applied sciences

Article

Research on Complete Coverage Path Planning of Agricultural Robots Based on Markov Chain Improved Genetic Algorithm

Jiangyi Han *, Weihao Li, Weimin Xia and Fan Wang

School of Automotive and Traffic Engineering, Jiangsu University, Zhenjiang 212013, China; 118307101798@163.com (W.L.); xiawm@gmail.com (W.X.); wangfan9026@163.com (F.W.)
* Correspondence: hjy0306@ujs.edu.cn; Tel.: +86-139-5140-8303

Featured Application: This study can be used for navigation operations of agricultural equipment or vehicles in farm environments.

Abstract: Due to the limitations of low coverage, high repetition rate, and slow convergence speed of the basic genetic algorithm (GA) in robot complete coverage path planning, the state transition matrix of the Markov chain is introduced to guide individual mutation based on the genetic mutation path planning algorithm, which can improve the quality of population individuals, enhancing the search ability and convergence speed of the genetic algorithm. The proposed improved genetic algorithm is used for complete coverage path planning simulation analysis in different work areas. The analysis results show that compared to traditional genetic algorithms, the improved genetic algorithm proposed in this paper reduces the average path length by 21.8%, the average number of turns by 6 times, the repetition rate by 83.8%, and the coverage rate by 7.76% in 6 different work areas. The results prove that the proposed improved genetic algorithm is applicable in complete coverage path planning. To verify whether the Markov chain genetic algorithm (MCGA) proposed is suitable for agricultural robot path tracking and operation, it was used to plan the path of an actual land parcel. An automatic navigation robot can track the planned path, which can verify the feasibility of the MCGA proposed.

Keywords: agricultural robots; complete coverage path planning; Markov chain; improve genetic algorithm

Citation: Han, J.; Li, W.; Xia, W.; Wang, F. Research on Complete Coverage Path Planning of Agricultural Robots Based on Markov Chain Improved Genetic Algorithm. *Appl. Sci.* **2024**, *14*, 9868. https://doi.org/10.3390/app14219868

Academic Editor: Luis Payá

Received: 1 October 2024
Revised: 21 October 2024
Accepted: 24 October 2024
Published: 28 October 2024

1. Introduction

The development of precision agriculture and intelligent agriculture in agricultural production has stimulated people's interest and research in artificial intelligence and robotics technology. Agricultural robots can replace humans in agricultural production activities such as farming, spraying pesticides, fertilizing, and harvesting, which can deal with the challenge of labor shortages, reduce health risks in agricultural production, and save time and energy [1]. Agricultural robots need to perform various tasks and operate in complex environments. Compared to robot paths in the industrial field, agricultural robot operations require paths that can complete full coverage of the farm area. Considering the impact of the robot's operations on crop growth, a low path repetition rate is required. Considering the operational efficiency of the robot, the number of turns should be reduced in the path. Thus, the complete coverage path planning algorithm is a key issue and hot topic in the automatic operation of agricultural robots [2]. Gonzalez et al. [3] proposed an extended BSA algorithm based on the traditional Backtracking Spiral Algorithm (BSA), which can be extended to most grid-based overlay algorithms. Reciprocating and spiral coverage are simple and intuitive, but often limited by specific environments and assumptions, lacking universality and flexibility. Zhu et al. [4] adopted a combination of bio-inspired neural networks and grids for complete coverage path planning of underwater robots, which can successfully avoid dynamic and static obstacles, but the mode was more complex and the calculation amount of the algorithm was very large. Li et al. [5] utilized the improved A* (A-Star) algorithm for complete coverage path planning of the sweeping robot, which effectively enhanced the operating efficiency of the algorithm and reduced the length and repetition rate of the path planning. De Carvalho et al. [6] studied cleaning robots and proposed a template-based method to control path execution, integrating the kinematics of mobile robots into the path planning process. However, the template method relies on preset path patterns, lacks flexibility, and is difficult to adapt to complex or irregular environments, leading to low coverage efficiency. To solve the problem of bulldozers easily falling into dead zones during unmanned operations, Rao Li et al. [7] added machine operation rules to the bio-inspired neural network algorithm to reduce path repetition rate, and used the A * algorithm to help bulldozers quickly escape when the machine is stuck in dead zones. Gaogao Shang et al. considered the height variation and energy consumption factors in the full coverage path planning for horticultural tractors in mountainous areas [8]. Selek et al. [9] proposed a traversal algorithm that completely covers the work areas based on the spanning tree coverage algorithm. However, this method has a high repetition rate and a high number of robot path turns; Le et al. [10] proposed a low repetition rate complete coverage path planning algorithm using an ant colony algorithm. However, the total path planned by this traversal algorithm is relatively long, and it is prone to falling into dead zones when encountering areas with dense obstacles.

The genetic algorithm (GA) is a well-known, meta-heuristic technique for addressing static mobile robot global path planning (MRGPP) issues [11]. However, genetic algorithms have certain shortcomings, such as a simple evolutionary process, inefficient population initialization, and a lack of prediction for the trend of random event changes. Therefore, robots with different job requirements will improve their path planning using GA based on specific operation requirements and environments. In automated guided vehicles (AGV) path planning by GA, Qihao Liu et al. [12] proposed integrating the shop scheduling scheme, operation sequencing, and transportation task assignment of vehicles into genetic factor coding, and designed a neighborhood searching strategy for critical paths to improve the effectiveness of local path search on AGV. In autonomous underwater vehicle (AUV) path planning, Kun Hao et al. [13] proposed an adaptive GA for AUV global path planning, which is used to optimize the global path based on the collision detection mechanism. Yang Ning et al. [14] proposed a path planning cost function based on the ocean current model to improve GA for the shortest time required to guide AUVs. In autonomous mobile robot path planning, Mohd N. Ab Wahab et al. [15] proposed an enhanced GA, a linear rank-based, or clearance-based, probabilistic road map (CBPRM) technique that overcomes these shortcomings. The new model guides the population initialization process by using the fitness score of each cell in the environment, lowering the number of infeasible pathways created. Zarevich et al. [16] proposed a pathfinding optimization process using GA, and its evaluation criteria for the path are path length and travel time. Yu-Ju Chen et al. [17] opposed a path planning algorithm based on the Markov decision process, which is used for the point-to-point path planning of wheeled robots.

