# Multi-Target Robot Path Planning Using Enhanced Genetic Algorithms and Probabilistic Roadmaps

Shaymaa Alzubairi[1,*], Alexander Petunin[1], Mohammed Majid Msallam[2], Hussam Lefta Alwan[3]

[1]Institute of New Materials and Technologies, Ural Federal University, Ekaterinburg, Russia

[2]Control and Systems Engineering Department, University of Technology, Baghdad, Iraq

[3]Production Engineering and Metallurgy, University of Technology, Baghdad, Iraq

Received 30 March 2025; received in revised form 20 July 2025; accepted 24 July 2025

DOI: https://doi.org/10.46604/peti.2025.14989

## Abstract

Path planning receives considerable attention over the last two decades. This study proposes a hybrid approach that combines the probabilistic roadmap with an enhanced genetic algorithm (EGA), enabling path planning for both single and multiple targets. Compared with existing genetic algorithm (GA) methods, the proposed approach offers three main advantages: (1) it employs an environment representation based on image processing and morphological operations; (2) it introduces a new strategy for creating the initial population of the GA; and (3) it incorporates a novel operator to increase the quality of the generated paths. To demonstrate the effectiveness of the probabilistic roadmap and enhanced genetic algorithm (PRMEGA), multiple simulation experiments are performed, with results compared against the GA, artificial bee colony, and particle swarm optimization. The proposed approach outperforms existing methods by 25.5%, achieving near-optimal paths for both single and multiple targets in fewer generations while also reducing computation time by 14.1%.

**Keywords:** path planning, genetic algorithm, probabilistic roadmap algorithm, mobile robot

## 1. Introduction

Mobile robots are widely applied across diverse fields [1-3], including exploration, medical treatment, education, hazard seeking, and target finding. The mobile robot navigation process involves several key prerequisites, such as path planning, robot localization, and environment characterization [4]. Path planning aims to find an optimal or near-optimal path from a starting point to one or multiple targets, which may be independent, dependent, or a combination of both [5]. The path planning problem has been actively investigated, with numerous approaches developed to address this issue. A novel approach based on the geometrical structure proposed by researchers [6] allows effective configuration space search and optimal path planning for robots, but it needs to balance accuracy and speed. An alternative method to improve robot path planning and obstacle avoidance efficiency was proposed by Wu et al. [7]. It adds a temperature component to the potential field function, but its sensitivity to parameter tuning remains a shortcoming.

An effective neural network model was developed by Diao et al. [8]. To guide path planning and avoid obstacles, the model generates weights for each neighbor based on the obstacle, the searched path, and the random geometric graph. However, this approach leads to overfitting and increases computational complexity. A hybrid particle swarm optimization–simulated annealing (PSO-SA) method for automated guided vehicle (AGV) path planning was developed by Lin et al. [9]. With faster

* Corresponding author. E-mail address: shaymaaalzubairi77@gmail.com

convergence and lower time consumption, this method reduces the possibility of becoming trapped in a local optimum, although at the expense of accuracy. Each approach offers advantages in specific aspects. Robot path-planning problems typically involve three key challenges: computational complexity, local optimum, and adaptability.

Genetic algorithms (GAs) have been effectively applied to numerous optimization problems since their introduction in 1975. Based on the principle of 'survival of the fittest,' they operate as stochastic search strategies inspired by natural evolution. Their success in many applications can be attributed to their parallel search function and their quick identification of high-performance regions [10]. Consequently, GAs are widely adopted for mobile robot path planning. An improved GA was proposed by Ab Wahab et al. [11]. The new model uses the fitness score of each cell in the environment to guide the population initialization process, which reduces the number of impractical paths. However, the cell-based approach still faces a challenge: it must maintain a balance between accuracy and speed.

Heng and Rahiman [12] present a new optimization technique called the modified ant colony optimization and genetic algorithm (MACOGA), which is designed to navigate grid spaces quickly and effectively; however, it is not suitable for large maps. Also, ACO still requires fine-tuning of parameters. A novel knowledge-based GA for creating a collision-free path in a complex environment is introduced by Li et al. [13]. It incorporates a local search technique with five operators tailored to the situation, but this improvement comes at the cost of increased time consumption. Nevertheless, GA approaches still face the following three important problems in motion planning:

(1) Environment representation: Cell-based methods divide the environment into a 2D matrix, marking each cell as black for obstacles and white for empty [14]. This approach requires a trade-off between speed and accuracy. If speed is prioritized, coarse-grained cells can be used, but they may misclassify free and occupied cells, leading to longer and lower-quality paths. Conversely, fine-grained cells provide more accurate classification and near-optimal paths, but at the cost of increased computation time [15].

(2) Existing GA methods generate an initial population of paths, regardless of whether they are feasible or unfeasible. This random population may slow down the convergence rate, which may result in finding an appropriate path in higher generations [16].

(3) The stochastic nature of GA in generating population paths may result in routes with sharp turns and zigzag movements, rendering them unsuitable for real-world applications.

This study proposes a hybrid approach combining the probabilistic roadmap method (PRM) with an enhanced genetic algorithm (EGA), referred to as the probabilistic roadmap and enhanced genetic algorithm (PRMEGA). It is designed to plan an optimal or near-optimal path for a single target or multiple independent targets. The approach depends on image processing and morphological operations to build realistic representations, replacing conventional cell-based methods. This strategy removes the trade-off between accuracy and speed in environment modeling. Furthermore, the proposed approach introduces a new method for generating the initial population in the GA, leveraging the PRM to produce collision-free paths [17-18]. This method speeds up convergence and enables the identification of a near-optimal path in fewer generations. In addition, a novel operator is incorporated to improve the quality of the paths produced by the PRMEGA.

