

# Research on Path Planning for Industrial Cleaning Robots in Complex Environments: An Enhanced Immunogenetic Algorithm Approach

Hengtuo Pan<sup>2,\*</sup>, Shimin Qu<sup>3</sup>, Xingfeng Wang<sup>1,3</sup>, Wei Zheng<sup>3</sup>, and Ziqi Wang<sup>1,2</sup>

<sup>1</sup> College of Control Science and Engineering, Zhejiang University, Zhejiang 310027, China

<sup>2</sup> Huzhou Institute of Zhejiang University, Zhejiang 313000, China

<sup>3</sup> Inner Mongolia Huomei Hongjun Aluminum Electric Co., Inner Mongolia 029200, China

\*htpan\_96@foxmail.com

## Abstract

As industrial automation advances, cleaning robots in complex environments have demonstrated significant value in enhancing efficiency and reducing labor costs. This paper presents a path planning method based on an improved Immune Genetic Algorithm (IGA) for comprehensive coverage path planning in complex industrial settings. Initially, the Boustrophedon Cellular Decomposition (BCD) method is employed to effectively segment the complex scene, identifying key areas for cleaning tasks. Subsequently, the path planning efficiency and adaptability are enhanced through an optimized genetic algorithm that integrates immune memory, Inver-Over operators, and 3-opt operators to refine the traversal sequence between regions. Moreover, this study incorporates the A\* algorithm and the Biologically Inspired Neural Network (BINN) algorithm for efficient inter-regional path transitions and precise intra-regional traversal, respectively. A series of simulation experiments validate the superiority of the proposed algorithm in terms of path length, coverage rate, and computational time. The results demonstrate that this method not only effectively addresses the challenges of path planning in complex industrial environments but also significantly improves cleaning efficiency, offering high practical value and potential for broader application.

## Keywords

Path Planning; Industrial Cleaning Robots; Immune Genetic Algorithm; A\* Algorithm; Biologically Inspired Neural Network.

## 1. Introduction

As global industrial automation continues to advance, cleaning and maintenance have become crucial aspects of industrial environment management. In high-precision and high-standard production settings, the cleanliness of the environment directly impacts production efficiency and safety. Therefore, the development of a robotic system capable of efficiently and autonomously performing cleaning tasks is of paramount importance. However, traditional cleaning methods, such as manual cleaning or simple automated devices, often struggle to meet the demands of complex industrial environments. These environments are typically filled with obstacles and irregular spatial layouts, posing significant challenges for path planning.

In existing research, robotic path planning techniques have been extensively explored, with the primary goal of improving path efficiency and comprehensive coverage. While numerous algorithms have been proposed to address path planning in open or relatively simple environments, their effectiveness often diminishes when faced with more complex settings. For

instance, many algorithms fail to effectively manage the diverse obstacle layouts found in industrial environments, leading to path planning failures or inefficiencies.

The Boustrophedon Cellular Decomposition (BCD) method has emerged as an effective strategy for area decomposition and path planning, showing promising performance in certain applications, particularly in predefined and relatively stable environments [1]. However, in industrial settings, the standalone BCD algorithm faces several limitations. For example, during inter-cellular path planning, it is prone to issues of redundant coverage, which not only reduces cleaning efficiency but also increases energy consumption and time costs. Additionally, environments with numerous obstacles can result in an increased number of cells, thereby raising computational complexity and execution difficulty.

Therefore, researching and developing an improved path planning algorithm that can effectively address the full coverage challenges in complex environments, while enhancing overall path planning efficiency and adaptability, is of significant theoretical and practical value for achieving automated industrial cleaning. By integrating advanced algorithmic concepts and strategies, such as Immune Genetic Algorithms (IGA) and Traveling Salesman Problem (TSP) optimization, it is possible to significantly improve path planning performance. This will enable more precise and efficient cleaning operations, meeting the high standards of automation and intelligence required in modern industries.

In the field of industrial robotic path planning, various methods have been developed to meet different application needs. Traditional path planning techniques, such as grid mapping, Voronoi diagram decomposition, and wavefront expansion algorithms, provide solutions for basic navigation and obstacle avoidance [2]. These methods perform well in open or relatively simple environments, but their performance is limited in complex industrial settings, particularly in scenarios requiring full coverage and high efficiency.

The BCD method offers an effective strategy for path planning in complex environments. This method works by decomposing the environment into multiple cells and independently planning the path within each cell to achieve full area coverage. Although the BCD algorithm can deliver satisfactory coverage efficiency in certain situations, its efficiency and practicality diminish as environmental complexity increases, especially in scenarios with numerous obstacles.

In existing research, attempts to improve the BCD algorithm have included optimizing the generation of cells and the determination of traversal sequences. However, most improvements focus on local optimization of the algorithm without fully considering its performance enhancement in global applications. Moreover, the issue of redundant coverage between cells remains a key challenge that current methods have not effectively addressed. These limitations have prompted us to explore more efficient global optimization strategies.