Among the existing agricultural robot or unmanned vehicle full coverage path planning algorithms, the template method, bio-inspired neural network algorithm, and GA have advantages, but shortcomings limit their applicability and efficiency in different operating areas. In the application of GA, the traditional GA performs well in various optimization problems, but has limitations in solving complete coverage path planning problems, such as slow convergence speed, easily getting stuck in local optima, sensitive parameter settings, and low efficiency when dealing with large work areas. Therefore, based on the traditional GA, the paper introduced the state transition matrix of Markov chains that can constrain random methods to guide population individuals to mutate, thereby improving the individual quality of the population, which will enhance the search ability and convergence speed of the algorithm, and improve the applicability and efficiency of path planning for different work areas.

Section 2 is materials and methods including the work area modeling analysis, the algorithm improvement based on Markov chain, the path evaluation function analysis, and the design of the path planning algorithm. Section 3 is the simulation analysis and field experiments. Section 4 is the conclusion.

2. Materials and Methods

This section mainly includes the modeling method for the work area, the improvement of the genetic algorithm based on Markov chain, the analysis of the path evaluation function, and the design of complete coverage path planner.

2.1. Modeling Method for Work Area

The grid map method is used to establish a map model of the work area [18]. The simplicity and intuitiveness of raster maps make the representation and understanding of spatial data easy and especially suitable for detailed analysis and visualization of spatial variables such as terrain and landforms. Meanwhile, the regularity and uniformity of raster data structures make the storage, retrieval, and processing of spatial data more efficient. In addition, the grid method can flexibly control the accuracy of modeling by adjusting the grid size. The steps for using the grid method for environment modeling are as follows:

I. Calculate the length and width of the work area based on the coordinates of the boundary points of the work area.
II. Using the operating width of agricultural robots as the grid size, divide the map into multiple grids of the same size.
III. Represent the position of each grid in latitude and longitude coordinates in the form of Bi, Li.
IV. Define the properties of the grid: boundary or obstacle grid (impassable area), unassigned grid, and worked grid.

2.2. Improved Genetic Algorithm Based on Markov Chain

John Holland proposed the GA in the early 1970s. The algorithm simulates natural selection, replication, crossover, and mutation in biological evolution, starting from the initial population and gradually evolving into individuals that are more suitable for the environment. Through continuous reproduction and iteration, the algorithm eventually stabilizes and finds the optimal or approximate solution. However, the algorithm can not predict the trend of random event changes. In the path planning of agriculture robots, the convergence speed of algorithms is slow, resulting in a high repetition rate of agricultural operation paths and low coverage of operations.

The state sequence and transition probability matrix of the Markov chain is used to predict the future state and change the trend of random events [19]. It is a stochastic process that describes the transition from one state to another in the state space. In other words, the probability of the state changing to X at step $n + 1$ depends only on the state at step n and does not depend on the states of all steps before step n. The mathematical definition of a homogeneous Markov chain is as follows:

If $\{X(n), n = 0, 1, 2, 3, \ldots\}$ is a discrete state space and has non-negative parameters (state space is I), then it satisfies the following equation:

$$\begin{aligned} \mathrm{P}_{ij} &= (X(n+1) = j | X(0) = i_0, X(I) = i_1, \ldots, X(n-1) = i_{n-1}, X(n) = i) \\ &= P\{X(n+1) = j | X(n) = i\} \end{aligned} \quad (1)$$

In Equation (1), $\{X(n)\}$ is a homogeneous Markov chain, P is the probability of state transition, and $X(n)$ is the state at time point n.

Based on the current state, the Markov chain prediction predicts future states by the state transition probability matrix. Therefore, the state transition probability matrix is the core of the Markov chain prediction. The mathematical expression of the state transition probability matrix is written as follows:

Let $X = \{Xn, n \geq 0\}$ be a homogeneous Markov chain. The one-step transition probability matrix is $P = (P_{ij})_{i,j \in S}$;

$$P_{ij} = P(X(n+1) = j | X_n(0) = i) = P(X_I = j | X_0 = i) \quad (2)$$

Thus,

$$P_{ij} \geq 0, \sum\nolimits_{j \in S} P_{ij} = 1 \quad (3)$$

Equations (2) and (3) are the fundamental properties of transition probability matrices.

As shown in Figure 1, the individuals of the population are the set of path points from the starting point to the target point, and the decimal grid number set is used as the encoding form of the individuals. For example, an encoded possible solution set can be represented as: {1, 2, 3, 6, 5, 4, 7, 8, 9, 12, 11, 10}, where 1–10 are all grid numbers.

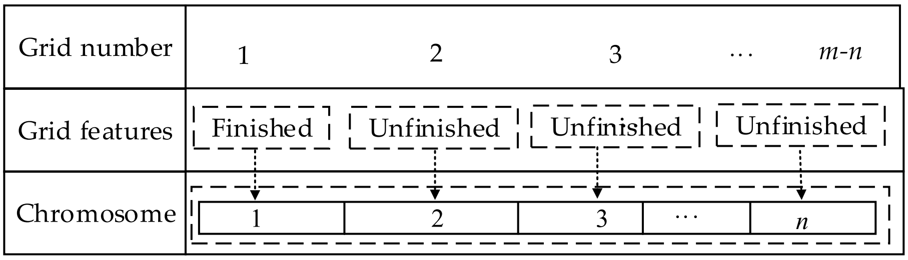


Figure 1. Gene coding method.

