拐点数量，验证了该方法的有效性。*

INTRODUCTION

With the development of machinery manufacturing industry in China, the popularization rate of agricultural machinery and fruit and vegetable machinery has gradually increased in recent years, accompanied by the elevated specialization level. Restricted by various factors, however, the automation degree of fruit and vegetable picking operations is very low, and a large quantity of manpower needs to be mobilized to collectively pick fruits and vegetables using picking tools like ladders, baskets, and scissors. Other picking methods have also been tried, e.g., picking the whole spurs of fruits and vegetables with long-handled garden shears or pulling down fruits and vegetables by rotating long-handled rubber claws, and even shaking fruits on the fruit tree in outdoor fruit and vegetable planting areas with the help of large-scale team and focusing on picking fruits falling to the ground. Meanwhile, picking robots have also been actively investigated by universities and research institutions in China *(Wang et al., 2012)*. Nevertheless, most of the current scientific picking methods fail to catch up with the flexibility and completeness of artificial picking in the current stage, so the actual popularization degree of intelligent fruit and vegetable picking remains to be improved.

Agriculture is one of the best scenarios for artificial intelligence to be put into practice. As an emerging field characterized by the deep integration of information technology and agriculture, intelligent agriculture has become an inevitable development trend through the organic combination of artificial intelligence and agriculture.

At present, picking environments have become complicated and diversified due to factors like the overlapping of fruit spurs in fruit trees, and the intelligence, automation, and mechanization levels of fruit and vegetable picking are relatively low *(Xu et al., 2014)*. Obstacles such as naturally growing branches and immature fruits bring difficulties to the picking path control design of picking equipment *(Gao et al., 2014)*. In order to improve the intelligence level of fruit and vegetable picking, liberate manpower, and improve the harvest quality and mechanization level, it is also necessary to improve the intelligent decision-making level of fruit and vegetable picking robots' dynamic trajectory planning in unstructured natural environments in addition to strengthening the standardized management of agronomy in the process of fruit tree planting *(Ji et al., 2011)*. Path planning is one of the most basic links for picking robots to realize autonomous navigation. It means that in the picking working environment with obstacles, the picking robot senses the picking environment by using the sensor device it carries, so as to continuously obtain information such as the size, position, distance, and obstacles of the working area to be picked. The parameters for picking robots to search for the optimal path should be planned, including the shortest, the safest, and collision-free path *(Liu et al., 2010)*. According to the understanding of the environment where the picking robot is located, path planning can be divided into two categories: global path planning with all the environment known and local path planning with environmental information partially known or unknown. The working environment of picking robots is generally known, so intelligent algorithms can be introduced to effectively improve the path planning efficiency of picking robots through the global-coverage path planning method.

STATE OF THE ART

In recent years, picking robots have attracted worldwide attention, and many research institutions and universities have successively carried out related research and development work, including mechanical structure design, visual system development, and intelligent algorithm research, and made some progress *(Grau et al., 2015)*. Satyam et al. designed a fruit and vegetable picking robot with seven degrees of freedom, which has accurate positioning ability with the picking success rate exceeding 80% *(Paul et al., 2021)*. Li et al. developed a fruit and vegetable picking robot, which acquired the contour of fruits and vegetables combining K-means clustering segmentation with Canny edge detection method, and then used the minimum circle wrapping method to extract the centre coordinates of fruits and vegetables. Finally, the spatial trajectory of the picking robot arm was drawn according to the quintic polynomial interpolation law to realize fruit and vegetable grabbing *(Li et al., 2019)*. Path planning is one of the key technologies of fruit and vegetable picking robots, which is mainly divided into global path planning based on known map environment and local path planning based on sensor information *(Orozco-Rosas et al., 2022)*. Global path planning mainly includes finding a path that connects the starting point and the target point while avoiding collision with obstacles. Therein, the path length, algorithm time, and spatial complexity are the main indexes to judge the performance of global path planning algorithms *(Tian et al., 2021)*. Commonly used global path planning algorithms are A* algorithm *(Chen et al., 2020)*, RRT algorithm *(Qureshi et al., 2021)*, and genetic algorithm (GA) *(Su et al., 2016)*. Among them, the RRT algorithm proposed by Sun et al. has complete probability, simple structure, and flexible search ability, making it suitable for path planning in various complex environments. Due to its random sampling mechanism, however, the RRT algorithm is subjected to such problems as low node utilization rate, high path complexity, and slow convergence *(Sun et al., 2019)*.

Local path planning focuses on the safety and real-time performance of the path, and the typical local path planning algorithms include A* algorithm *(Li et al., 2019)*, artificial potential field method *(Orozco-Rosas et al., 2019)*, and particle swarm optimization (PSO) algorithm *(Cao et al., 2019)*. In recent years, with the progress of sensor technology, scholars have made relevant research on path planning of agricultural robots based on machine vision *(Mondal et al., 2021)* or Lidar *(Ma et al., 2022)*. Wu et al. obtained crop row information from the camera, eliminated outliers based on the random sample consistency (RANSAC) algorithm, and extracted navigation paths using the least square method (LSM). The path detection rate can reach more than 90% *(Wu et al., 2020)*. Peng et al. improved the robustness and applicability of crop row extraction based on the automatic Hough transform-cumulative threshold method. Lidar sensors can accurately obtain environmental distance information along with strong anti-interference ability *(Peng et al., 2021)*. Meng et al. obtained the position of fruit trees using a laser scanner and planned the navigation path of agricultural machinery via LSM *(Meng et al., 2022)*. Ping et al. used two-dimensional Lidar to obtain the point clouds of vineyards, and proposed a vineyard path planning algorithm based on Support Vector Machine *(Ping et al., 2017)*.

Masumori et al. used 16-line Lidar to obtain three-dimensional point cloud information of maize, introduced confidence interval and K-means algorithm to cluster maize plants, and finally used the radial basis function path planning method to avoid plant trunks as much as possible.