This study aims to address the challenge of full coverage path planning for cleaning robots in complex industrial environments, with a particular focus on improving path planning efficiency and reducing redundant coverage. To this end, we propose an innovative path planning framework that combines an IGA with the BCD method. By optimizing the crossover and mutation strategies of the Genetic Algorithm (GA) and introducing an immune memory mechanism, this study significantly enhances the algorithm's search efficiency and stability in complex environments. Additionally, we integrate the optimization strategies of the TSP to optimize the traversal sequence between the decomposed cells, effectively reducing path redundancy and the number of round trips, thereby lowering overall runtime and energy consumption.

Specifically, the objectives of this research include developing an improved IGA that incorporates an immune memory mechanism for optimizing the robot's path planning and integrating the BCD algorithm with TSP optimization strategies to optimize the traversal

sequence. We will implement these algorithms and test them through simulations of complex industrial environments to verify their effectiveness and applicability. By conducting a comparative analysis with existing technologies, this study will demonstrate the advantages of the proposed approach.

## 2. Research Status and Related Work

In modern industrial production, cleaning and maintenance play a crucial role in ensuring production efficiency and product quality. This is particularly true in complex industrial environments, such as large manufacturing plants, warehouses, chemical plants, and other heavy industrial facilities, where environmental cleanliness not only affects product quality but also directly impacts the operational efficiency of equipment and the safety of workers. As a result, the application of industrial cleaning robots has increasingly become an important means of enhancing industrial automation and efficiency.

These robots are indispensable in high-risk and labor-intensive environments, such as chemical plants and heavy industrial production lines. They are designed and implemented to improve cleaning efficiency, reduce labor costs, and ensure safety during operations. With technological advancements, industrial cleaning robots must contend with increasingly complex and variable environments, including irregular surfaces, narrow spaces, and mechanical equipment scattered throughout the work area.

Industrial cleaning robots typically operate autonomously without direct supervision, capable of performing efficient and precise cleaning tasks in complex industrial settings. Equipped with a variety of sensors and navigation systems, such as LiDAR, cameras, and infrared sensors, these robots can accurately identify and avoid obstacles while assessing the condition of the cleaning area. Furthermore, they can autonomously navigate and optimize their cleaning paths using pre-programmed path planning algorithms, achieving full coverage and efficient cleaning. Technological advancements have significantly enhanced the applicability of industrial cleaning robots in complex environments. Modern robots integrate sensor technology, artificial intelligence, and machine learning, enabling them to perceive the environment in real-time and conduct precise path planning. Path planning algorithms, such as BCD, allow complex cleaning areas to be decomposed into multiple sub-regions, ensuring comprehensive coverage. Additionally, the combination of improved GA and the A\* algorithm optimizes the robot's traversal paths, reducing unnecessary repetition and time wastage.

In the field of automation and robotics, environmental modeling is fundamental to achieving efficient robotic operations. A well-constructed map model not only ensures that robots effectively understand their operating environment but also provides a solid foundation for key operations such as navigation, target recognition, path planning, and trajectory tracking. Although robots may operate in three-dimensional spaces in reality, their specific tasks often only require a two-dimensional map model. This model facilitates full environmental coverage through map manipulation, and the way the environment is described directly affects the performance of the traversal algorithm employed.

Currently, common representations of two-dimensional environmental maps mainly include the Grid Map [3], Visibility Graph [4], and Topological Map [5]. Each of these methods has its own characteristics and applicable scenarios, directly impacting the efficiency and effectiveness of path planning.

The Grid Map divides the working area into multiple equal-sized square grids, with each grid classified based on whether it contains an obstacle. This method directly corresponds to the target area, making it easy to generate and maintain. Grid maps are currently the most widely used method, especially suited for detailed and precise environmental descriptions. However, as the number of grids increases, the computational load and memory consumption for path

planning also increase, which can become a limitation when dealing with large-scale environments.

In contrast, the Visibility Graph method approximates obstacles in the working area using polygons and primarily extracts geometric features such as points, lines, and surfaces, thereby simplifying the map representation. The size of the map in this method is directly related to the number of obstacles, allowing for improved algorithmic efficiency. However, as the number of obstacles increases, the complexity of the map also increases, potentially affecting the real-time performance of the algorithm.

The Topological Map method focuses on describing the connectivity between nodes in the environment rather than specific scale information. This method transforms the high-dimensional path planning problem into a lower-dimensional topological space, significantly reducing storage space and computational demands. However, due to the lack of scale information, when a robot enters the same node from different locations, its internal positioning accuracy may decrease, leading to difficulties in localization.

Given the advantages and disadvantages of these various methods, this study adopts an environmental modeling strategy that combines Grid Mapping and Topological Mapping. This strategy first uses the Grid Mapping method to establish an accurate environmental model, then decomposes this grid model into cells and constructs a topological model for each cell. By combining these two models, this approach optimizes the performance of the path planning algorithm while ensuring comprehensive coverage, achieving both efficient environmental modeling and accurate path planning. The innovation of this method lies in its ability to merge the strengths of both models, maintaining detailed environmental descriptions while enhancing the capability to handle large-scale environments.

Once the map construction is completed, choosing an appropriate path planning method becomes a key step in achieving efficient task execution. The BCD method plays a crucial role in this stage. Implemented on the constructed two-dimensional map, the BCD method optimizes the robot's coverage path planning by effectively decomposing the environment into manageable cell regions. By dividing complex environments into more easily managed cells, the BCD method makes path planning not only more efficient but also easier to implement and monitor. This approach ensures comprehensive coverage while significantly enhancing the flexibility and responsiveness of operations.