Basic genetic algorithms [20] generate new solutions through gene crossover. Here, the state transition matrix in the Markov chain is proposed to guide the changes in gene sequences (solution vectors). The process simulates the mutation and selection process in genetic algorithms, using constrained random methods to generate new individuals. Assuming the initial solution set is {1, 2, 3, 6, 5, 4, 7, 8, 9, 12, 11, 10}, the process randomly selects a location to slice the solution set and removes the latter half of the solution set. For example, choosing to slice backward at grid position 4 results in a solution set of {1, 2, 3, 6, 5, 4, □, □, □, □, □, □}. Starting from the fourth grid at the end of the sliced solution set, it generates subsequent grid numbers as the new solution set. Starting from grid 4, the reachable grid is {1, 2, 5, 8, 7}. Since grid 5 has already been traversed, to reduce coverage, the possibility of retroversion grid 5 will not be considered. Therefore, the path reachable by grid 4 is {1, 2, 8, 7}.

As shown in Figure 2, when the robot moves from grid 1 to grid 4 and its current position is grid 4, to reduce the repetition rate of path planning and improve navigation efficiency, the specific work rules were set during the calculation of the state transition matrix: (i) The transition probability from grid 4 to grid 1 is set to 0, which can prevent the robot from returning to the traversed position; (ii) the transition probability from grid 4 to itself is also set to 0, which can ensure that the robot does not stagnate at its current position; (iii) the transition probabilities to the other four reachable grids are set to be equal value, all at 0.25, to ensure that the robot traverses the other areas uniformly.

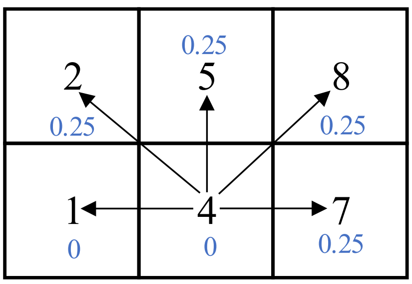


Figure 2. Path probability distribution.

Define the value $P(i,j)$ of matrix P at a certain position $P(j/i)$, which is the probability of transitioning from state i to state j. Therefore, the state transition matrix is:

$$P = \begin{bmatrix} 0 & 0.25 & 0.25 & 0 & 0.25 & 0.25 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4}$$

Since the state transition matrix of the Markov chain converges to a stable probability distribution independent of the initial state probability distribution, the probability matrix is as follows:

$$P = \begin{bmatrix} 0 & 0.25 & 0.25 & 0 & 0.25 & 0.25 \end{bmatrix} \tag{5}$$

The probability matrix generates the next reachable region based on the probability distribution of the transition matrix. For example, if the next arrived area is grid 8, then the current solution set is {1, 2, 3, 6, 5, 4, 8, □, □, □, □, □}. starting from grid 8, the next reachable area is generatetd, and this process is repeated until all grids have been traversed.

2.3. Design of Path Fitness Function

In the iterative process of the genetic algorithm, numerous individuals will be generated. To select the optimal solution, it is necessary to design a reasonable fitness function to assess the performance of individual paths. The fitness function can evaluate the efficiency and applicability of each path. Considering the requirements of agricultural robots for paths, the evaluation factors proposed include the coverage rate, repetition rate, smoothness of the paths, and operation time. The path selected through the fitness function can not only maximize the efficiency of the task but also take into account the smoothness of the path and the task time. The designed fitness function is as follows:

$$F(x) = w_1 \cdot \frac{A_h \bigcap \{U_i S^i\}}{A_h} + w_1 \cdot \frac{N_r}{N_p} + w_3 \cdot \int_0^{t_f} \theta^2(t) dt + w_4 \cdot N_S \tag{6}$$

where, the $A_h \bigcap \{U_i S^i\} / A_h$ is the coverage rate of the path, S^i is the ith path, the $\{U_i\ S^i\}$ is the set of paths, and A_h is the total operation area. The N_r / N_p is the repetition rate of path, the N_r is the grids of the repeated operation, and the N_p is the completed operation grids of the whole field. The $\int_0^{t_f} \theta^2(t) dt$ is the smoothness of the path, and the position of the rotation angle of the steering wheel of the robot is related to the smoothness of the path. The N_s is the number of turns. The w_1~w_4 represents the weight coefficients of each evaluation indicator. Different types of target requirements can be met by adjusting the size of the weight coefficients.

In fitness function (1), the dimensions of coverage and repetition rate functions are between [0–1]. However, the dimensions of the path smoothness function are relatively large, so data normalization is required for comparison or comprehensive evaluation. There are three common normalization methods: the maximum normalization method, the standard score normalization method, and the median normalization method [18]. Due to the fixed range of the front wheel steering angle for agricultural robots ($-40°$~$40°$), the maximum normalization method is adopted as follows:

$$\mathrm{J}_{norm} = (J - J_{min}) / (J_{max} - J_{min}) \tag{7}$$

where, J_{min} and J_{max} are the minimum and maximum values of the path smoothness within the predetermined evaluation range, respectively. In order to generate a safe, efficient, and trackable path that satisfies the kinematic constraints of the robot, a cubic B-spline curve [21] is used to smooth the path. The curvature k of each path point in the turning path segment is obtained through sampling, and the maximum curvature $k_{\max}$ of each path point is constrained. Therefore, as long as each path point meets the requirements, the entire path must satisfy the constraint conditions, and the final generated path sequence can satisfy the kinematic constraints of the robot and have continuity.

In the fitness function design, this paper does not use the method of gene crossover to generate new solutions based on the traditional GA. Instead, it employs the state transition matrix in a Markov chain to guide the evolution of gene sequences (i.e., solution vectors). This process simulates the mutation and selection mechanisms in genetic algorithms, using a constrained random method to generate new individuals. The pseudocode for the fitness function is shown in Algorithm 1 below.