## 2. Problem Description

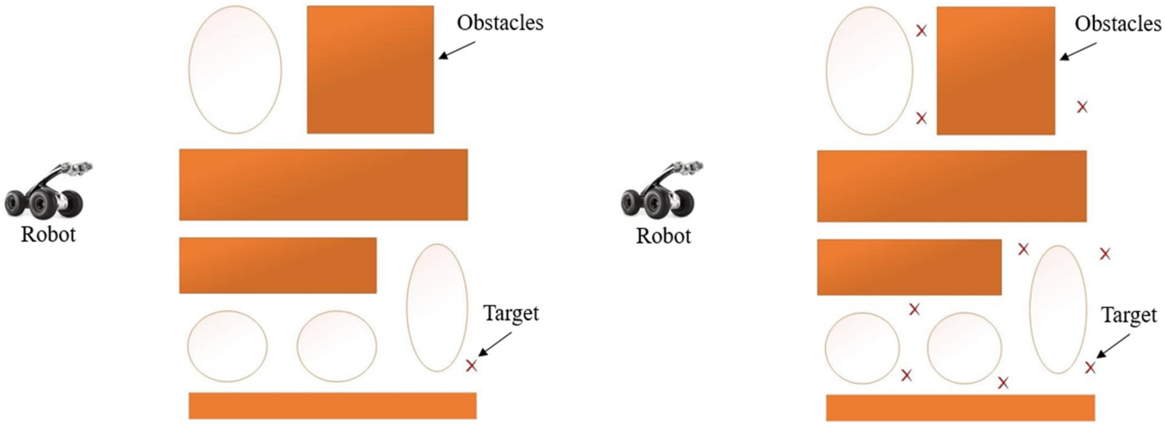
This study addresses the challenge of planning an optimal or near-optimal path between a start point and a single target in environments filled with various obstacles, as shown in Fig. 1(a), and between a start point and multiple independent targets, as shown in Fig. 1(b). There is no particular sequence for visiting the targets since they are independent. This is considered a difficult case because the proposed approach must determine the visiting sequence within the route planning to achieve the shortest and fastest path. The following assumptions were taken into consideration during the research:

(1) All environments are assumed to be static.

(2) All environmental information is assumed to be known in advance.

(3) No information is provided regarding the order of targets, as the targets are independent of each other.

(4) In this research, the mobile robot is modeled as a point, and the obstacle boundary is defined as the actual limits of the obstacles plus the minimum safety distance required for the robot[19].




(a) Single target (b) Multiple targets

Fig. 1 An environment with a single target VS. multiple targets

## 3. Proposed Methodology: A Hybrid PRM and EGA Approach

The methodology in this study is a hybrid of the PRM and EGA, hereafter referred to as PRMEGA. It utilizes PRM to generate feasible paths that serve as the initial GA population. Furthermore, it improves the basic GA operators and exploits the parallel search capability of GA to produce an optimal or near-optimal smooth path.

### *3.1. Environment representation (map building)*

This work employs image processing and morphological operations to interpret realistic maps and compose the robot's environments. The method consists of a set of codes, implementable in any programming language, that first reads a 2D map image in .jpg format. The image is then converted to grayscale and subsequently to a binary image. Afterwards, all objects and their sizes are identified. The proposed method outlines the objects' boundaries using morphological operations, and finally fills all identified objects with black. The environment representation consists of the following steps:

Input: 2D image map
Output: A map that can be recognized by algorithms

1. Read the 2D map as an image.
2. Convert the image to grayscale, then convert it to a binary image (0.8).
3. Define main objects inside the image.
4. Apply erosion and Dilation (morphological operation).
5. Find the outer boundaries of the objects in the image.
6. With the objects' borders, create a new binary picture.
7. Program features of the binary image's regions.
8. Create a map of binary occupancy.
9. Display the occupancy map.

### *3.2. Path representation (GA chromosome representation)*

In the proposed approach, a path can consist of a start and a single target, as shown in Fig. 2(a), or a start and several n targets $\{T_1, T_2, T_3, \cdots, T_n\}$, which may appear in any order, as shown in Fig. 2(b).

| S, ...................., T | S, ...... $T_1$...... $T_2$, ...... $T_3$ ..........., $T_n$ |
|---|---|
| (a) Single target | (b) Multiple targets |

Fig. 2 Path of the robot

Paths are represented as chromosomes of varying lengths, and path nodes (Cartesian floating points) serve as chromosome genes. A chromosome is shown in Fig. 3; where $(x_0, y_0)$ is the starting point and $(x_n, y_n)$ is the last point, with $(x_i, y_i)$ and $(x_{i+1}, y_{i+1})$ connected by a line segment.

| | | | | |
|---|---|---|---|---|
| $(X_1, Y_1)$ | $(X_2, Y_2)$ | … | $(X_{n-1}, Y_{n-1})$ | $(X_n, Y_n)$ |

Fig. 3 Chromosome representation

### *3.3. Initial population*

The proposed approach has a main advantage over other existing GAs in how it generates the initial population. Conventional GAs are blind. They generate paths from the starting point to the goal without considering potential collisions with obstacles, which increases both the number of generations and the time required to achieve a feasible solution. In contrast, the proposed approach utilizes PRM to construct feasible paths, as collision avoidance is the primary feature of PRM. The three major steps of the PRM [20] are (1) creating random uniform milestones in the free configuration space, (2) connecting these milestones to form a graph by linking edges that cross the free space, and (3) analyzing the graph to find the shortest path between the start and target milestones.

The PRM steps are as follows:

Input: Map and PPM parameters
Output: Feasible path

1. Initialize an empty graph G.
2. In the configuration space, generate random samples and look for instances where they collide with obstacles.
3. Add collision-free samples to graph G.
4. Connect nearby samples in G using local planners and check for collisions.
5. Add collision-free edges to graph G
6. Repeat steps 2–5 until a sufficient number of samples and connections are added to G.
7. Utilize a path planning method (such as Dijkstra's or A *) to determine a route from the start point to the goal.

PRM performs well in maps with small, medium, and somewhat large areas, but its efficiency decreases in maps with complex or very large areas, where computational time and memory usage increase, and node distribution deteriorates. These drawbacks can be reduced by adopting efficient strategies in sampling techniques, data structures, and by optimizing the number of milestones and edges.

If only one target is present, the PRM generates a path from the start point to the target. When multiple targets are present, a different tactic must be used. The PRM is applied between the start point and all targets, as well as between each target and the remaining targets, at least three times. This process creates several segment lines between the start point and each target, as well as between each target and the others. Next, these lines are connected to form feasible paths from the start point to all targets in a random sequence, ensuring that no target is visited more than once in any path. This process is repeated until the required GA population is generated. Although these paths are collision-free, they may contain sharp turns and be unsuitable for real-world applications. Therefore, these paths must be refined using genetic operators to achieve optimal or near-optimal solutions.

*3.4. Objective function*

The fitness function evaluates the quality of the paths. Since every path in this work is feasible, the evaluation focuses solely on path length, without considering feasibility. Accordingly, the objective function $f$ is minimized, with distance serving as the optimization criterion, as expressed in the following equation.

$$f = \frac{1}{\sum_{i=1}^{n} d_i} \tag{1}$$

where $\sum_{i=1}^{n} d_i$ is the summation of distances between the nodes along the path, and $n$ is the number of path nodes. In this work, the Euclidean distance is used to calculate the $d_i$ [21-22].

$$d_i = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \tag{2}$$

In Eq. (2), $x_i$ and $x_{i+1}$ are the X-coordinates of the $i$th and $(i+1)$th node of path P, respectively. Similarly, $y_i$ and $y_{i+1}$ are the Y-coordinates of the $i$th and $(i+1)$th node of path P.