After discussing how the BCD method optimizes path planning on a pre-constructed map, we must further consider how the robot can efficiently traverse these segmented cells to complete the cleaning tasks. This need naturally introduces the TSP, a classic optimization problem. In practical applications, we often encounter a specific form of the TSP, known as the Open Traveling Salesman Problem (OTSP), where the path's endpoint does not need to return to the starting point. This form is more common in scenarios involving linear or one-way routes, such as cleaning and inspection tasks.

There are various methods to solve the TSP, including exact algorithms and approximation algorithms. Exact algorithms, such as dynamic programming, branch and bound, and linear programming, can provide precise optimal solutions but have low computational efficiency for large-scale problems. On the other hand, approximation algorithms, like the greedy algorithm, minimum spanning tree heuristic, and geometry-based Christofides algorithm, offer fast solutions that may not be optimal. Additionally, heuristic algorithms, such as the nearest neighbor algorithm, insertion heuristic, and greedy cycle algorithm, are widely used in practical applications due to their simplicity and efficiency.

To address the Open Traveling Salesman Problem, we selected a metaheuristic algorithm, the GA. The GA simulates the process of natural selection, iteratively searching for the optimal solution through operations such as selection, crossover, and mutation. This algorithm is

particularly suited for handling OTSP because it effectively manages the path's openness and can find near-optimal solutions within a reasonable time frame, especially when dealing with large-scale and complex path planning tasks. The flexibility and robust global search capability of the GA make it an ideal choice for solving such optimization problems, particularly in industrial applications where optimizing the robot's inter-cell movement path enhances cleaning efficiency and reduces operational costs.

Building on the use of the GA to address the OTSP, we further explored the application of the IGA. The IGA is a heuristic search algorithm that combines the evolutionary mechanisms of the GA with the characteristics of the immune system, particularly its memory and recognition functions, to improve search efficiency and solution quality.

In recent years, the BINN algorithm has gradually been applied to mobile robot path planning. Wang et al. [6] improved the BINN algorithm by adding virtual non-obstacle adjacent points near boundaries and between obstacles, modifying the path decision method, and optimizing path quality. Zhu et al. [7] applied the BINN algorithm to underwater robots, solving the problems of multi-robot task allocation and path planning. The BINN algorithms in the above literature have been effectively applied to mobile robot path planning but have not fully utilized known environmental information. Based on prior environmental information, this paper employs a combination of BCD and BINN methods to solve the full-coverage path planning problem for mobile robots in complex industrial settings. This includes both intra-cell full-coverage path planning and inter-cell path transitions. The overall process is shown in Figure 1.