Algorithm 1: Pseudocode for Fitness Function Design

Input: *Ga_x*, *marko_p*, Cell_Location
Output: Fitness Function Value
Initialize parameter settings
  Set the weight of the objective function: w_1, w_2
  Set the values for path smoothness: J_{max}, J_{min}
  Calculate path coverage and repetition rate
  N←*Ga_x* Number of rows (Population size)
  M←*Ga_x* Number of columns (Gene length)
For individual I, $i \in N$ in N
  temp←$Ga\_x[i : N - 1]$
  *value*1←Coverage←Number of no-repeat elements (temp)
  *value*2←Repetition←Number of no-repeat elements (temp) −M + 1
If $Ga\_x[i, N] \neq 1$
  Initialization steps step = 1
While *temp_marko_p*(1,*Ga_x*(*i*, end)) = 0
  Using *marko_p* update *temp_marko_p*
  Step count increases automatically
End
  Coverage←Coverage—setp
End if
  Calculate path smoothness
  Calculate the vectors of two adjacent vectors
  Calculate the angle between two vectors *value*3←Path_smoothness←$\int_0^t \theta dt$
If last grid of the path sequence is not the starting point
  Calculate the norm of the position and starting point vectors, and update the smoothness of the path
End if
  value←$w_1 \times value1 + w_2 \times value2 -$ (value3 $- J_{min})/(J_{max} - J_{min})$
  Return value
End

2.4. Design of Complete Coverage Path Planner

Based on the basic GA, the state transition matrix (6) of the Markov chain is used to guide the mutation of the gene sequence, generating new individuals through random and guided methods. The pseudocode of the algorithm is shown in Algorithm 2.

Algorithm 2: Improved Genetic Algorithm Based on Markov Chain

Input: *field_length*, *field_width*, robot length: *length*, operation width: *swath*, Initial population size: *N*, maximum number of iterations: *max_iter*
Output: Optimal path sequence, Optimal individual fitness value
  Grid the field and calculate the center point of the grid
For each x from 0 to field_length in step *length*
For each y starts from 0 to field-width in step *swath*

Algorithm 2: *Cont.*

Calculate and store the coordinates of each grid
Establish solution set *Ga_x*
For each grid
If i is an odd row
Number in order
Else
Number in reverse order
Generate distance matrix *marko_r*
For i ranges from 1 to the total numeber of grids
For j ranges from 1 to the total number of grids
If $i \neq j$
Calculate the distance between grid i and j
Save to *marko_r* [i, j]
Generate state transition matrix *marko_p*
For i range each grid
Find the grid j with the minimum distance
Calculate transition probability based on minimum distance
Save to *marko_p* [i, j]
Optimizing paths using Markov chains
For $iter \leftarrow [i : max\_iter]$
For Every individual i
Generate new gene sequences using *marko_p*
Calculate the fitness value of a new gene sequence
If Fitness of the new gene sequence > Current optimal fitness
Update the optimal fitness value and optimal path sequence
Determine and output the optimal solution
End

The algorithm is executed once each time the robot performs a task, meaning the algorithm runs once per task. The sampling accuracy of the path points is set to 0.01 m.

3. Simulation Analysis of Path Planning Algorithm Based on Improved Genetic Algorithm

Based on the above analysis, a complete coverage path planning simulation platform was developed using MATLAB's APP Designer toolbox (version 2020). The platform can achieve the following functions: visual display of the full coverage paths and output evaluation indicators, such as coverage rate, repetition rate, number of turns, path length, etc.

3.1. Simulation System

The simulation platform interface is shown in Figure 3. The simulation interface consists of four parts: the path visualization window, input module, output module, and start button; The path visualization window is used to display the paths planned by the complete coverage path planning algorithm within the work area. The input module is used to set simulation parameters. The output module can output the results of path planning. The start button is used to start running the simulation system.

In the simulation, path planning is performed on multiple lands of different areas that can verify the adaptability of the proposed full coverage path planning algorithm in handling work areas of different areas. The lands are 1400 m^2, 500 m^2, 150 m^2, 30 m^2, 18 m^2, and 8 m^2 respectively. Due to the operating width of the agricultural robot being 1 m, the areas of 70 × 20 m, 50 × 10 m, and 30 × 5 m can all be considered large-scale operating areas. For the convenience of discussion, the improved genetic algorithm based on the Markov chain is abbreviated as MCGA, and the basic genetic algorithm is abbreviated as GA.

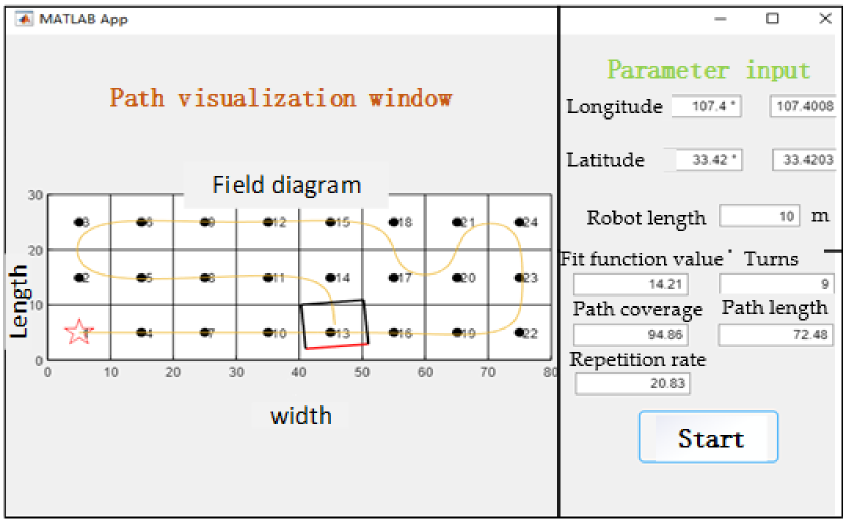


Figure 3. GUI of complete coverage path planning. The ☆ is the entrance, and the yellow line indicates the planned path. Black dots indicate the grid that has been traversed.