*3.5. Operators of GA*

The three genetic operators (selection, crossover, and mutation) used in traditional GAs to mimic natural selection provide their powerful search capabilities [23-24]. In this section, the genetic operators are refined to maintain and enhance the feasibility of population paths and achieve optimal solutions. The flowchart of the proposed approach is depicted in Fig. 4.

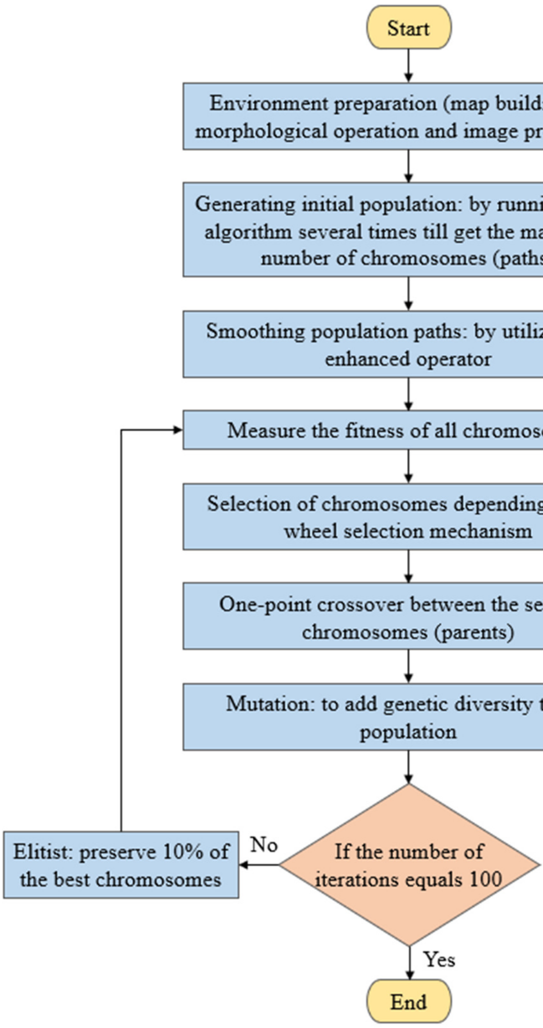



Fig. 4 Flowchart of the proposed approach

(a) Enhanced operator

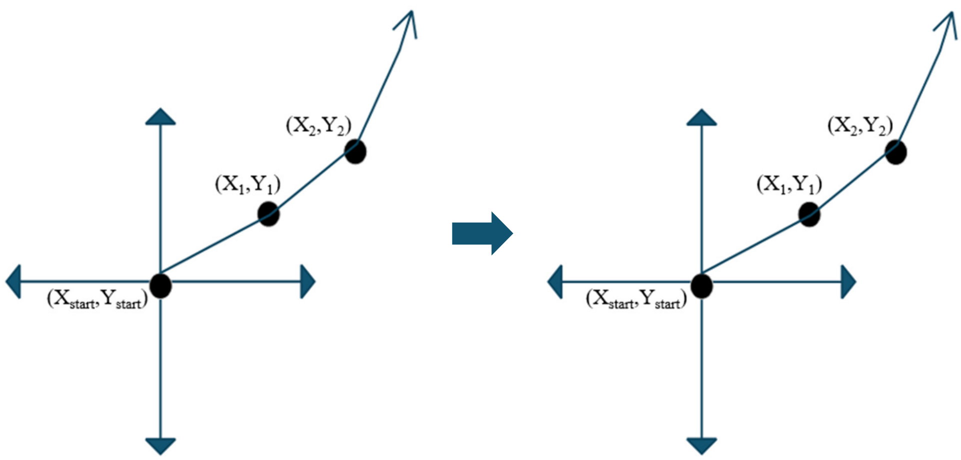
The path generated by PRMEGA is collision-free and achieves the highest fitness value. However, due to the randomized nature of both PRM and GA, the resulting paths often contain sharp turns and zigzag movements. This issue is particularly evident immediately after the start point and before the target point when a single target is present, or before each target point when multiple targets exist. These irregularities render the paths unsuitable for real-world applications. An enhanced operator is developed to refine the path of PRMEGA by smoothing sharp turns and improving overall path quality.

This operator utilizes a slope-based calculation and focuses on the first two nodes after the start point, as well as the last two nodes before each target in the path.

$$Slope = \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1} \tag{3}$$

where $\Delta y$ represents the change in Y, and $\Delta x$ represents the change in X. Here, $x_1$ and $x_2$ are the respective X-coordinates of the given points, whereas $y_1$ and $y_2$ are the respective Y-coordinates of the given points. The operator determines the quadrant of the circle in which the first and second points are located, according to the path's starting point, by applying Eq. (3) between the start point and the first point, and between the start point and the second point.

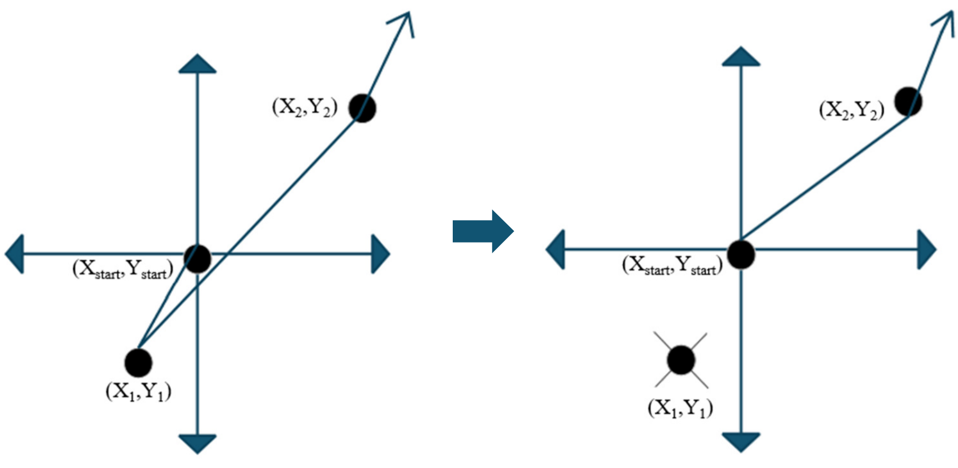
(1) If the two points lie in the same quadrant, the operator performs no modification, as illustrated in Fig. 5.