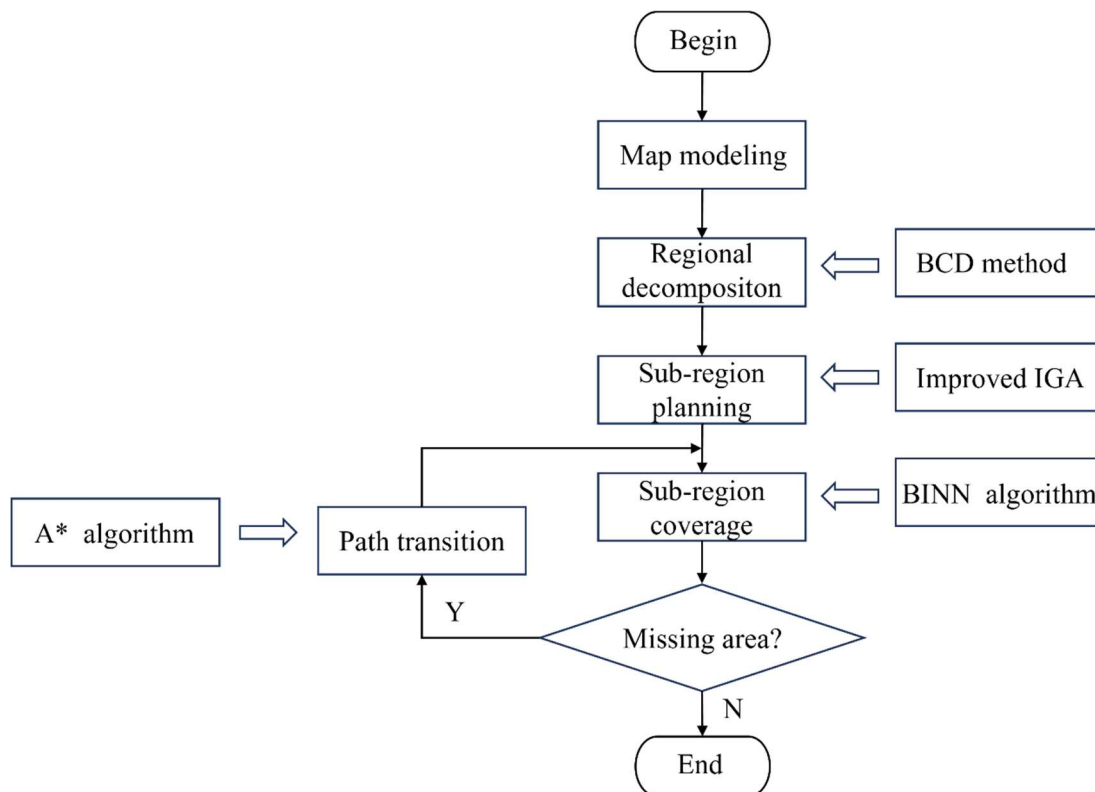


Figure 1. Flow chart of complete coverage path planning

### 3. Algorithm Principle

#### 3.1. Map Construction

The BCD method divides the environment into a relatively small number of sub-regions, which helps reduce the robot's redundant paths and repeated coverage, thereby ensuring operational

efficiency. Therefore, this study employs an environment segmentation method based on Boustrophedon Cellular Decomposition.

The implementation process of the Boustrophedon Cellular Decomposition method is as follows:

Step 1: Set the initial position and direction of the sweeping line, and determine the boundaries of the area as well as obstacle information.

Step 2: Gradually translate the line according to the predefined direction and step length.

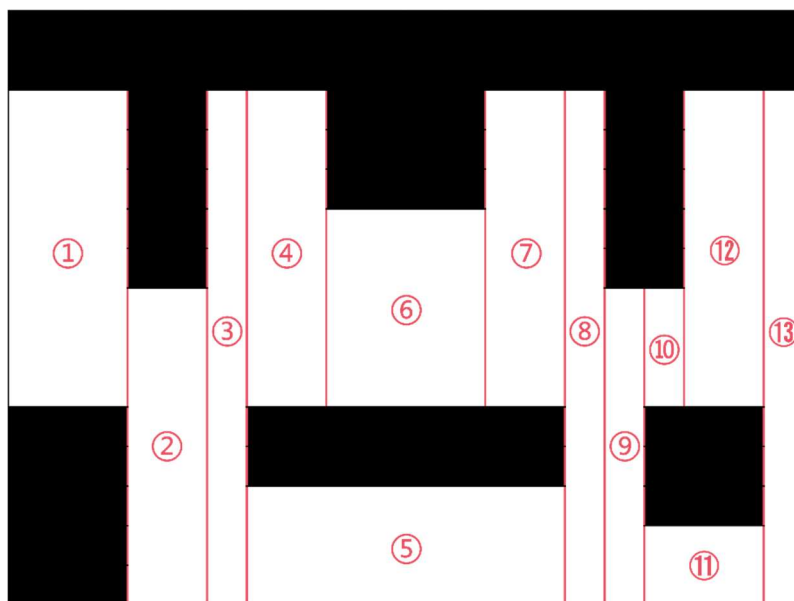
Step 3: During the translation process, determine whether the line intersects with the area's boundaries or obstacles.

Step 4: When the line intersects with a boundary or obstacle, divide the current area into two or more new sub-regions based on the location of the intersection points.

Step 5: Adjust the position and direction of the line according to the new sub-region boundaries.

Step 6: Repeat Steps 2 through 5 until the entire area is fully decomposed or until the predefined termination conditions are met.

The results of applying the Boustrophedon Decomposition method to segment the reference target area are shown in Figure 2. The sub-regions are labeled accordingly, and the results demonstrate that this method can effectively segment the environmental map, including various obstacles.



**Figure 2.** Environmental area decomposition map

### 3.2. Sub-region Traversal Sequence Planning

After a detailed division of the target area, multiple adjacent sub-regions can be obtained. Based on the adjacency of these sub-regions, a connectivity graph is constructed. To ensure efficient cleaning operations, Area 1 is designated as the starting area for traversal in the article. To determine the optimal traversal sequence for the robot, the article innovatively proposes an improved Immune Genetic Algorithm.

The Genetic Algorithm is a probabilistic search algorithm that simulates the natural genetic and evolutionary processes to achieve adaptive global optimization. This algorithm leverages the mechanisms of natural evolution, enabling the solution of problems by finding the best solution among multiple possible ones. Within the Genetic Algorithm, solutions are metaphorically referred to as "chromosomes," which are represented internally as a collection of symbols.

Through selection, crossover, and mutation, this produces individuals that are more suited to the environment.

The fitness function of the Genetic Algorithm, as shown in Equation (1), evaluates suitability based on the length of the path. The shorter the path, the higher its suitability, and thus, the greater the likelihood of it being selected.

$$F_{GA} = \frac{1}{(1 + \frac{1}{\sqrt{M-1}}) L_{GA}} \quad (1)$$

In the equation,  $F_{GA}$  represents the fitness function of the Genetic Algorithm,  $M$  denotes the number of grids in the environmental map model, and  $L_{GA}$  is the total length of the path.

Sub-region traversal sequence planning is performed using an improved IGA. The flowchart of the genetic algorithm is shown in Figure 3 and can be divided into the following steps:

#### Step 1: Initialization

- (1) Set initial population: Generate chromosomes for the initial population.
- (2) Initialize the immune memory bank: Establish a memory bank that stores historically excellent chromosomes, which can be reused in subsequent processes.