The existing research mostly focuses on single plant or crop row detection, and the planned path is straight, which is convenient for robot control but not conducive to avoiding obstacles such as tree crowns and pedestrians between rows *(Masumori et al., 2020)*.

To sum up, the irregular crowns of fruit trees constitute one of the hidden dangers impeding the robot when doing fruit and vegetable picking operations. The traditional orchard planning paths are mostly straight lines, which is not conducive to avoiding obstacles such as crowns. In this study, taking fruit and vegetable picking as the research object, an improved artificial potential field method was proposed to optimize the robot navigation path between rows, specifically as follows: The A* algorithm was used to extract the ridge line and initial path, while the crowns of fruit trees and pedestrians were regarded as obstacles. Then, the initial path was optimized combining the PSO algorithm. Finally, a safe and reliable driving path for the picking robot was planned, expecting to provide technical reference for the picking path planning of mobile picking robots in orchards.

MATERIALS AND METHODS

Problem description

In an orchard, several fruit and vegetable picking robots were equipped to perform fruit and vegetable picking operations. In this study, the picking path of fruit and vegetable picking robots was studied, and the layout of the orchard is shown in Fig. 1. It is assumed that the fruit and vegetable picking robots could move vertically and horizontally in the orchard, which consisted of fruit trees, ridges, fruit and vegetable picking robots, and I/O starting points, where the I/O starting points were the starting and ending positions of the fruit and vegetable picking robots. It is assumed that the length of a single fruit tree in the orchard in the X direction is l, the width in the Y direction is w, the height is h, the ridge width is C, and the I/O starting point is rightly opposite to the first ridge. The orchard layout plan structure is as shown in Fig. 1.

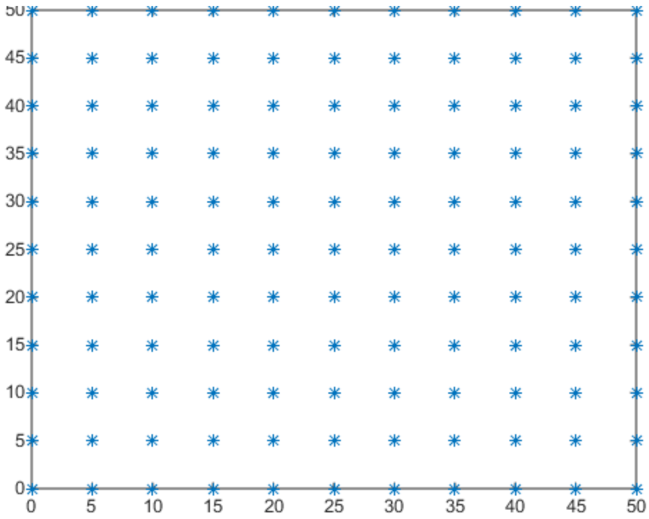

Fig. 1 - Orchard Layout Plan

In order to facilitate observation and calculation, every fruit tree in the orchard was arranged and distributed in the form of a grid. It is assumed that the system receives n orders in total, and the storage information of fruits and vegetables in each order is known. The maximum carrying weight of the fruit and vegetable picking robots is M, and the number of self-carried goods is Q, that is, the fruit and vegetable picking robots can carry at most Q fruits and vegetables in a single trip. On the premise that the fruit and vegetable picking robots met the double constraints of carrying capacity and container number, all the fruits and vegetables to be picked in the order were picked in batches, and the picking time of each batch was added. The optimization goal was to obtain the picking path with the least total time after addition.

Basic assumptions

Before establishing the mathematical model for picking path planning of fruit and vegetable robots, the following assumptions and parameters were set for fruit and vegetable picking robots and shelves:

(1) Each picking unit (Unit) in a picking task list of a picking robot corresponds to each fruit tree in the orchard;

(2) The number of Units in each fruit tree meets the number of orders selected;

(3) When the fruit and vegetable picking robot performs a certain action, it cannot perform other actions at the same time, and it will not interrupt the current action due to other factors, thus ensuring the continuity of picking operations;

(4) When starting to perform the picking task, the fruit and vegetable picking robot starts from the I/O starting point and needs to return to the I/O point after completing the picking task or reaching the constraint conditions;

(5) When the fruit and vegetable picking robot moves from one fruit tree to another, it needs to go through acceleration, uniform velocity, and deceleration stages. Assuming that the acceleration and decelerations of its horizontal movement are both under no-load and loaded conditions, the vertical velocity of the lifting platform in the vertical direction is the same as the falling velocity, which is set as v2.

Modelling

The path planning model for batched picking of fruit and vegetable picking robots was established as follows. Firstly, the symbols appearing in the optimization model are explained: *O* stands for the order set; S represents a set of fruit trees; B={0,1,2,......,n} represents the set of picking points, where 0 is the starting point (sorting table); (x_i,y_i,z_i,m_i) presents the information on the Units to be picked on the order *O*, for example, (5,6,2,1) indicates the fruit trees of one Unit in Columns 5 and 6 in the orchard, with the height of 2 m and number of 1.

The picking time of fruit and vegetable picking robots for a complete order consists of 3 periods:
① the horizontal moving time between picking points, that is, the time T_1 required to move on the xoy surface;
② In the z-axis direction, the sum of the time needed by the lifting platform to rise and fall is T_2;
③ The picking and unloading time for Unit by fruit and vegetable picking robots is fixed, and the total time is T_3. The order picking time of fruit and vegetable picking robots is expressed as:

$$T=T_1+T_2+T_3 \tag{1}$$

Assuming that the horizontal moving distance of the picking robot is r_{ijs}.