3.2. Simulation Experiment and Analysis

In simulation, the weight coefficients of the fitness function (1) need to be determined. The initial values of the weight coefficients are set based on experience, with the main goal of improving coverage and reducing repetition rate. The weights of the smoothness of the path and the steering are assigned as 0.1, which improves the smoothness of the path and work efficiency. By observing the impact of different weight coefficients on simulation results, the weights are gradually adjusted to ensure that the weight coefficients fully reflect the importance of different performance indicators. The final weighting coefficients are work coverage rate weight w_1 = 0.7, repetition rate weight w_2 = 0.1, path smoothness weight w_3 = 0.1, and number of turns weight w_4 = 0.1. The simulation parameters set system are shown in Table 1.

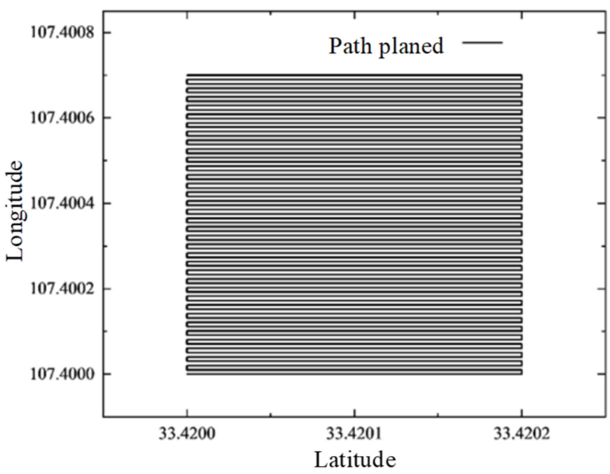
Table 1. Simulation parameters of complete coverage path planning.

Parameter	Value
Operation width	1 m
Length	70 m
Width	20 m
Minimum turning radius	1.9 m
Initial population size	2500
Maximum number of iterations	1000
Path point sampling rate	0.01 m

The coordinates of the boundary points of the work area in the simulation are A (33.42,107.4), B (33.4202,107.4), C (33.4202,107.4007), and D (33.42,107.4007). A total of 1400 grids were generated in this simulation, and all the above parameters were inputed into the full coverage path planning simulation platform.

The path planning results are shown in the Figure 4a,b displays the optimal fitness values for each generation during the iteration process of MCGA and GA. In Figure 4b, it can be seen that the initial fitness value of MCGA is higher than that of GA, which is due to the guiding role of using Markov chains in the initial population generation process. By selecting initial individuals with high fitness values based on the transition probabilities between states, the quality of the solution set can be improved. Secondly, after 393 iterations, the fitness function value of MCGA reached the optimal value of 11.8

and remained stable in subsequent iterations. The fitness function value of GA reached its optimal value of 11.63 after 980 iterations, thus proving that MCGA has significantly higher efficiency in finding the optimal solution over GA, and has a faster convergence speed and higher fitness level.



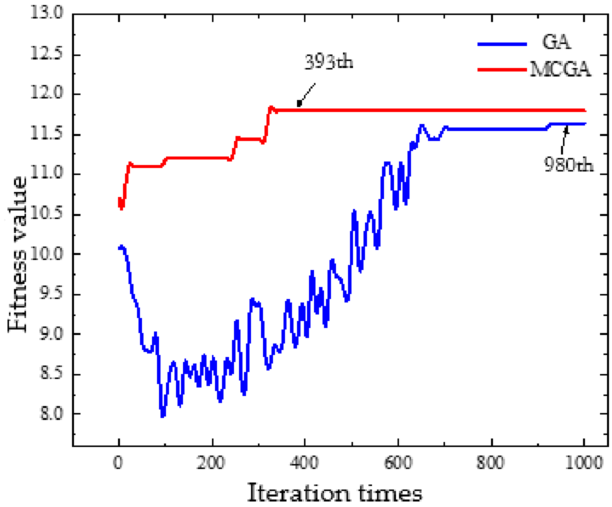


Figure 4. Path planning results of MCGA: (**a**) Path planning results; (**b**) Fitness value iteration curve.

The path planning for other field size regions has also been executed by GA and MCGA, and the important results of path planning for MCGA and GA algorithms are shown in Table 2. Compared to GA, the path length planned by MCGA is reduced by an average of 21.8%, the number of turns is reduced by an average of 6 times, the repetition rate is reduced by 83.8%, and the coverage rate is increased by 7.76%. These results show that the proposed approach significantly improves the quality of complete coverage path planning by combining Markov chains with genetic algorithms.

Table 2. Comparison of algorithm effectiveness between MCGA and GA.