#### Step 2: Fitness Evaluation

Evaluate and update the memory bank: Assess the fitness of all individuals in the current population. Consider adding excellent individuals to the immune memory bank or updating existing chromosomes in the bank.

#### Step 3: Selection

- (1) Immune memory selection: Select a portion of excellent individuals from the immune memory bank to directly enter the next generation, preserving historically excellent genes.
- (2) Clone selection: Select the fittest individuals from the current population for cloning. These clones will undergo high mutation rates in the mutation phase.

#### Step 4: Crossover

Introduce the Inver-Over operator, which selects a gene from a parent chromosome and then inverts a segment of the chromosome based on the position of this gene and another randomly selected gene.

#### Step 5: Mutation

Apply the 3-opt operator: During the mutation phase, apply the 3-opt operator to selected chromosomes for local path optimization. This helps to further improve the solution to path problems by adjusting three nodes in the path to find a shorter total path.

#### Step 6: Formation of a New Generation

(1) Use the replacement operator to optimize the population structure when forming a new generation. Enhance individual fitness by comparing two chromosomes and selecting the better segments for replacement. Randomly select two chromosomes  $S_1, S_2$  from population  $\{P\}$ , let  $f(S_2) > f(S_1)$ ; randomly select a gene segment  $\Delta S_1$  from  $S_1$ , then judge if there is gene segment  $\Delta S_2$  in  $S_2$  that meets the conditions below: it contains the same points as  $\Delta S_1$ , but in a different order. If  $\Delta S_2$  exists, replace  $\Delta S_1$  by  $\Delta S_2$  in chromosome  $S_1$ , then the rest genes are adjusted follow the partial mapping rules.

(2) Continue to perform negative selection on the current population, removing those individuals with low fitness and similar to high fitness solutions to optimize population quality.

(3) Integrate new individuals: Combine individuals that have undergone selection, crossover, and mutation to form a new generation. Individuals from immune memory can also be used at this stage to replace underperforming individuals.

Step 7: Termination Condition Evaluation

Repeat the above steps until the preset termination conditions are met, such as reaching the maximum number of iterations, achieving a fitness threshold, or no significant improvement in the solution.

Step 8: Output the Optimal Solution

Select the best chromosome: Choose the individual with the highest fitness from the final population as the final solution to the problem.

Through the improved Immune Genetic Algorithm, an optimized sub-region traversal sequence diagram is obtained, as shown in Figure 4. This sequence diagram not only ensures that the cleaning robot can efficiently complete the cleaning tasks of the entire target area but also significantly reduces the time and energy consumption of the robot moving between different sub-regions.