(a) Before the enhanced operator (b) After the enhanced operator

Fig. 5 First case: the first and second points are located in the same quarter

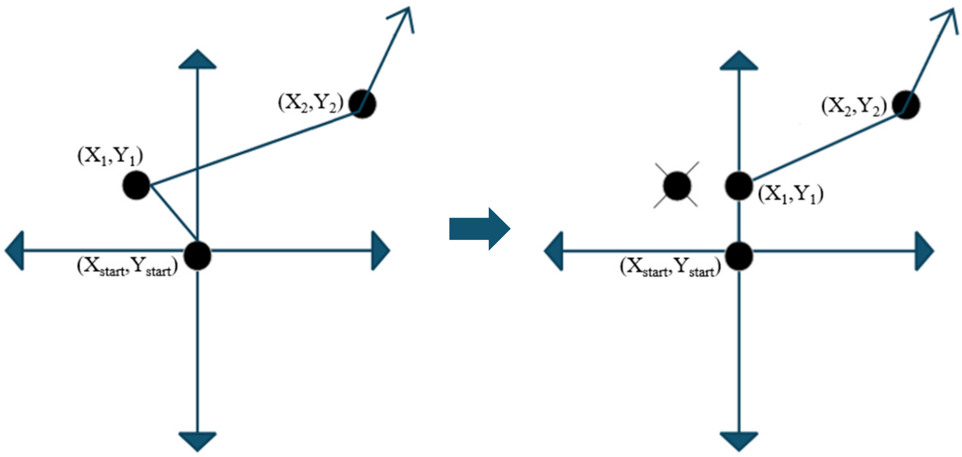
(2) If the two points lie in opposite quadrants, the operator removes the first point and connects the start point to the second point, since no obstacles are present in this region, as depicted in Fig. 6.




(a) Before the enhanced operator (b) After the enhanced operator

Fig. 6 Second case: the first and second points are located in opposite quarters

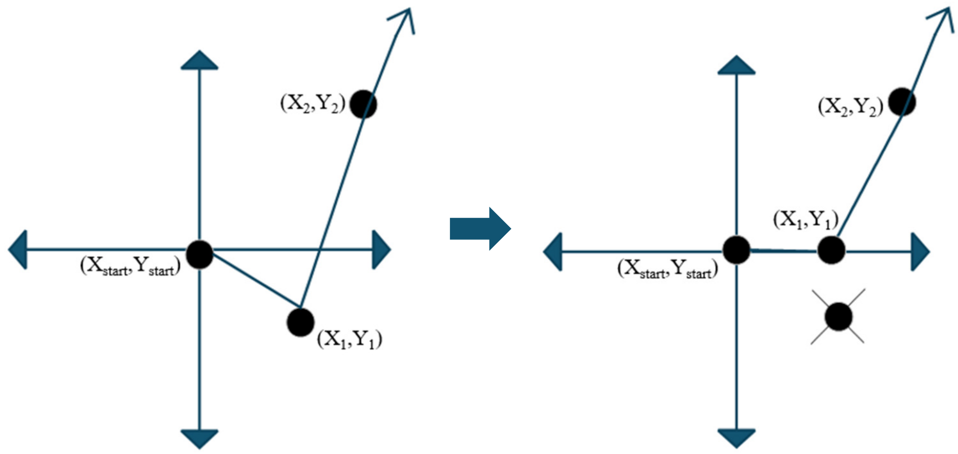
(3) If the two points are located in horizontally adjacent quadrants, the operator removes the first point and generates a new point, where $X_2$ will is set equal to $X_{start}$ and $Y_2$ is kept the same, so the first becomes $(X_{start}, Y_2)$. This adjustment accounts for the potential presence of an obstacle between the start point and the second point, as shown in Fig. 7.




(a) Before the enhanced operator (b) After the enhanced operator

Fig. 7 Third case: the first and second points are located on horizontally adjacent quarters

(4) If the two points lie in vertically adjacent quadrants, the operator deletes the first point and generates a new point, where $Y_2$ is set equal to $Y_{start}$ and $X_2$ is kept the same as the original value. Thus, the new first point becomes $(X_2, Y_{start})$. This adjustment accounts for the potential presence of an obstacle between the start point and the second point, as illustrated in Fig. 8.




(a) Before the enhanced operator (b) After the enhanced operator

Fig. 8 Four cases: the first and second points are located on vertically adjacent quarters

The enhanced operator is also applied to the target. If the path contains a single target, it operates on that target; if multiple targets exist, it is applied to all of them. In this case, the four cases are evaluated for the last two points preceding each target, but in the opposite direction. As a result, the produced path becomes smoother and more suitable for real-world applications. The steps of the enhanced operator are summarized in Pseudocode 1.

| Pseudocode1: an enhanced operator |
| --- |

Function-enhanced path
1. Initialize p = path
2. Calculate leg = Length(p)
3. S = p(start point), p1 = p(first point), p2 = p(second point)
4. If leg > 6:
   - Calculate th1 = angle between S and p1

- Calculate th2 = angle between S and p2
- Determine the quadrant of th1 and th2
- If th1 and th2 are in the same quadrant:
  - No modification
- Else if th1 and th2 are in opposite quadrants:
  - Delete p1
- Else if th1 and th2 are in different quadrants:
  - Modify p1 according to the quadrant difference
    - Delete p1
    - Update x-coordinate of p1
    - Update y-coordinate of p1

4. If leg > 5 (after modification):

T = p(target), p1 = p(target-1), p2 = p(target-2)
- Calculate th1 = angle between T and p1
- Calculate th2 = angle between T and p2
- Determine the quadrant of th1 and th2
- If th1 and th2 are in the same quadrant:
  - No modification
- Else if th1 and th2 are in opposite quadrants:
  - Delete p1
- Else if th1 and th2 are in different quadrants:
  - Modify p1 according to the quadrant difference
    - Delete p1
    - Update x-coordinate of p1
    - Update y-coordinate of p1

5. Return the modified path p

(b) Selection operator

The selection operator is the most important factor influencing the performance of a GA. It is based on the principle of 'survival of the fittest' [10], where individuals with higher fitness values have a greater probability of being selected for the next generation. Several selection methods exist; this work adopts roulette wheel selection. Wheel selection is a mechanism in which individuals (paths) from the population are chosen for reproduction according to their fitness scores. The higher the fitness, the greater the probability of being selected.

(c) Crossover operator

The crossover operator enables two parent chromosomes to exchange information, producing two offspring for the next generation. Each path in this work may contain multiple target nodes arranged in a random order, in addition to a start node. In this case, the path must be divided into segments corresponding to the number of targets before performing the crossover operation. A one-point crossover is then carried out between the segments of the two chosen paths if they belong to the first or second case, as described by Alabbadi and Kanan [25]. Finally, the segments of the newly created paths are merged and rearranged according to the paths' original order. In contrast, for paths containing a single target, the one-point crossover is performed directly without segment division or merging.

(d) Mutation operator

Mutation is considered the most important operator in GA, as it introduces genetic diversity into the population and helps prevent convergence to local optima by exploring the solution space [25]. In this work, the mutation operator randomly selects a node from the path, excluding the start and target points, and generates several nodes around the chosen node. The operator then selects the candidate nodes that produce the best fitness value for the path.