To facilitate the path planning of picking robots, their horizontal trajectory r_{ijs} is divided into 3 stages:

(1) The horizontal distance from the I/O starting point to the first fruit and vegetable picking point *j*.

$$r_{ijs}=\begin{cases}(x_j\text{-}2)\times 1+(\frac{x_j\text{-}2}{2})\times c+y_j\times w, x_i \text{ is the starting point, } x_j\neq 1 \text{ or } 2\text{, and it's even}\\(x_j\text{-}1)\times 1+(\frac{x_j\text{-}1}{2})\times c+y_j\times w, x_i \text{ is the starting point, } x_j\neq 1 \text{ or } 2\text{, and an odd number}\\y_j\times w,\ x_i \text{ is the starting point, } x_j=1 \text{ or } 2\end{cases} \tag{2}$$

where both x_i and x_j represent the column positions of the picking point.

(2) The horizontal distance from the current fruit and vegetable picking point *i* to the fruit and vegetable picking point *j* to be picked, which is divided into two circumstances. When the fruit and vegetable picking points *i* and *j* are located at different ridges, namely, $x_i \neq x_j$. If the total numbers of rows at points *i* and *j* is smaller than *f*, the robot will choose to move out from the next ridge. Therein, *f* represents the number of rows of a single-column shelf in the y-axis direction.

$$r_{ijs}=\begin{cases} 2l+c+(y_i+y_j)\times w, x_j\text{-}x_i=1,\ x_j \text{ is even}, y_i+y_j \le f \\ (x_j\text{-}x_i\text{-}1)\times l+(\frac{x_j\text{-}x_i\text{-}1}{2})\times c+(y_i+y_j)\times w, x_j\text{-}x_i>1,\ x_j \text{ is odd number}, y_i+y_j \le f \\ (x_j\text{-}x_i+1)\times l+(\frac{x_j\text{-}x_i+1}{2})\times c+(y_i+y_j)\times w, x_j\text{-}x_i>1, \text{odd number},\ x_j \text{ is even},\ y_i+y_j \le f \\ (x_j\text{-}x_i)\times l+(\frac{x_j\text{-}x_i}{2})\times c+(y_i+y_j)\times w, x_j\text{-}x_i>1, \text{it's even},\ y_i+y_j \le f \end{cases} \tag{3}$$

If the total number of rows at points *i* and *j* is greater than *f*, the robot will move out from the last ridge.

$$r_{ijs}=\begin{cases} 2l+c+(2f\text{-}y_i\text{-}y_j)\times w,\ x_j\text{-}x_i=1,\ x_j \text{ is even},\ y_i+y_j>f \\ (x_j\text{-}x_i\text{-}1)\times l+(\frac{x_j\text{-}x_i\text{-}1}{2})\times c+(2f\text{-}y_i\text{-}y_j)\times w,\ x_j\text{-}x_i>1, \text{odd number},\ x_j \text{ is odd number},\ y_i+y_j>f \\ (x_j\text{-}x_i+1)\times l+(\frac{x_j\text{-}x_i+1}{2})\times c+(2f\text{-}y_i\text{-}y_j)\times w,\ x_j\text{-}x_i>1, \text{odd number},\ x_j \text{ is even},\ y_i+y_j>f \\ (x_j\text{-}x_i)\times l+(\frac{x_j\text{-}x_i}{2})\times c+(2f\text{-}y_i\text{-}y_j)\times w,\ \ x_j\text{-}x_i>1, \text{even},\ y_i+y_j>f \end{cases} \tag{4}$$

Vegetable picking points *i* and *j* are in the same ridge.

(3) The horizontal distance from the last shelf picking point sorted by the picking robot in a single trip to the I/O starting point.

$$r_{ijs}=\begin{cases} (x_i\text{-}1)\times l+\frac{(x_i\text{-}1)}{2}\times c+y_i\times w,\ x_i\neq 0 \text{ and an odd number},\ x_i=0 \\ x_i\times l+\frac{x_i}{2}\times c+y_i\times w,\ x_i\neq 0 \text{ and it's even},\ x_j=0 \end{cases} \tag{5}$$

Assuming that the picking robot moves under the fruit and vegetable to the picked, the vertical moving distance of the lifting platform is d_1, expressed as follows:

$$d_1=\left|z_j\text{-}z_i\right|\times h \tag{6}$$

where z_i is the location of fruit and vegetable *i*; z_j is the location of fruit and vegetable *j*. The time T_1 of the picking robot in the horizontal direction depends on the maximum velocity and acceleration when moving. Assuming that the maximum velocity that the picking robot can reach when moving horizontally is v_1, it can be divided into two motion states: first, the velocity of the picking robot does not reach the maximum velocity v_1 during the movement on *xoy* plane, and then it makes uniform deceleration motion after uniform acceleration, that is, the moving distance of the picking robot is $s<\frac{v_1^2}{a}$; second, the velocity of the picking robot reaches the maximum velocity v_1 during the movement on *xoy* plane, and it moves at a uniform velocity later and then moves at a uniform deceleration, that is, the moving distance is $s<\frac{v_1^2}{a}$.

The driving time of the picking robot for the n^{th} time is calculated according to the horizontal moving distance r_{ijs} and whether the maximum velocity is reached, as below:

$$T_{1n}=\begin{cases} \left(\frac{r_{ijs}}{a}\right)^{\frac{1}{2}},\ r_{ijs}<\frac{v_1^2}{a} \\ \frac{2v_1}{a}+\frac{r_{ijs}\text{-}\frac{v_1^2}{a}}{v_1},\ r_{ijs}\geqslant\frac{v_1^2}{a} \end{cases} \tag{7}$$

The vertical rising and falling time of the picking robot's mechanical arm for the n^{th} time is T_{2n}, expressed as follows:

$$T_{2n}=d_{1n}/v_2 \tag{8}$$

Assuming that the loading time and unloading time spent by the picking robot when picking a single piece of fruit and vegetable are identical, being *s*, then the loading and uploading time spent by the picking robot in completing the picking of one batch of fruits and vegetables is:

$$T_3=n\times s \tag{9}$$

According to the above description, the model for the picking robot's picking operation time can be expressed as below:

$$T_{min}=\sum_{n=1}^{n}\sum_{i=1}^{n}\sum_{j^1i}^{n}(T_{1n}+T_{2n})\times K_{ij}+T_3 \tag{10}$$