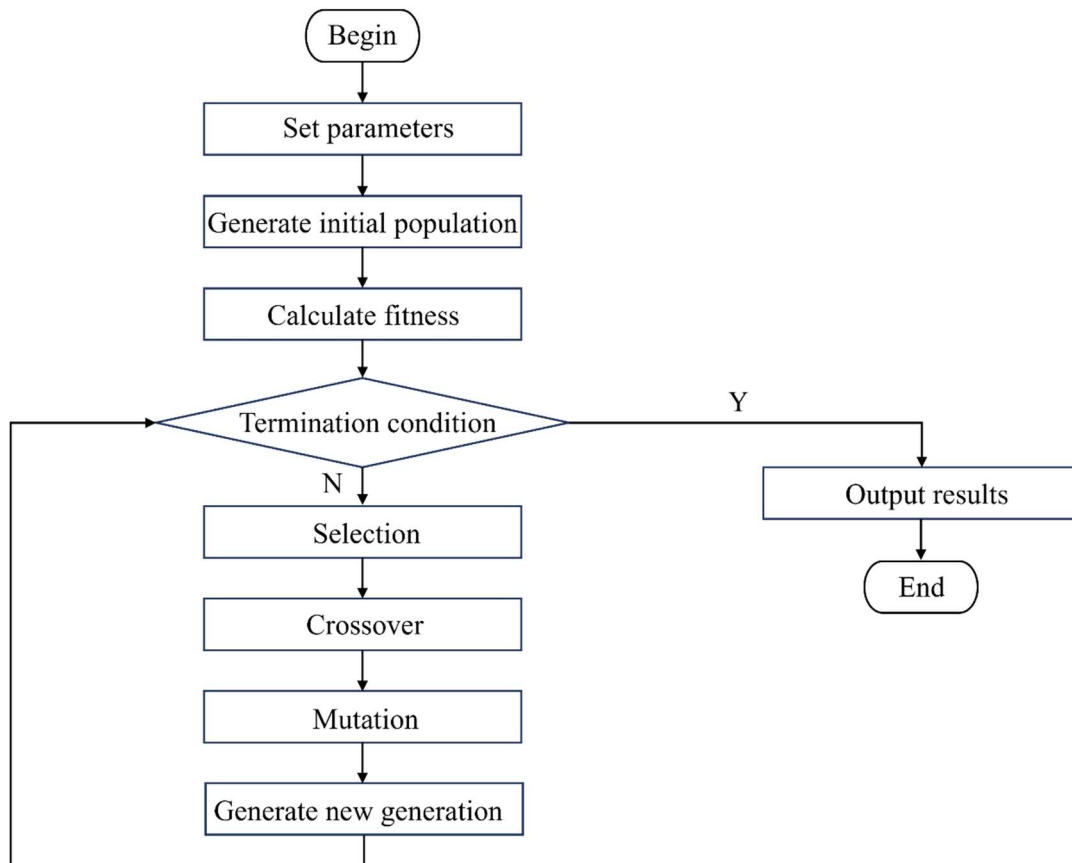


Figure 3. Flowchart of genetic algorithm

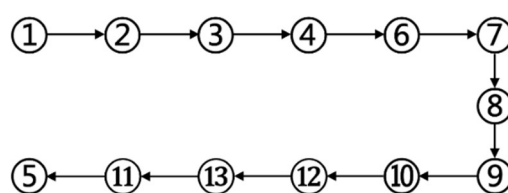


Figure 4. Sub-region traversal sequence diagram

### 3.3. Intra-subregion Traversal

After a detailed division of the target area, multiple adjacent sub-regions can be obtained. Based on the adjacency of these sub-regions, a connectivity graph is constructed. To ensure efficient cleaning operations, Area 1 is designated as the starting area for traversal in the article. To determine the optimal traversal sequence for the robot, the article innovatively proposes an improved Immune Genetic Algorithm.

The BINN algorithm is an online planning algorithm that treats each grid as a neuron, turning the entire grid map into a topological state space composed of neural networks. The activity value of each neuron is represented by the following equation:

$$\frac{dx_i}{dt} = -Ax_i + (B - x_i) \left( [I_i]^+ + \sum_{j=1}^k w_{ij} [x_j]^+ \right) - (D + x_i) [I_i]^- \tag{2}$$

Equation (2) is the flow distribution equation, which can be used to calculate the activity value of each neuron in the grid map. In the equation:  $x_i$  represents the activity value of the  $i$ -th neuron;  $t$  is the time variable;  $A$ ,  $B$ , and  $D$  are non-negative constants, where  $A$  represents the decay rate,  $B$  represents the upper limit of the neuron's activity state, and  $D$  represents the lower limit of the neuron's activity state;  $k$  is the number of neurons directly connected to the  $i$ -th neuron;  $I_i$  represents the external input, which can be defined as:

$$I_i = \begin{cases} E, & \text{Target point} \\ -E, & \text{Obstacle} \\ 0, & \text{others} \end{cases} \tag{3}$$

In the equation,  $E$  is a constant significantly greater than  $B$ ;  $([I_i]^+ + \sum_{j=1}^k w_{ij} [x_j]^+)$  and  $[I_i]^-$  respectively represent excitatory and inhibitory inputs, where  $w_{ij} = f(d_{ij})$ . Here,  $d_{ij}$  is the Euclidean distance between the location of the  $i$ -th neuron and the  $j$ -th neuron in the state space, and  $f$  is an arbitrary monotonically decreasing function, which can be defined as follows:

$$f(d_{ij}) = \begin{cases} \frac{u}{d_{ij}}, & 0 < d_{ij} < r_0 \\ 0, & d_{ij} \geq r_0 \end{cases} \tag{4}$$

In the formula,  $u$  and  $r_0$  are both positive constants, with  $u$  representing the lateral connections between the current neuron and surrounding neurons, where  $0 < u < 1$ . Neurons are only locally connected within a small range  $(0, r_0)$ , which is referred to as the receptive field of neuron  $i$ . This receptive field ensures that the 8 surrounding neurons are all within the local connection range. The length of each grid is 1, and  $r_0$  falls within the range  $(\sqrt{2}, 2\sqrt{2})$ . The synaptic weights between neurons are symmetrical, i.e.,  $w_{ij} = w_{ji}$ .

The robot generates a complete coverage path as follows:

$$p_n \leftarrow x_{p_n} = \max \{x_j + cy_j, j=1, 2, \dots, k\} \tag{5}$$

In the model,  $c$  is a positive constant representing the choice of directional weights, and  $y_j$  is a function related to the robot's previous position  $p_p$ , current position  $p_c$ , and next position  $p_j$ . The function  $y_j$  is defined as follows:

$$y_j = 1 - \frac{\Delta\theta_j}{\pi} \tag{6}$$

In the equation,  $\Delta\theta_j \in [0, \pi]$ , represents the angle between the current direction of movement and the next direction of movement. The process of full-coverage path planning is shown in Figure 5.

### 3.4. Regional Path Transition

After the internal coverage of the current sub-region is completed, a transition to the next sub-region is required before starting its internal coverage. This transition allows the mobile robot to cover all sub-regions according to the planned sequence of sub-regions. To address this issue, this section primarily utilizes the A\* algorithm to plan and search for the shortest path from the endpoint of traversal in one sub-region to the starting point of traversal in the next sub-region, calculating the total cost of the algorithm in terms of the shortest path and time consumption.

The A\* algorithm is a widely used pathfinding and graph traversal algorithm that can find the shortest path from a start point to an endpoint. The core of the A\* algorithm is its heuristic function, typically expressed as  $f(n) = g(n) + h(n)$ , where  $g(n)$  is the actual distance from the start point to the current node  $n$ ;  $h(n)$  is the estimated distance from node  $n$  to the target point, estimated using Euclidean distance.