Field Size	Pathlength (m)		Number of Turns		Repetition (%)		Coverage Rate	
	GA	MCGA	GA	MCGA	GA	MCGA	GA	MCGA
70 × 20 m	1689.3	1507.2	96	82	23.1	6.707	85.63%	96.42%
50 × 10 m	615.2	499.4	35	28	10.6	3.7	87.52%	98.35%
30 × 5 m	182.7	136.1	11	7	8.3	2.6	90.5%	99.13%
10 × 3 m	54.6	43.3	9	5	6.5	0.8	96.4%	99.6%
6 × 3 m	28.5	20.3	6	4	3.4	0	99.4%	100%
4 × 2 m	12.5	9.3	4	3	2.2	0	100%	100%
Improve	21.8%		6		83.8%		7.76%	

3.3. Field Experiment

In order to verify whether the MCGA proposed is suitable for agricultural robot path tracking and operation, it is next used to plan the path of an actual land parcel, and an automatic navigation robot is used to track the planned path, which can verify the performance of the proposed MCGA. A suitable agricultural robot experimental system is built for tracking the planned path. The agricultural robot experimental system adopts car-like robots. As shown in Figure 5a, the chassis of the robot (Agilent Robotics Company, HUNTER-SE, Santa Clara, CA, USA) is an electrically rear wheel drive with front wheel electrically steering. The wheelbase of HUNTER-SE is 550 mm, the track width is 520 mm, its length is 820 mm, the width is 640 mm and the work width is 1 m. The minimum turning radius is 1.9 mm. As shown in Figure 5b, a navigation system (Shanghai Lianshi, RTK-GPS, Shanghai, China, positioning accuracy: 2.5 cm), radar sensor (Shenzhen SLAMTEC company, RPLidaR-A1, Shanghai, China), and IMU inertial sensor (Shenzhen WIT, HWT101CT-232, Shenzhen, China) are installed on the HUNTER-SE.

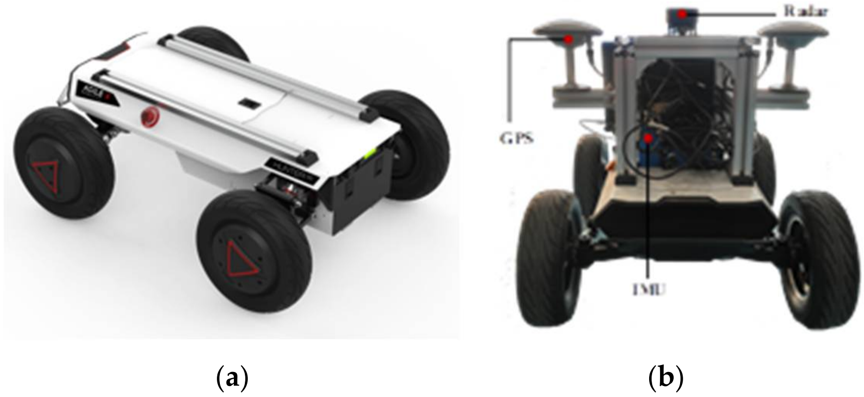


Figure 5. Robot platform construction: (**a**) Robot chassis; (**b**) Robot with GPS navigation system.

After the automatic navigation robot platform is built, the Jiangsu University playground is selected as the experiment site for path planning and path-tracking experiments. As shown in Figure 6, the blue box represents the experimental area, which is 68 m long and 17 m wide. The latitude and longitude coordinates of the four boundary points are A (32.202038,119.509920), B (32.202160,119.510625), C (32.202013,119.510659), and D (32.201895,119.509959), respectively. The Pentagram symbol is the starting point of agricultural robot operation and the black squares are the divided grids in the figure. Due to the farm tools on a robot, the robot's operating width is set to 1 m, which means the size of each grid is 1 m × 1 m. The initial parameters of the robot are set as follows: the speed is 0.5 m/s, the heading angle is 0, and the coordinate of the start point is (32.202010, 119.510658). The proposed MCGA is used to plan the robot's path in the experimental area, and the path coordinate points are output in the form of latitude and longitude.

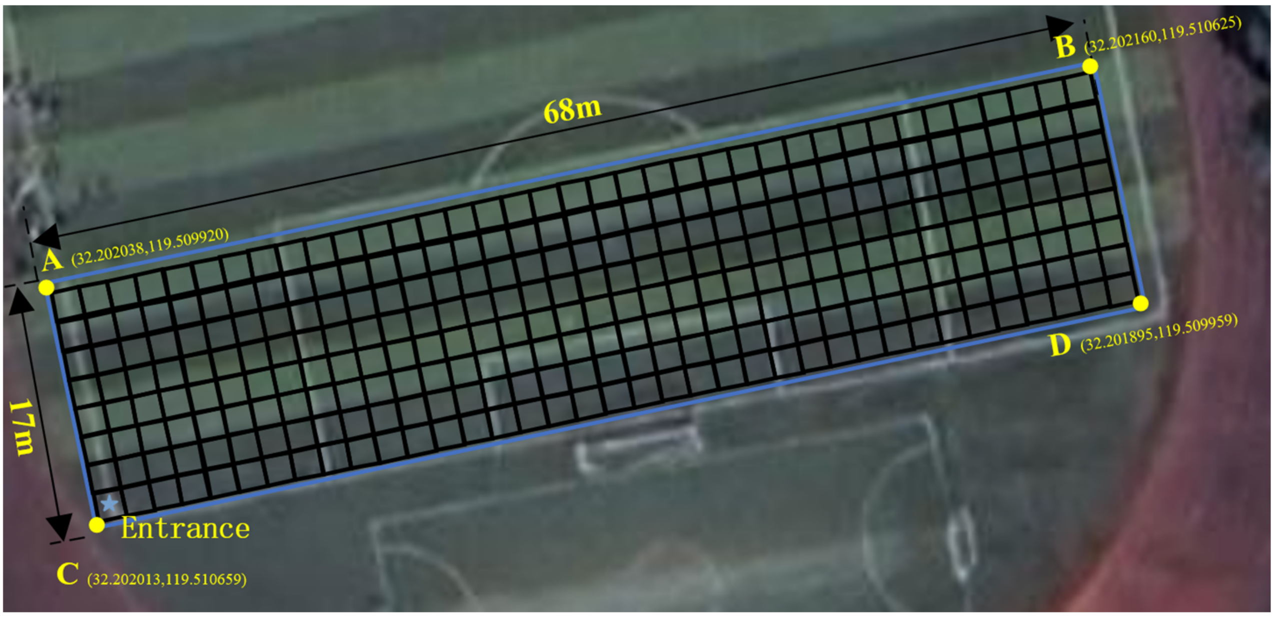


Figure 6. Test site map.