The constraint conditions are displayed as follows:

$$\sum_{i=1}^{Q}m_i\leqslant M,\ \ i=1,2,L,Q \tag{11}$$

$$q<Q \tag{12}$$

$$\sum_{i=1}^{n}\sum_{j\neq i}^{n}b_{ij}=n,b\in B;\ i=1,2,L,n;\ j=1,2,L,n \tag{13}$$

$$K_{ij}=\begin{cases}1,\textit{the selection point i is selected, and then the}\\ \textit{selection j operation is carried out}, i,\ j=1,2,L,n\\ 0,\textit{other}\end{cases} \tag{14}$$

$$m_j=m_i\times K_{ij}+1 \tag{15}$$

$$\sum_{m}X_O^m=\sum_{m}X_{m+1}^O=X_O \tag{16}$$

Where:

Equation (10) indicates the optimization indexes for picking time; Equation (11) means that the weight of fruits and vegetables picked by the picking robot in a single trip cannot exceed *m*, or otherwise, it needs to return to the starting point first and upload for the next picking operation; Equation (12) represents the maximum number of grids of the picking robot, i.e., at most *Q* units can be accommodated in a single trip; Equation (13) reflects that all fruits and vegetables displayed in the order will be picked; Equations (14) and (15) represent the picking sequence of Units, and the continuity of the picking robot when picking a Unit is guaranteed; in Equation (16), X_O^m indicates that the picking robot returns to the starting point after Units are full loaded in the *m*-th trip or the weight reaches the upper limit, and X_{m+1}^O means that the picking robot starts from the starting point to pick up fruits and vegetables with empty load in the m+1-th trip, namely, the final full-loading position in the *m* -th trip equals the starting position in the *m* +1-th trip.

A* algorithm

The A* algorithm, a heuristic search method with high search efficiency, can solve the shortest and most effective path in static environments. In A*, the weight of each path is the sum of two costs, namely, the cost from the starting node to the current node and the heuristic distance. Searching in 8 directions from the starting point, the values of extension nodes are acquired using an evaluation function for comparisons, the minimum value is chosen as the next extension node, and this process is circulated until the end of the search, thus obtaining the final path. Since the node with the minimum estimated value is used as the next extension node in each search, the final cost is the lowest. The evaluation function is shown in Eq. (17):

$$f(n)=g(n)+h(n) \tag{17}$$

In Equation (17), *f*(*n*) is the valuation function of node *n*; *g*(*n*) represents the actual cost from the starting node to node *n*; *h*(*n*) represents the estimated cost from node *n* to the ending node. The determination of the valuation function is closely related to the actual situation and will directly affect the result of the A* algorithm. Therefore, Manhattan distance is used as a heuristic function:

$$h(n)=\left|x_d-x_n\right|+\left|y_d-y_n\right| \tag{18}$$

In Equation (18), x_d is the x-coordinate of the target point; y_d represents the y-coordinate of the target point; x_n is the x-coordinate of node *n*; y_n is the y-coordinate of node *n*.

PSO algorithm

The idea of PSO comes from the study on birds' foraging behaviour, and birds can find the best destination through collective information sharing. The advantages of PSO mainly lie in its fast convergence speed, few parameters, simple algorithm, and easy implementation. Especially for high-dimensional optimization problems, it converges to the optimal solution faster and more efficiently than GA, but PSO can easily fall into a local optimal solution. In order to solve the above problems, therefore, an improved PSO algorithm was proposed in this study to solve the problem without affecting the search speed. The local and global particle search capability was enhanced. This algorithm is more conducive to solving the path planning problem of robots.

Steps of the hybrid algorithm

The basic A* algorithm uses a heuristic evaluation function to search, which is efficient and can realize the shortest path between two points. However, when the operation scene is large, the temporal and spatial complexity of the search increase exponentially. Therefore, the integrated longicorn search algorithm must be used for global path planning, which can not only realize local planning but also be suitable for large-scale environment. The concrete steps of the improved A* algorithm and PSO to realize robot path planning are as follows:

(1) A model diagram is constructed. Obstacles are randomly distributed according to the obstacle rate, the open list and close list are defined, to which initial values are assigned, and starting nodes are put into the lists.

(2) Real number encoding is adopted for particle swarm initialization. A particle swarm consisting of n particles is defined, followed by spatial search in the raster map. Particles have position attribute x and velocity attribute v, and a fitness function is defined as the optimization objective of path planning. The position of the i-th particle at time t is expressed as follows:

$$x_t^i=(x_t^{i1},x_t^{i2},L,x_t^{iD}) \quad i=1,2,L,n \tag{19}$$

The velocity of the i-th particle at time t is:

$$v_t^i=(v_t^{i1},v_t^{i2},L,v_t^{iD}) \quad i=1,2,L,n \tag{20}$$

where t represents the current number of iterations and D denotes the number of key points searched.

In this study, the path was mainly composed of searched key points connected with line segments, so Euclidean distance was selected as the fitness function of the algorithm, but there were obstacles in the map. The obstacles that the path passed through were taken as regular terms to realize obstacle-free path planning:

$$fit(x)=\sum_{i=1}^{D}\sqrt{\left(x_i-x_{i-1}\right)^2+\left(y_i-y_{i-1}\right)^2}+\lambda\times T \quad i=1,2,L,D \tag{21}$$

where x and y represent the grid positions corresponding to the real number codes of key points; λ is a penalty factor; T refers to the number of obstacles passed through by the current path.