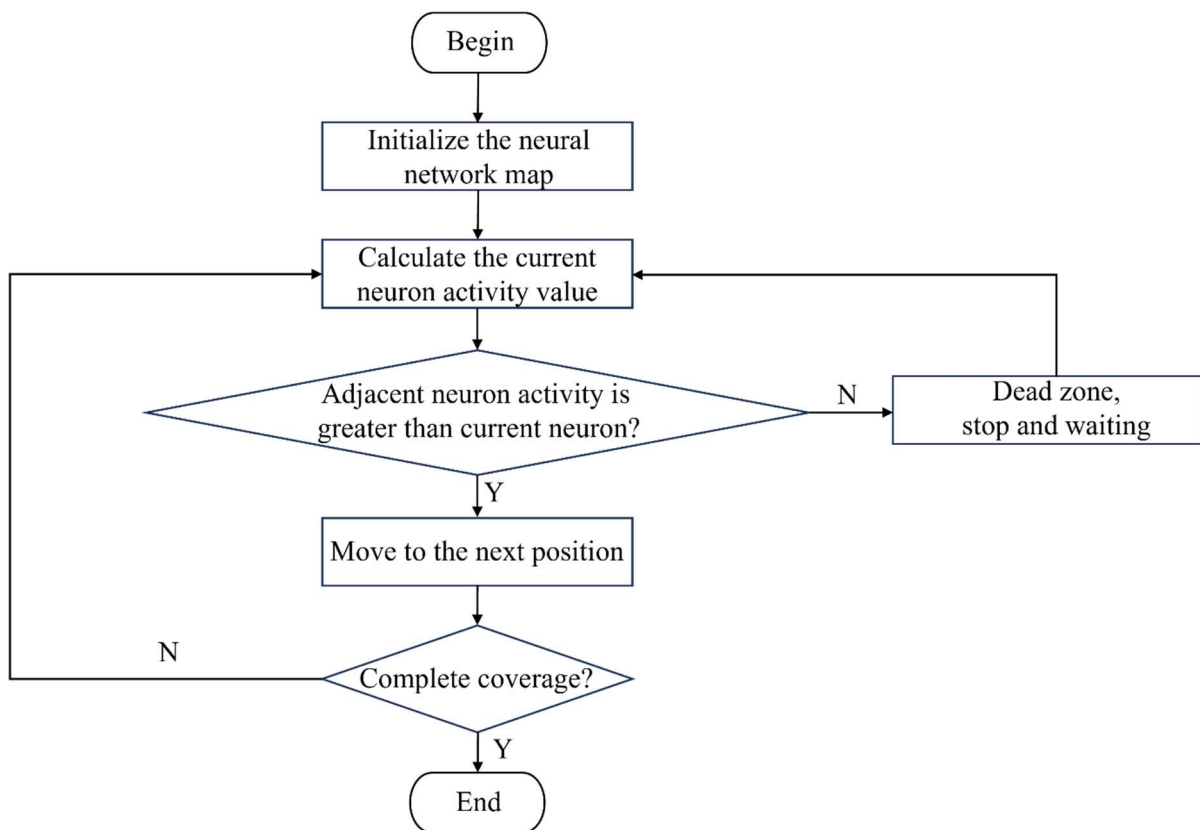


Figure 5. Flowchart of the BINN algorithm

The execution process of the A\* algorithm is as follows:

**Step 1: Initialization**

Place the starting node into the open list.

**Step 2: Loop**

Remove the node  $n$  with the smallest  $f(n)$  value from the open list and move it to the closed list.

Check if  $n$  is the target node. If it is, reconstruct the path and complete the search.

Otherwise, expand node  $n$  by exploring all its adjacent nodes, calculate the  $f$  values for these nodes, and add them to the open list.

Update the  $f$ ,  $g$ , and  $h$  values of adjacent nodes. If an adjacent node is already in the open list, update its path only if the  $g$  value calculated through the current node  $n$  is lower.

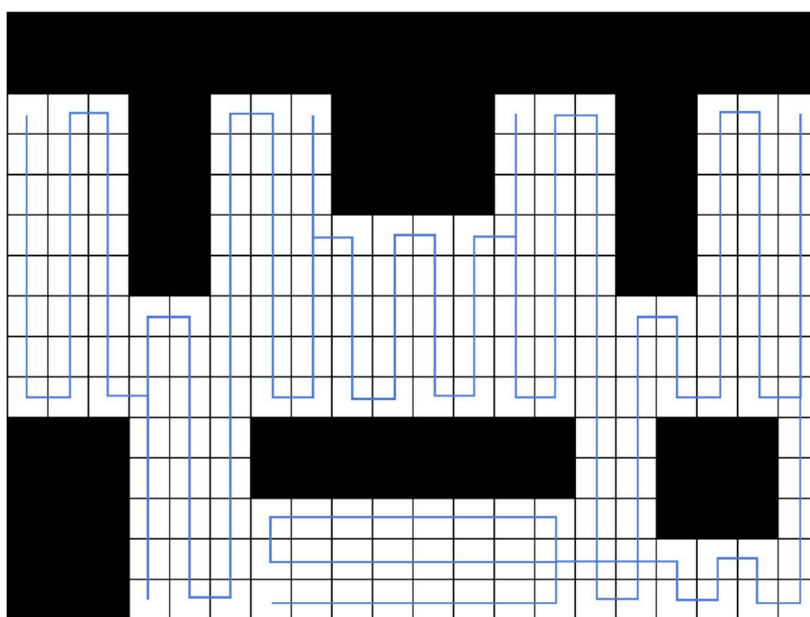
**Step 3: Repeat**

Continue this process until the open list is empty or the shortest path is found.