*3.6. Elitist strategy*

This strategy is crucial and is typically adopted in modern GA algorithms. Since chromosomes may change during crossover or mutation operations, the best chromosome from the previous generation may be lost. To prevent this, an elitist strategy is employed to preserve the best chromosomes across generations. This approach maintains a steady convergence rate and enhances the overall quality of the solutions [26]. There are two main types of elitist strategies:

(1) Strong Elitism: Only the best individual(s) in the current population are preserved, ensuring they are carried forward to the next generation.

(2) Weak Elitism: On the other hand, it retains a specific proportion of the best individuals rather than just the single best one, typically ranging from 5% to 20%.

In this study, 10% of the best chromosomes were preserved, which is sufficient to maintain high-quality solutions without interfering with GA operators in the search for better solutions.

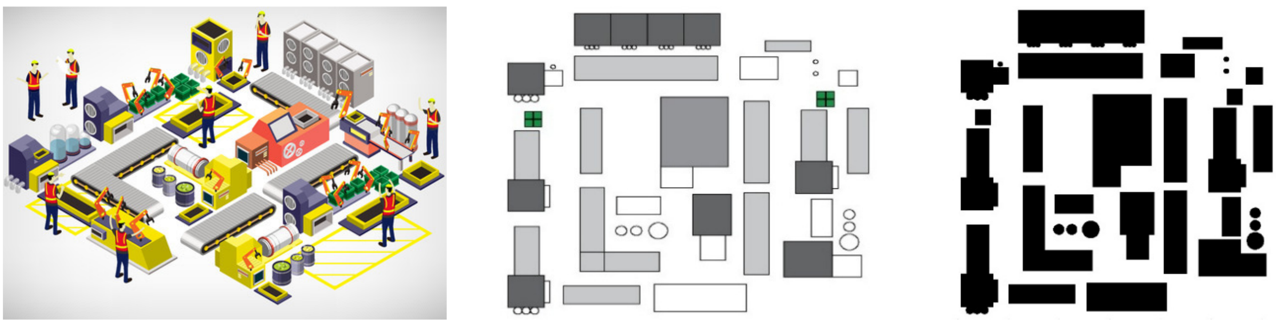
*3.7. Termination condition*

For GAs, there are no universally accepted stopping criteria [27]. Termination typically depends on the nature of the problem and the size of the solution space. In most cases, the algorithm stops either when predefined criteria are satisfied or when the maximum number of generations is reached. In some cases, additional conditions, such as fitness stability or a lack of improvement over successive generations, are also employed alongside the generation limit. The choice of termination condition ultimately depends on the specific objectives and constraints of the problem. In this study, the algorithm terminates when the total number of generations exceeds 100. By fixing the termination condition at 100 generations, the algorithm achieves a balance between computational effort and solution quality. This strategy is effective for solving similar optimization problems.

## 4. Experimental Results

This section presents the experimental results of applying the proposed approach to the path planning problem. Section 4.1 reports the results for a single target, including a performance comparison of the proposed approach with traditional GA [28], the artificial bee colony (ABC) algorithm [29], and enhanced diversity particle swarm optimization (EDPSO) [30]. Section 4.2 presents the results for multiple targets. All experiments were conducted on a laptop computer equipped with a Core (TM) i7-11800H CPU and 16 GB RAM, and implemented in MATLAB.

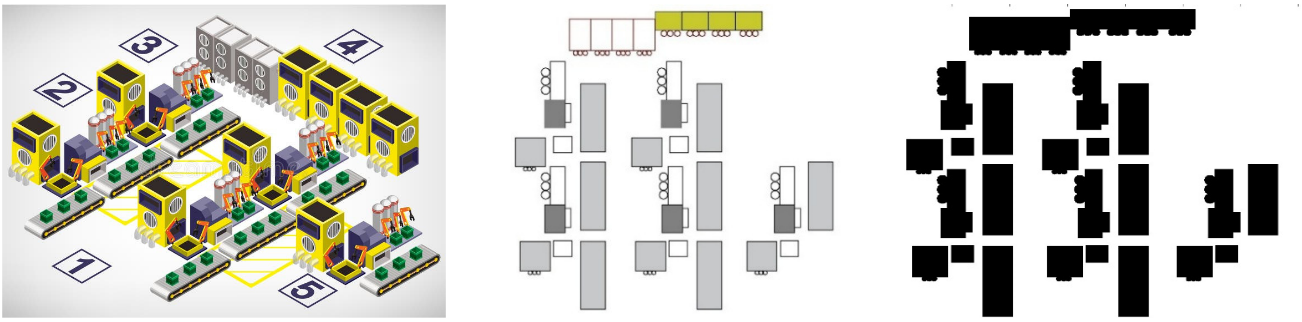
*4.1. Experimental results of the robot's path with a single target*



(a) Original complex map (b) Draw a simple map using AutoCAD (c) Processed complex map

Fig. 9 Processing a complex map by the proposed approach

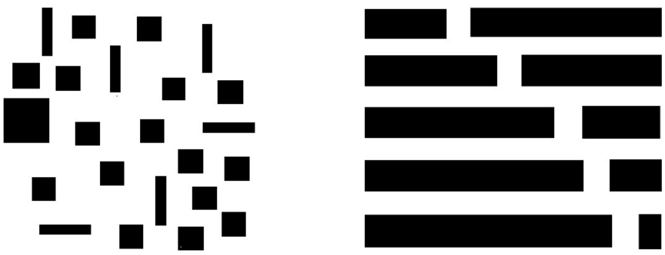
To demonstrate the feasibility and effectiveness of the proposed approach, simulation experiments were conducted on four maps: two realistic maps with different levels of complexity, a cluttered map, and a narrow passage map. The two realistic maps (obtained from online sources) were initially redrawn as 2D layouts using AutoCAD, and subsequently processed and constructed image processing and morphological operations, as illustrated in Fig. 9 and Fig. 10. Fig. 9(a) shows the complex map image, while Fig. 10(a) shows the simple map image. During the initialization stage, the two map layouts were drawn using AutoCAD, and the results were saved as images in the .jpg format, as depicted in Figs. 9(b) and 10(b). Subsequently, the two map images were processed and prepared in a way that makes them recognizable for search algorithms, as shown in Fig. 9(c) for the complex map and Fig. 10(c) for the simple map.