(3) According to the environment selection strategy designed in this study, the particle swarm was divided into two categories: the selection, crossover, and mutation operators of GA* algorithm were embedded into the first category; the chemotactic operation of the BFO algorithm was embedded into the other category, and the parallel training comprehensively strengthened the local optimization ability of the particle swarm. The specific implementation of the environmental strategy is as follows: because the population updating mode in GA follows Darwin's evolution theory of natural selection, each population evolution is guided by the environment. The updating method of the bacterial foraging algorithm is to imitate the swimming, tumbling, staying, and other movement modes of individual bacteria when looking for food, and the movement of each step is controlled by the current nutritional gradient. When the two algorithms are used to solve the same target problem, because the update strategy is affected by different environments, the environment selected by the target individual is also different. Under the same environment, some potential individuals will be lost and the possibility of optimal solution will be reduced. In this study, the environmental coefficient Y_1 of GA and the environmental coefficient Y_2 of the bacterial foraging algorithm was calculated respectively. In GA, the population will be so affected by the environment s to produce selection, crossover, and mutation. In the bacterial foraging algorithm, the population is affected by the nutritional gradient.

Therefore, the Equations of Y_1 and Y_2 can be expressed as:

$$Y_1\left(x,i\right)=\sum_{i=1}^{n} mutate_{\beta}\times select_{\alpha}\left(\sum_{m=1}^{D} x_m\right)+cross_{\delta}(1\text{-}select_{\alpha})\times(\sum_{m=1}^{D} x_m) \tag{22}$$

$$Y_2\left(x,i\right)=\sum_{i=1}^{n}\left[-d_{atact}exp\left(-w_{atact}\sum_{m=1}^{D}\left(x_m\text{-}x_{im}\right)^2\right)\right]+\sum_{i=1}^{n}[d_{repellant}exp(-w_{repellant}\sum_{m=1}^{D}(x_m\text{-}x_{im})^2] \tag{23}$$

In Equation (4), $mutate_\beta$, $select_\alpha$, and $cross_\delta$ represent the mutation, selection, and crossover probabilities of particles in GA, respectively. In Equation (23), d_{atact}, $d_{repellant}$, w_{atact}, and $w_{repellant}$ denotes the stress factors between individuals in the bacterial foraging algorithm, being the m-th element of the position attribute of the i-th particle. After the environmental coefficients of different algorithms are determined, the correlation between target individuals and different environments is solved through the Pearson correlation Equation, and individuals are subjected to optimization training according to associated environments selected. The calculation Equation for the individual environmental selection is:

$$\rho_{x,Y}=\frac{E((x\text{-}\mu_x)(Y\text{-}\mu_Y))}{\sigma_x\sigma_y} \tag{24}$$

Where μ_X and μ_Y represent the mean values of x and y, respectively; σ_x and σ_y denote the standard deviations of x and y, respectively. After the selection of the environmental strategy, the two kinds of particles are respectively subjected to selection, crossover, mutation of GA and chemotaxis of BFO. Among them, the selection, crossover, and mutation operations produce new particles as follows:

$$x_{t+1}^{i}=\eta x_t^j+\left(1\text{-}\eta\right)x_t^i;\ i,j=1,2,L,n,\ \ i\neq j \tag{25}$$

$$x_{t+1}^{j}=\eta x_t^i+\left(1\text{-}\eta\right)x_t^j;\ \ i,j=1,2,L,n,\ \ i\neq j \tag{26}$$

The new particles generated by the chemotactic operation are as follows:

$$x_{t+1}^{i}=x_t^i+C(i)\times\Phi(i)\quad i=1,2,L,n \tag{27}$$

where $C(i)>0$ represents the swimming step of forward particles; $\Phi(i)$ denotes a unit direction vector randomly selected after rotation.

(4) The two parallel particle swarm branches are fused to judge whether the optimal solution of the current particle swarm meets the set target threshold. If not, the update Equation of the improved PSO algorithm is executed. In this study, corresponding improvements were made according to the update Equation of the basic PSO algorithm. After the parallel optimization of GA and bacterial foraging algorithm, the mixed particle swarm reaches two global optimums. To integrate the optimization ability of the two algorithms, the two global optimal parameters were both introduced into the update Equation of particles, the updating of particles would be affected by G_{GA_best}, G_{BFO_best}, and P_{best}, and the improved update Equation of the PSO algorithm is as follows:

$$\begin{gathered} v_{t+1}^{i}=\omega\times v_t^i+c_1\times random(0,1)\times(P_{best\,t}^{i}-x_t^i)+c_2\times random(0,1)\times(G_{GA\_best\,t}^{i}\text{-}x_t^i)+ \\ c_3\times random(0,1)\times(G_{BFO\_best\,t}^{i}\text{-}x_t^i) \end{gathered} \tag{28}$$

$$x_{t+1}^{i}=x_t^i+v_{t+1}^{i} \tag{29}$$

where ω is the inertia weight, which decides the influence of historical velocity on the current velocity; c_1, c_2, and c_3 are learning factors; G_{GA_best} represents the global historical optimal solution obtained by genetic optimization in the hybrid particle swarm; G_{BFO_best} stands for the global historical optimal solution obtained by bacterial foraging optimization in the hybrid particle swarm; P_{best} is the historical optimal value of single particles in the search process. After particle swarm updating is completed, whether the optimal fitness function value of the particle swarm meets the set target threshold is judged again. If yes, the algorithm is ended; if not, Steps (2)-(5) are executed once again.

RESULTS

In this study, the number of picking points of fruit and vegetable picking robots in the orchard was assumed to be 40, and their position was represented by the corresponding *X*-coordinate and *Y*-coordinate values. The traveling speed of the fruit and vegetable picking robots was known, and the coordinates and picking quantity of 40 inspection points are shown in Table 1.