## 4. Results and Discussion

In this study, simulation experiments were conducted to validate the effectiveness and efficiency of the proposed path planning method in complex industrial environments. The experimental layout included various obstacle configurations to simulate the complexity of real industrial settings. In the result diagrams, black grids represent obstacles that hinder the robot's straight-line movement, while red lines indicate the movement path of the cleaning robot during task execution.

The experimental environment consisted of a simulated industrial area, including fixed mechanical equipment and randomly distributed temporary obstacles. The simulated environment was a 20x15 grid, with each grid representing one square meter of actual space. The task of the cleaning robot was to cover all accessible non-obstacle areas and optimize the walking path to reduce total cleaning time and energy consumption.



**Figure 6.** Coverage path

The results of the simulation experiments showed that after applying a path planning strategy combining an improved IGA with the A\* algorithm and BINN algorithm, the cleaning robot was able to effectively avoid obstacles and find nearly the shortest possible path for area cleaning.

As shown in Figure 6, the blue lines clearly marked the robot's travel route, smoothly transitioning from the endpoint of one sub-region to the starting point of the next, demonstrating the coherence and efficiency of the path planning.

Furthermore, compared to traditional reciprocating path planning methods, the path planning strategy optimized with the Immune Genetic Algorithm reduced the total walking distance by about 15%. This significant improvement was due to the algorithm's enhanced global optimization capabilities and adaptability to complex environments.

The analysis of results indicates that the path planning strategy, integrating the BCD method and the BINN path coverage algorithm, is not only theoretically feasible but also demonstrates high efficiency and practical performance in real applications. The Immune Genetic Algorithm showed superior adaptability and robustness compared to traditional methods when dealing with dynamically changing environments and diverse obstacle layouts.

## 5. Conclusion

In this study, we proposed a path planning method combining the BCD and A\* algorithm, aimed at optimizing the efficiency and route of cleaning robots in complex industrial environments. By incorporating an improved IGA, we not only enhanced the global optimization capability of the path planning but also increased its adaptability to complex and changing environments. The experimental results indicate that the proposed method can effectively handle complete coverage cleaning tasks in complex environments, significantly improving the efficiency and cost-effectiveness of path planning. These achievements demonstrate that the strategy presented has high efficiency and practical performance in the field of industrial cleaning, and also shows potential for other applications such as warehouse logistics, automated inspections, and emergency response.

Despite the positive outcomes of this research, future work can further explore and expand in several directions. First, more heuristic and metaheuristic strategies could be explored to adapt to more complex or dynamically changing environments. Second, researching path planning strategies for collaborative cleaning with multiple robots could further enhance cleaning efficiency and response times in large-scale industrial settings. Lastly, deploying the developed path planning method in real industrial environments for field testing and data collection will help verify the practicality and stability of the algorithm, ensuring its effectiveness in real applications. Through these measures, future research will continue to advance cleaning robot technology, meeting the growing needs of industrial automation and intelligence.

## Acknowledgments

This paper was supported by Smart Aluminum Plant Technological Innovation Project of Inner Mongolia Huomei Hongjun Aluminum Electric Co., Ltd (K20231334).

## References

- [1] T. M. Cabreira, L. B. Brisolará and R. F. J. Paulo. Survey on coverage path planning with unmanned aerial vehicles, *Drones*, Vol. 3 (2019) No.1, p.4.
- [2] D. Li, P. Wang and L. Du. Path planning technologies for autonomous underwater vehicles-a review, *Ieee Access*, Vol. 7 (2018), p.9745-9768.
- [3] H. Ali, D. Gong, M. Wang, et al. Path planning of mobile robot with improved ant colony algorithm and MDP to produce smooth trajectory in grid-based environment, *Frontiers in neurorobotics*, Vol. 14 (2020), p.44.
- [4] W. Lee, G. H. Choi and T. Kim. Visibility graph-based path-planning algorithm with quadtree representation, *Applied Ocean Research*, Vol. 117 (2021), p.102887.

- [5] E. Beeching, J. Dibangoye, O. Simonin, et al. Learning to plan with uncertain topological maps, Computer Vision – ECCV 2020, (Glasgow, UK, August 23-28, 2020). Vol. 12348, p.473-490.
- [6] Y. N. Wang, Q. Pan and Y. J. Chen. Path planning method based on improved biologically inspired neural network, Control Eng China, Vol. 25 (2018) No.4, p.541.
- [7] D. Q. Zhu, M. Z. Yan. Survey on technology of mobile robot path planning, Control Des, Vol. 25 (2010) No.7, p.961.