(a) Original simple map (b) Draw a simple map using AutoCAD (c) Processed simple map

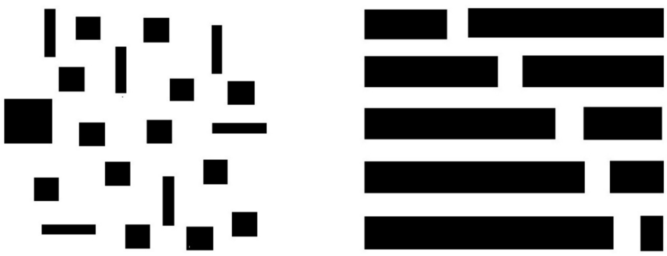
Fig. 10 Processing a simple map by the proposed approach

Fig. 11 shows a cluttered map image and a narrow passage map in .jpg format. Fig. 12 illustrates the corresponding images processed using the proposed approach. Despite differences in size and complexity, the processing time required for each of the four maps is approximately 1.5 seconds. This makes the proposed approach suitable and practical for real-world applications, in contrast to grid-based decomposition, which is typically used for mapping or environment representation. The processing speed of this method is influenced by the map’s complexity, size, and required level of accuracy, all of which significantly affect execution time.



(a) Original cluttered map (b) Original narrow passages map

Fig. 11 The original maps



(a) Processed cluttered map (b) Processed narrow passages map

Fig. 12 Processing the maps by the proposed approach

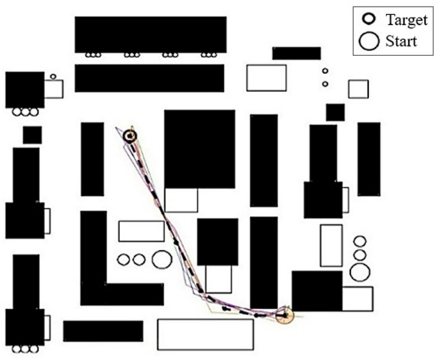
The PRM was subsequently employed to generate feasible paths, producing 1,000 nodes for all simulation experiments across all maps. The number of nodes has a significant impact on computation time. A larger number of nodes expands the search space, increasing the likelihood of finding an optimal path. However, it also substantially increases computational time. Conversely, a smaller number of nodes reduces computation time, but may result in less efficient paths. Therefore, selecting an appropriate number of nodes is essential. The EGA used paths generated by the PRM as its initial population and employs them to produce optimal or near-optimal paths. Table 1 summarizes the control parameters of the EGA.

Table 1 Control parameters of the EGA

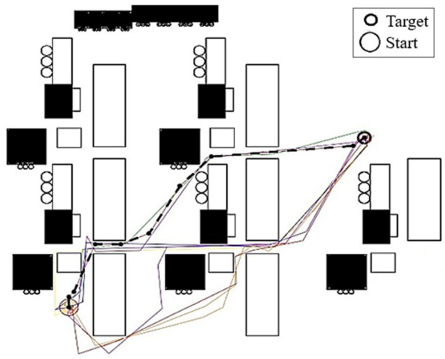
| Parameter | Value |
|---|---|
| Generation No. | 100 |
| Papulation size | 10 |
| Selection probability | 0.33 |
| Crossover probability | 0.85 |
| Mutation probability | 0.01 |

As illustrated in Fig. 13(a), a complex map, Fig. 13(b), a simple map, Fig. 13(c), a cluttered map, and Fig.13(d), a narrow passages map, the large circle represents the starting point, and the small circle represents the target point. The straight lines represent feasible paths generated by the PRM, whereas the dashed line depicts the best solution (path) obtained by EGA.

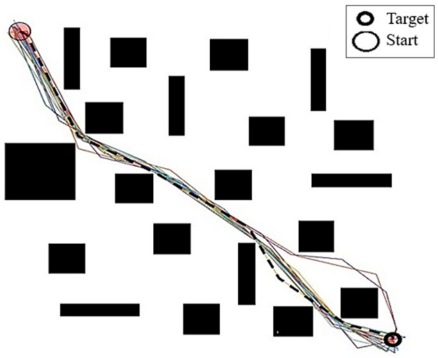
Moreover, as shown in Fig. 14, the simulation process using the PRMEGA algorithm yields a near-optimal path with only 23 iterations, 4 despite the maximum number of iterations being set to 100 for all maps. This result demonstrates the capability of the proposed approach to generate an optimal or near-ideal collision-free path with a relatively small number of generations and offspring.



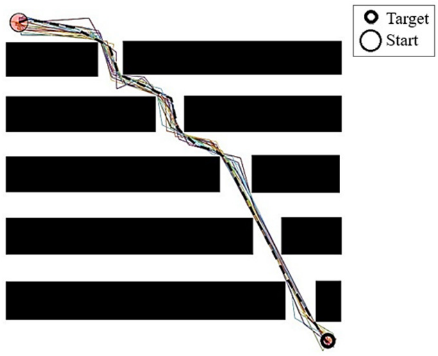
(a) Best path planned on the complex map



(b) Best path planned on the simple map

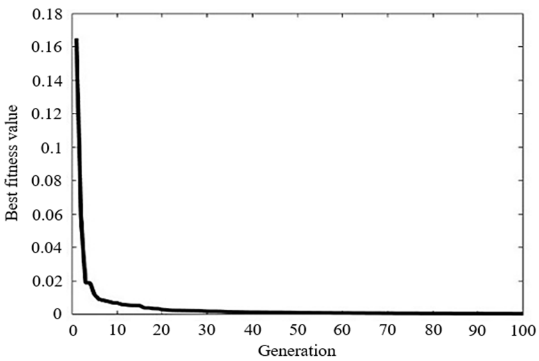


(c) Best path planned on the cluttered map

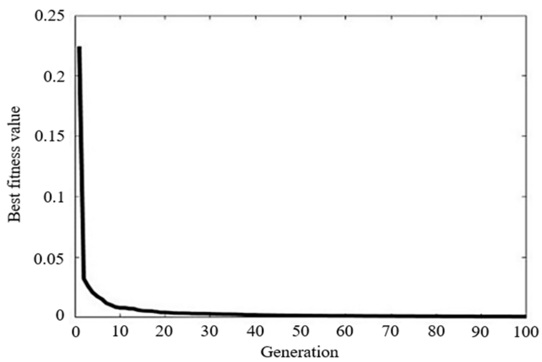


(d) Best path planned on the narrow passages map

Fig. 13 Simulation result of the proposed approach on all maps

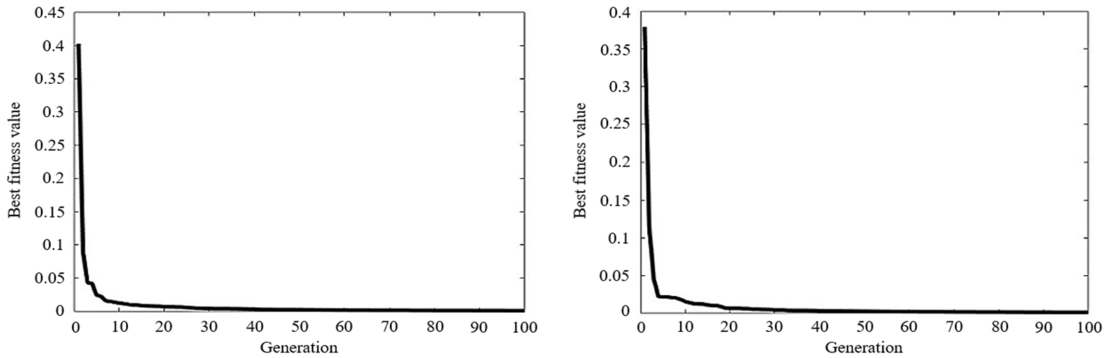


(a) Complex map



(b) Simple map

Fig. 14 Evolution procedure of the proposed approach on different maps




(c) Cluttering map (d) Narrow passages map

Fig. 14 Evolution procedure of the proposed approach on different maps (continued)