The robot's state parameters need to be collected during the experiment. Considering that ROS (Robot Operating System) is an open-source operating system suitable for robot development, it provides a structured communication layer on top of the host operating system of heterogeneous computing clusters [22]. The Rosbag tool of the ROS framework can provide an effective method for recording, replaying, analyzing, and processing data. By capturing data streams, the Rosbag can help researchers and developers quickly reproduce experimental scenarios based on offline data in repeatable, low-cost analysis, and debugging. Therefore, the ROS is used for sensor data collection and analysis. The ROS toolbox in MATLAB (2020 version) is used to extract and process the data packet that was recorded using the Rosbag tool. The ROS toolbox can provide an interface to connect MATLAB and Simulink (2020 version) with ROS. The toolbox includes MATLAB functions and Simulink blocks, which can visualize and analyze ROS data by recording, importing, and replaying the Rosbag files. Figure 7 shows the experimental site where the robot tracks the planned path.

Figure 7. Testing site.

After the experiment is completed, the data is extracted by the ROS toolbox and a planned robot tracking trajectory map is drawn. The trajectory and tracking trajectory of the path planning are shown in Figure 8, with the X axis representing longitude and the Y axis representing latitude. In the figure, the solid line is the planned path trajectory and desired path, and the point line is the trajectory tracked by the robot. From Figure 8, it can be seen that the robot can complete trajectory tracking along the planned path line. Using the planned path as a reference, the root mean square error of the robot's lateral position is 0.369 m, the average lateral error is 0.228 m, and the maximum lateral error is 0.566 m, as shown in Table 3. After on-site inspection of the trajectory points with significant lateral errors, it was found that the locations with large errors have some bumpy terrains, which indicates that the terrain on the ground has a certain impact on the walking trajectory of robots. In summary, the experimental results demonstrate the effectiveness and feasibility of the MCGA.

Table 3. Tracking performance of experimental environments.

RMSE (m)	Average Error (m)	Maximum Error (m)
0.369	0.228	0.566

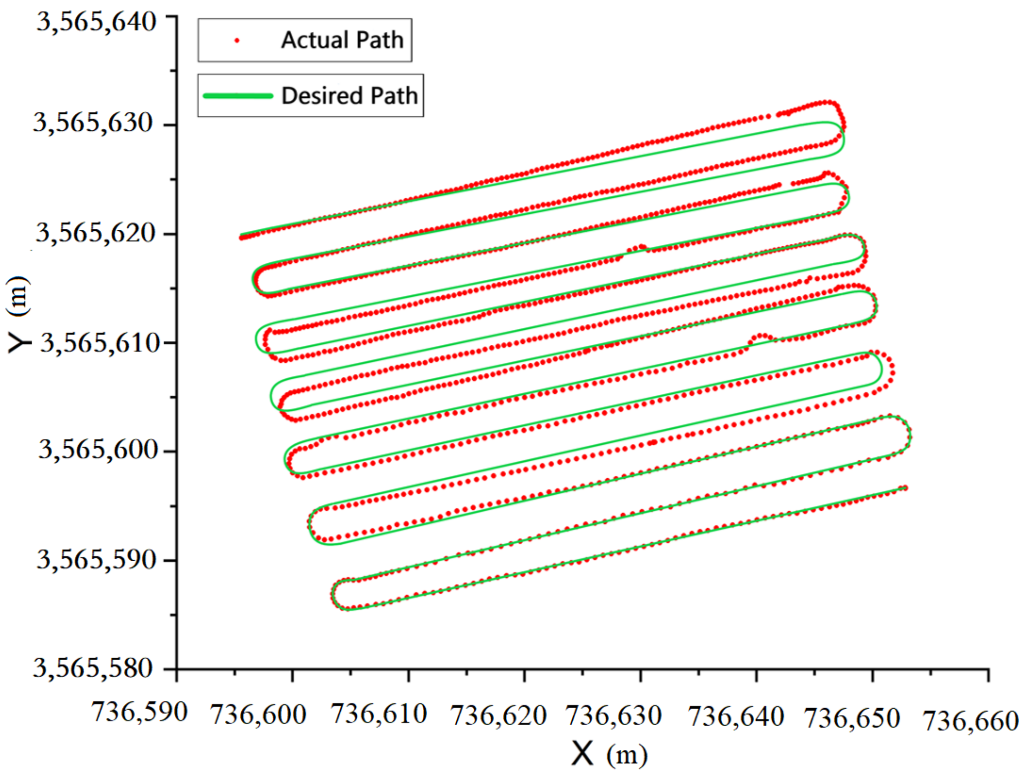


Figure 8. Robot tracking trajectory and path planning.

4. Conclusions

To solve the problems of low coverage, high repetition rate, and slow convergence speed of basic genetic algorithms in dealing with complete coverage path planning problems in a farmland environment, the paper introduced the Markov chain state transition matrix, which could guide the changes in gene sequences (solution vectors). The process simulates the mutation and selection process in genetic algorithms, using constrained random methods to generate new individuals. A complete covered path planner based on an improved genetic algorithm is designed.

A path planning simulation platform was built by MATLAB's APP Designer toolbox. The improved genetic algorithm designed is analyzed on the 6 different sizes field in the simulation platform. The simulation results show that compared with traditional genetic algorithms, the proposed improved genetic algorithm reduces the average path length by 21.8%, decreased the average number of turns by 6 times, decreased the repetition rate by 83.8%, and increased the coverage rate by 7.76%. The improved genetic algorithm proposed has significantly better performance and adaptability than traditional genetic algorithms in different work areas.

The improved genetic algorithm proposed is then used for path planning on the experimental site, which is applied to robot path tracking experiments. The experimental data shows that the walking trajectory of the robot is continuous, with a root mean square error of 0.369 m in lateral position and a maximum lateral error of 0.566 m.