Table 1

Coordinates and picking quantity of picking points

Picking point	Coordinate (*X*)	Coordinate (*Y*)	Picking quantity (units)	Picking point (units)	Coordinate (*X*)	Coordinate (*Y*)	Picking quantity (units)
1	5	0	1	21	30	30	2
2	15	10	3	22	30	40	3
3	0	10	1	23	30	50	3
4	0	15	2	24	35	25	3
5	5	20	2	25	35	30	3
6	5	30	1	26	40	20	2
7	5	40	2	27	40	5	3
8	10	45	1	28	40	10	3
9	10	5	3	29	40	25	1
10	10	15	3	30	40	30	1
11	15	5	2	31	40	35	3
12	15	15	3	32	40	40	3
13	20	35	2	33	40	45	3
14	20	5	1	34	40	50	2
15	25	25	1	35	45	30	1
16	25	30	3	36	45	5	2
17	25	35	3	37	45	50	1
18	25	40	3	38	35	15	3
19	25	45	3	39	35	45	3
20	15	45	3	40	50	50	2

In this study, experiments were carried out on Intel i7 processor using Matlab2014a, and the optimization model of inspection path of agricultural robots was solved by the improved A* + PSO hybrid algorithm. The relevant parameters were set as follows: the number of particles was m=50; the inertia constant was w=0.5; the acceleration constant was c_1=c_2=c_3=2; the maximum flight velocity of particles generally did not exceed 10%-20% of the maximum velocity.

The path record table was Table=zeros (m, n); the maximum number of iterations was iter_max=100; the optimal path of each generation was Route_best=zeros(iter_max,n); the length of the best path of each generation was Length_ best =zeros (iter_max,1); the average path length of each generation was Length_ave= zeros(iter_ max,1). Simulation tests were carried out through the improved A* + PSO hybrid algorithm, respectively, and the corresponding calculation results were obtained.

In order to eliminate the influence of various random factors and verify the advantages and disadvantages of the improved A* + PSO hybrid algorithm designed in this study, the path optimization problem of fruit and vegetable picking robots was run 100 times by using the improved A* + PSO hybrid algorithm.

The convergence curve of the improved A* + PSO hybrid algorithm is shown in Fig. 2, and the optimal travel path of the agricultural inspection robots is shown in Fig. 3.

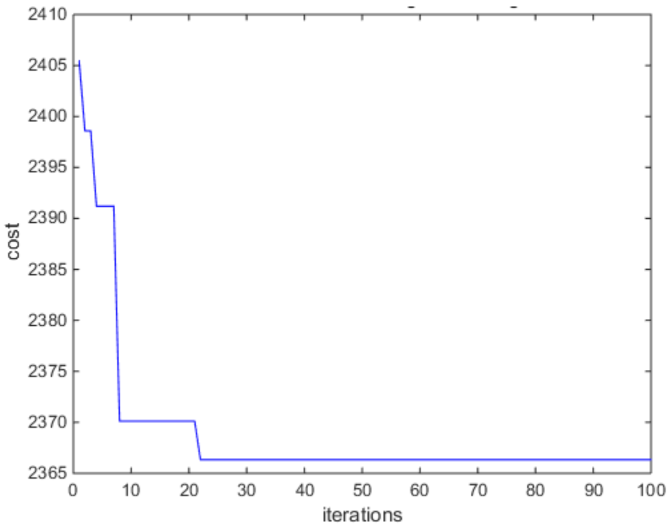

Fig. 2 - Convergence curve of improved algorithm and PSO algorithm

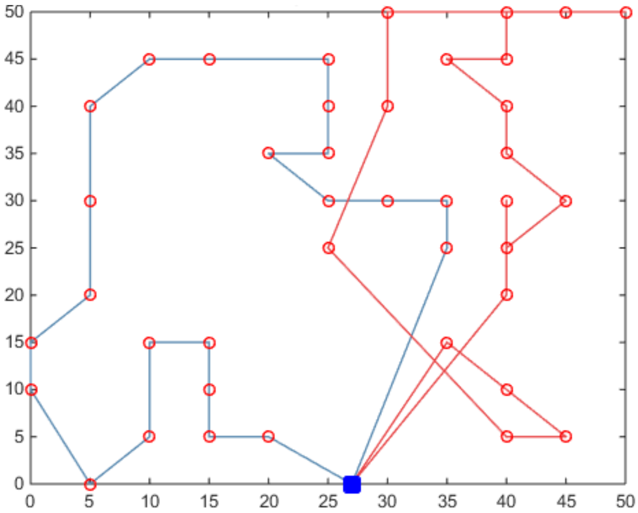

Fig. 3 - Optimal driving path of fruit and vegetable picking robots

In order to verify the effectiveness of the model and algorithm, the traditional PSO algorithm was used on the same platform, and the optimization model proposed in this study was solved with the same parameters. To achieve more scientific and effective experimental results, the maximum number of iterations of the traditional ant colony algorithm was also set to 100. The convergence curve of the traditional PSO algorithm is shown in Fig. 4, and the optimal path of fruit and vegetable picking robots is exhibited in Fig. 5.