To verify the efficiency of the proposed approach, comparisons were conducted with conventional GA [28], ABC [29], and EDPSO [30]. The control parameters for each algorithm are summarized in Table 2. All parameters were adopted directly from the original literature, except for the number of generations and the population size, for which fair values were assigned to ensure a balanced comparison

Table 2 Values of algorithms' parameters

| Algorithm | Parameter | Values |
|---|---|---|
| GA | Population size | 50 |
| | Chromosome length | 10 |
| | Generation No. | 100 |
| | Crossover probability | 0.80 |
| | Mutation probability | 0.15 |
| ABC | Population size of bees | 50 |
| | Iteration No. | 100 |
| | No. of handling points | 5 |
| | Acceleration coefficient | 1 |
| | Abandonment limit parameter | 200 |
| EDPSO | Population size | 50 |
| | Generation No. | 100 |
| | $C_2$, $C_2$ | 0.4, 0.4 |
| | $f_t$ | 2 |
| | $n_s$ | 6 |

Since optimization algorithms yield different numerical results in each run, even under identical parameters and workspace, each algorithm was executed 20 times for path planning on the complex map. For each algorithm, the mean, best, worst, and standard deviation values of the running time and fitness value were calculated. The results are presented in Table 3 and Table 4, respectively. As shown in Table 3, PRMEGA required less execution time to plan a near-optimal path compared with the other algorithms. The results in Table 4 further indicate that the proposed approach achieved the best fitness value, indicating that the planned path was shorter than those generated by the other algorithms.

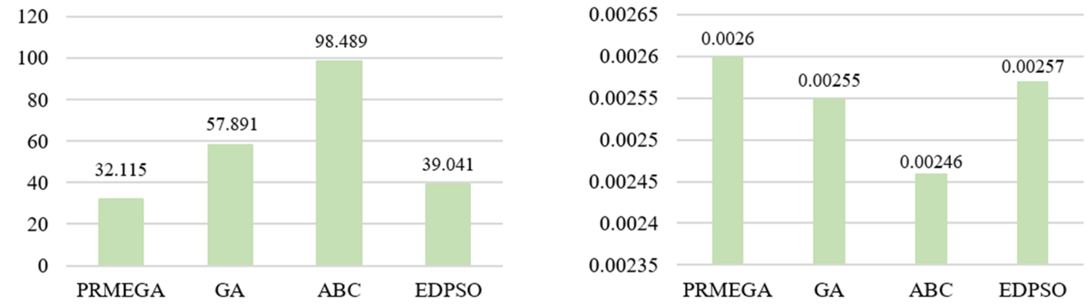
Table 3 Running time comparison

| Parameter (Second) | PRMEGA | GA | ABC | EDPSO |
|---|---|---|---|---|
| Mean running time | 32.115 | 57.891 | 98.489 | 39.041 |
| Best running time | 31.779 | 56.143 | 96.899 | 37.892 |
| Worst running time | 32.937 | 59.031 | 100.021 | 40.122 |

Table 4 Fitness value comparison

| Parameter | PRMEGA | GA | ABC | EDPSO |
|---|---|---|---|---|
| Mean fitness value | 0.00260 | 0.00255 | 0.00246 | 0.00257 |
| Best fitness value | 0.00268 | 0.00266 | 0.00259 | 0.00268 |
| Worst fitness value | 0.00255 | 0.00249 | 0.00240 | 0.00253 |

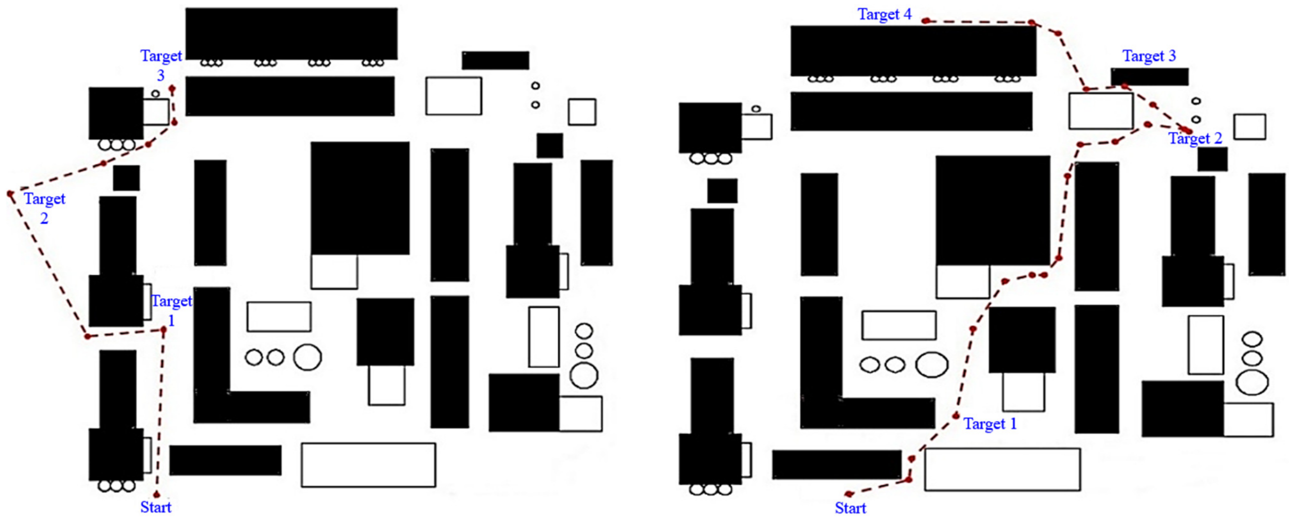
Fig. 15(a) shows that the proposed approach outperforms traditional GA, ABC, and EDPSO in execution time by up to 25.5%. Fig. 15(b) indicates that it achieves a better fitness value than the other three algorithms by up to 14.1%. Although all strategies are capable of generating a collision-free path, the computational time required to obtain them must be considered. The proposed method demonstrates its superiority by building a nearly optimal path in both execution time and fitness.