The MCGA proposed in this article has been proven effective and adaptable in both simulation and practical environments. However, in the face of the complexity of agricultural robot operating environments, further research and improvement are still needed in existing studies. For example, research on the improvement of MCGA based on environmental factors and specific agricultural operation requirements of concave and convex plots with different shapes.

Author Contributions: Software, W.X. and F.W.; Data curation, W.L.; Writing—original draft, J.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Foundation of the Taizhou Science and Technology Program (No: TG202301).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is contained within the article.

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

References

1. Ren, G.; Lin, T.; Ying, Y.; Chowdhary, G.; Ting, K. Ting. Agricultural Robotics Research Applicable to Poultry Production: A Review. *Comput. Electron. Agric.* **2020**, *169*, 105216. [CrossRef]
2. Liu, C.; Gong, L.; Yuan, J. Current Status and development trends of agriculture robots. *Trans. Chin. Soc. Agric. Mach.* **2022**, *53*, 1–22.
3. Gonzalez, E.; Alvarez, O.; Diaz, Y.; Parra, C.; Bustacara, C. BSA: A complete coverage algorithm. In Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, 18–22 April 2005; pp. 2040–2044.
4. Zhu, D.; Zhu, T.; Yan, M. MultiAUV Complete Coverage Path Planning Based on Improved Neural Network. *J. Syst. Simul.* **2020**, *32*, 1505–1514.
5. Li, K.; Chen, Y.F.; Jin, Z.Y.; Liu, T.; Liu, Z.T.; Zheng, J.Z. A full coverage path planning algorithm based on backtracking method. *Comput. Eng. Sci.* **2019**, *41*, 1227–1235.
6. De Carvalho, R.N.; Vidal, H.A.; Vieira, P.; Ribeiro, M.I. Complete coverage path planning and guidance for cleaning robots. In Proceeding of the IEEE International Symposium on Industrial Electronics, Guimaraes, Portugal, 7–11 July 1997; pp. 677–682.
7. Li, R.; Zhou, C.; Dou, Q.; Hu, B. Complete coverage path planning and performance factor analysis for autonomous bulldozer. *J. Field Robot.* **2022**, *39*, 1014–1034. [CrossRef]
8. Shang, G.; Liu, G.; Zhu, P.; Han, J. Complete Coverage Path Planning for Horticultural Electric Tractors Based on an Improved Genetic Algorithm. *J. Appl. Sci. Eng.* **2020**, *24*, 447–456.
9. Selek, A.; Seder, M.; Petrovic, I. Mobile robot navigation for complete coverage of an environment. *IFAC-PapersOnLine* **2018**, *51*, 512–517. [CrossRef]
10. Le, A.V.; Nhan, N.H.K.; Mohan, R.E. Evolutionary Algorithm-Based Complete Coverage Path Planning for Tetriamond Tiling Robots. *Sensors* **2020**, *20*, 445. [CrossRef] [PubMed]
11. Zhou, C.; Huang, B.; Fränti, P. A review of motion planning algorithms for intelligent robots. *J. Intell. Manuf.* **2022**, *33*, 387–424. [CrossRef]
12. Liu, Q.; Wang, C.; Li, X.; Gao, L. An improved genetic algorithm with modified critical path-based searching for integrated process planning and scheduling problem considering automated guided vehicle transportation task. *J. Manuf. Syst.* **2023**, *70*, 127–136. [CrossRef]
13. Hao, K.; Zhao, J.; Li, Z.; Liu, Y.; Zhao, L. Dynamic Path Planning of a Three-dimensional Underwater AUV Based on an Adaptive Genetic Algorithm. *Ocean Eng.* **2022**, *263*, 112421. [CrossRef]
14. Ning, Y.; Zhang, F.; Jin, B.; Wang, M. Three-dimensional path planning for a novel sediment sampler in ocean environment based on an improved mutation operator genetic algorithm. *Ocean Eng.* **2023**, *289*, 116142. [CrossRef]
15. Ab Wahab, M.N.; Nazir, A.; Khalil, A.; Ho, W.J.; Akbar, M.F.; Noor, M.H.M.; Mohamed, A.S.A. Improved Genetic Algorithm for Mobile Robot Path Planning in Static Environments. *Expert Syst. Appl.* **2024**, *249*, 123762. [CrossRef]
16. Zarevich, A.; Oksyuta, M.; Gusev, P.; Nechetnyy, N. Modeling the behavior of a mobile robot using genetic algorithms in harsh ecological environment. *E3S Web Conf.* **2023**, *460*, 09032. [CrossRef]
17. Chen, Y.J.; Jhong, B.-G.; Chen, M.-Y. A Real-Time Path Planning Algorithm Based on the Markov Decision Process in a Dynamic Environment for Wheeled Mobile Robots. *Actuators* **2023**, *12*, 166. [CrossRef]
18. Edwards, G.T.; Hinge, J.; Skou-Nielsen, N.; Villa-Henriksen, A.; Sørensen, C.A.G.; Green, O. Route planning evaluation of a prototype optimized infield route planner for neutral material flow agricultural operations. *Biosyst. Eng.* **2017**, *153*, 149–157. [CrossRef]
19. Behrends, E. Markov chains. *Nature* **1972**, *236*, 291.
20. Holland, J.H. Genetic algorithms. *Scientific American.* **1992**, *267*, 66–73. [CrossRef]
21. Mao, Z.Y.; Liu, Z.J. A Trajectory Planning Method for Cubic Uniform B-spline Curve. *China Mech. Eng.* **2010**, *21*, 2569.
22. Quigley, M.; Conley, K.; Gerkey, B.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; Ng, A.Y. ROS: An open-source Robot Operating System. *ICRA Workshop Open Source Softw.* **2009**, *3*, 5.