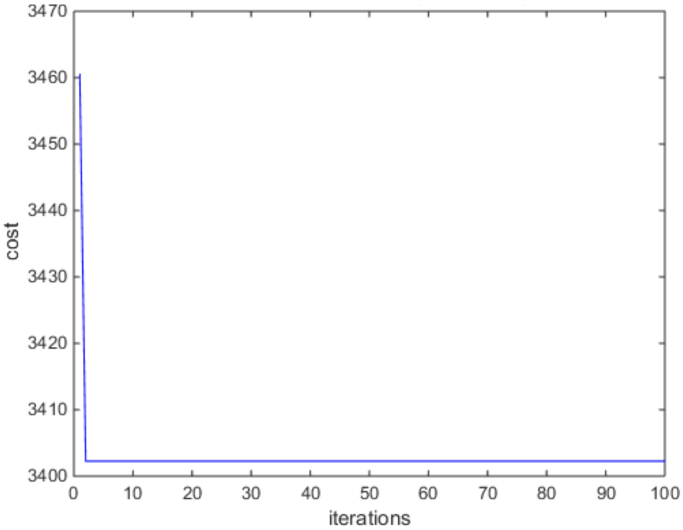

Fig. 4 - Convergence curve of the traditional PSO algorithm

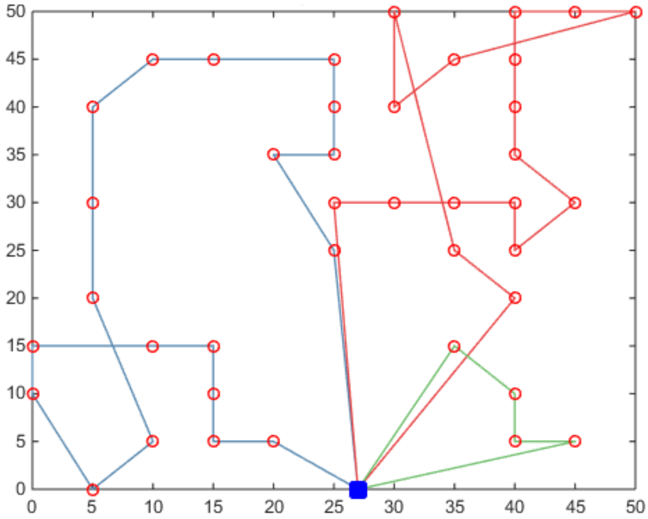

Fig. 5 - Optimal driving path of fruit and vegetable picking robots

The improved A* + PSO hybrid algorithm was compared with the traditional PSO algorithm in inspection path, total distance travelled by fruit and vegetable picking robots, and convergence time of the algorithm, and the comparison results are listed in Table 2.

Table 2

Comparison of improved A*+ PSO algorithm and traditional PSO algorithm

Algorithm	Inspection path	Robot	Distance (m)	Convergence time (s)	Total cost (yuan)
Improved A*+ PSO algorithm	0→14→11→2→12→10→9→1→3→4→5→6→7→8→20→19→18→17→13→16→21→25→24→0	2	366.34	100.17	2366
	0→26→30→29→35→31→32→39→33→34→40→37→23→22→15→27→36→28→38→0				
Traditional PSO algorithm	0→14→11→2→12→10→4→3→1→9→5→6→7→8→20→19→18→17→13→15→0	3	393.41	118.26	3402
	0→16→21→25→30→29→35→31→32→33→34→37→40→39→22→23→24→26→0				
	0→38→28→27→36→0				

It could be seen from Table 2 that the optimization ability and convergence of both A* algorithm and PSO algorithm were stronger than those of the traditional PSO algorithm. From the convergence curve of the algorithm, it could be intuitively seen that in terms of the driving distance of picking agricultural robots, the optimal path length obtained by the improved A* algorithm and particle swarm optimization algorithm was better than that obtained by the traditional ant colony algorithm. Compared with the traditional GA, the total inspection distance acquired through the improved ant colony algorithm was shortened by 27.07 m with a reduction rate of 6.88%. From the number of picking robots, the improved A* + PSO hybrid algorithm used 2 picking robots, while the PSO algorithm needed three; as for the convergence time, the traditional PSO algorithm spent longer convergence time than the improved A* + PSO hybrid algorithm. The convergence time of the improved A* + PSO hybrid algorithm was 18.09 seconds shorter (by 15.30%) than that of the traditional PSO algorithm.

CONCLUSIONS

In order to improve the efficiency of path planning and the accuracy of global coverage planning in the process of automatic navigation of picking robots, the improved A* + PSO hybrid path planning method was introduced into the path planning model of picking robots, and the efficiency of path planning was improved by optimizing the distance of path planning. The path planning algorithm of picking robots was simulated under the planting environment of large-scale plots.

Aiming at the multi-objective and batch picking path planning problem of picking robots in fruit and vegetable gardens, a three-dimensional path planning model that could pick Units in the order in turn was established by constructing three-dimensional spatial coordinates in this study and then solved by designing the improved A* + PSO hybrid algorithm. The experimental results show that, compared with other heuristic algorithms, the improved A* + PSO hybrid algorithm has a faster convergence speed under the same parameters, which can effectively prevent the algorithm from falling into the local optimum and get a solution with higher accuracy. In conclusion, the improved A* algorithm + PSO hybrid algorithm can effectively improve the path search accuracy of the PSO algorithm and then improve the efficiency of picking robots.

To sum up, the picking robot model and the improved A* + PSO hybrid algorithm constructed in this study reach a good fitting effect in fruit and vegetable picking, which can effectively improve the overall fruit and vegetable picking efficiency. In this study, however, only the order picking path planning under the single-instruction picking background of picking robots was investigated, that is, the picking robots started from the starting point, moved along the planned path to the row and column where the fruits and vegetables were to be picked, and returned to the starting point after picking the fruits and vegetables in batches. Therefore, the path planning of picking robots in multi-instruction operations will be deeply explored in the future.

REFERENCES

[1] Cao X., Zou X., Jia C., Chen M., Zeng Z. (2019), RRT-based path planning for an intelligent litchi-picking manipulator, *Computers and Electronics in Agriculture*, Vol. 156, ISSN 0168-1699, pp. 105-118, England;

[2] Chen M., Zhang H., Wang X., Li X. (2020), Design of apple picking robot based on iterative learning PID algorithm (基于迭代学习 PID 算法的苹果采摘机器人设计), *Journal of Agricultural Mechanization Research*, Vol. 42, no. 6, ISSN 1003-188X, pp. 83-86, China;

[3] Grau J., Grosse I., Keilwagen J. (2015), PRROC: computing and visualizing precision-recall and receiver operating characteristic curves in R, *Bioinformatics*, Vol. 31, no. 15, ISSN 1367-4803, pp. 2595-2597, England;

[4] Gao H., Wang H., Chen J. (2014), Research and design of kiwi fruit picking robots (猕猴桃采摘机器人的研究与设计), *Journal of Agricultural Mechanization Research*, Vol. 29, no. 2, ISSN 1003-188X, pp.75-79, China;

[5] Ji C., Feng Q., Yuan T., Tan Y., Li W. (2011), Development and performance analysis on cucumber harvesting robot system in greenhouse (温室黄瓜采摘机器人系统研制及性能分析), *Robot*, Vol. 33, no. 6, ISSN 1002-0446, pp. 726-730, China;

[6] Li S, Huang N. (2019), High precision control simulation of spherical fruit picking robot (球形水果采摘机械手的高精度控制仿真), *Computer Simulation*, Vol. 36, no. 10, ISSN 1006-9348, pp. 302-306, China;

[7] Liu J., Li P., Ni Q., Li Z. (2010), Design and test of the vacuum suction device for tomato harvesting robot (番茄采摘机器人真空吸盘装置设计与试验), *Transactions of the Chinese Society for Agricultural Machinery*, Vol. 41, no. 10, ISSN 1000-1298, pp. 170-173, 184, China;

[8] Meng B H., Godage I S., Kanj I. (2022), RRT∗-Based path planning for continuum arms, *IEEE Robotics and Automation Letters*, Vol. 7, no. 3, ISSN 2377-3766, pp. 6830–6837, United States;

[9] Mondal S C., Marquez P L C., Tokhi M O. (2021), Analysis of mechanical adhesion climbing robot design for wind tower inspection, *Journal of Artificial Intelligence and Technology*, Vol. 1, no. 4, ISSN 2766-8649, pp. 219-227, United States;

[10] Masumori A., Sinapayen L., Maruyama N., Mita T., Ikegami T. (2020), Neural autopoiesis: organizing self-boundary by stimulus avoidance in biological and artificial neural networks, *Artificial Life*, Vol. 2020, no. 4871, ISSN 1064-5462, pp. 1-22, United States;

[11] Ma N., Wang J., Meng Q H. (2022), Conditional generative adversarial networks for optimal path planning, *IEEE Transactions on Cognitive and Developmental Systems*, Vol. 14, no. 2, ISSN 2379-8920, pp. 662-671, United States;

[12] Orozco-Rosas U., Montiel O., Sepúlveda R. (2019), Mobile robot path planning using membrane evolutionary artificial potential field, *Applied Soft Computing*, Vol. 77, ISSN 1568-4946, pp. 236-251, Netherlands;

[13] Orozco-Rosas U., Picos K., Pantrigo J.J., Montemayor A.S., Cuesta-Infante A. (2022), Mobile robot path planning using a QAPF learning algorithm for known and unknown environments, *IEEE Access*, Vol. 10, ISSN 2169-3536, pp. 84648-84663, United States;

[14] Paul S., Arunachalam A., Khodadad D., Andreasson H., Rubanenko O. (2021), Fuzzy tuned PID controller for envisioned agricultural manipulator, *International Journal of Automation and Computing*. Vol. 18, no. 4, ISSN 2731-538X, pp. 568-580, China;

[15] Peng J., Gao X., Long H., Zhou C. (2021), A parameter measurement method for autonomous-rail rapid tram based on four-view stereo vision (基于四目立体视觉的智轨电车参数测量方法与系统), *Journal of Chongqing University*, Vol. 44, no. 4, ISSN 1000-582X, pp. 29-36, China;

[16] Ping X., Liu Y., Dong X., Zhao Y., Zhang Y. (2017), 3-D reconstruction of texture less and high-reflective target by polarization and binocular stereo vision (基于偏振双目视觉的无纹理高反光目标三维重构), *Journal of Infrared and Millimeter Waves*, Vol. 36, no. 4, ISSN 1001-9014, pp. 432-438, China;

[17] Qureshi A H., Miao Y., Simeonov A., Yip M C. (2021), Motion planning networks: bridging GAp between learning-based and classical motion planners, *IEEE Transactions on Robotics*, Vol. 37, no. 1, ISSN 1552-3098, pp. 48-66, United States;

[18] Sun J., Liu G., Tian G., Zhang J. (2019), Smart obstacle avoidance using a danger index for a dynamic environment, *Applied Sciences*, Vol. 9, no. 8, ISSN 1454-5101, pp. 1589, Romania;

[19] Su Y., Yang L., Song X. (2016), Design and experiment of intelligent mobile apple picking robot (智能移动苹果采摘机器人的设计及试验), *Journal of Agricultural Mechanization Research*, Vol. 2016, no. 1, ISSN 1003-188X, pp. 159-162, China;

[20] Tian Y., Zhu X., Meng D., Wang X., Liang B. (2021), An overall configuration planning method of continuum hyper-redundant manipulators based on improved artificial potential field method. *IEEE Robotics and Automation Letters*, Vol. 6, no. 3, ISSN 2377-3766, pp. 4867-4874. United States.

[21] Wang H., Mao W., Liu G., Hu X., Li S. (2012), Identification and location of fruit and vegetable robot based on visual combination (基于视觉组合的苹果作业机器人识别与定位), *Transactions of the Chinese Society for Agricultural Machinery*, Vol. 43, no. 12, ISSN 1000-1298, pp. 165-170, China;

[22] Wu K., Wang H., Esfahani M A., Yuan S. (2020), Achieving real-time path planning in unknown environments through deep neural networks, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 23, no. 3, ISSN 1524-9050, pp. 1-10, United States;

[23] Xu M., Deng R. (2014), Comparison of domestic and foreign facilities in agricultural development (国内外设施农业发展的比较), *Journal of Beijing University of Agriculture*, Vol. 29, no. 2, ISSN 1002-3186, pp. 75-79, China.