(a) Mean running time (b) Mean fitness value

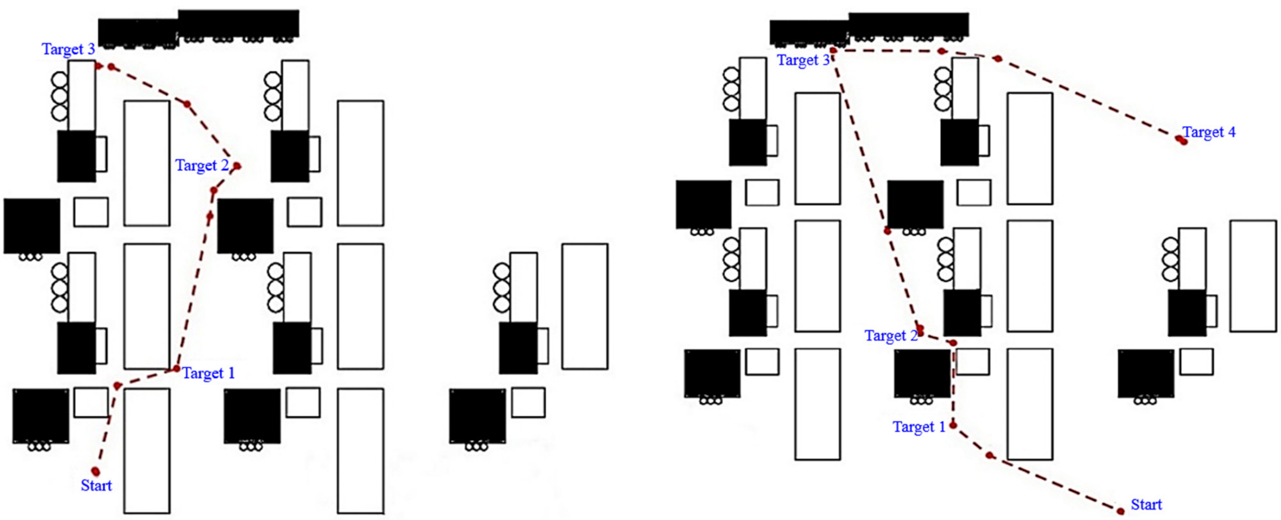
Fig. 15 Performance of PRMEGA, GA, ABC, and EDPSO

### 4.2. Experimental results of the robot's path containing multiple targets




(a) Single start and three targets (b) Single start and four targets

Fig. 16 Path with a single start and multiple targets in a complex map




(a) Single start and three targets (b) Single start and four targets

Fig. 17 Path with a single start and multiple targets in a simple map

In this section, the proposed PRMEGA approach was applied to both complex and simple maps with multiple independent targets, using the same parameter values as specified in Section 4.1. Fig. 16(a) and 16(b) illustrate the optimal paths generated by PRMEGA in a complex map with different numbers of targets, three targets in Fig. 16(a) and four targets in Fig. 16(b). Figs. 17(a) and 17(b) illustrate the optimal paths generated by the proposed approach in a simple map with a single start point and multiple independent targets, three targets in Fig. 17(a), and four targets in Fig. 17(b). The resulting paths successfully visit all targets without colliding with any obstacles.

The experimental results demonstrate the effectiveness of the proposed method in both simple and complex environments with varying numbers of independent targets. These findings indicate that PRMEGA is scalable with respect to the number of targets, the size of the environment, and its complexity. However, as the number of targets increases, PRM requires more time to plan routes between them and to supply EGA with the necessary information to find the optimal or near-optimal path from the start point to all targets. Nevertheless, it remains more computationally efficient than planning routes sequentially from the start point to the nearest target and then to the next one.

## 5. Conclusion

This study introduces PRMEGA, a new algorithm designed to plan collision-free, near-optimal paths from a starting point to one or multiple independent targets in realistic environments with obstacles. The proposed approach significantly improves the efficiency of existing GA-based techniques.

(1) Concerning environmental representation, PRMEGA relies on morphological procedures and image processing to represent the environment. This approach eliminates the typical trade-off between accuracy and speed found in conventional cell-based methods.

(2) A novel method is introduced to generate an initial population composed entirely of collision-free paths, which expedites the evolutionary process and allows finding a nearly optimal path to one or multiple targets in fewer generations and offspring.

(3) An enhanced operator is developed to smooth paths and improve their quality. Simulation results demonstrate that the proposed approach successfully and efficiently plans collision-free, near-optimal paths for both single and multiple independent targets.

The results of PRMEGA are compared with those obtained by traditional GA, ABC, and EDPSO. The results demonstrate that PRMEGA greatly outperforms the other approaches by producing smoother, shorter paths and reducing runtime by 14.1%. Future work should include additional comparisons based on various criteria and diverse environments. Moreover, the proposed approach can be further developed to handle unknown environments or dynamic obstacles and targets.

## Conflicts of Interest

The authors declare no conflict of interest.

## References

[1] S. Lu and C. Guo, “Improving Sensing Measurements Using Laser Self-Mixing Interference in Non-Line-of-Sight Optical Communication via Systems,” HighTech and Innovation Journal, vol. 5, no. 4, pp. 1038-1054, 2024.

[2] J. Cornejo, J. Cornejo, M. Vargas, M. Carvajal, P. Perales, G. Rodríguez, et al., “SY-MIS Project: Biomedical Design of Endo-Robotic and Laparoscopic Training System for Surgery on the Earth and Space,” Emerging Science Journal, vol. 8, no. 2, pp. 372-393, 2024.

[3] K. Puentes, L. Morales, D. F. Pozo-Espin, and V. Moya, “Enhancing Control Systems with Neural Network-Based Intelligent Controllers,” Emerging Science Journal, vol. 8, no. 4, pp. 1243-1261, 2024.

[4] J. E. Solanes and L. Gracia, “Mobile Robots: Trajectory Analysis, Positioning and Control,” Applied Sciences, vol. 15, no. 1, article no. 355, 2025.
[5] J. Cui, L. Wu, X. Huang, D. Xu, C. Liu, and W. Xiao, “Multi-Strategy Adaptable Ant Colony Optimization Algorithm and Its Application in Robot Path Planning,” Knowledge-Based Systems, vol. 288, article no. 111459, 2024.
[6] B. Lacevic and D. Osmankovic, “Improved C-Space Exploration and Path Planning for Robotic Manipulators Using Distance Information,” IEEE International Conference on Robotics and Automation, pp. 1176-1182, 2020.
[7] Z. Wu, J. Dai, B. Jiang, and H. R. Karimi, “Robot Path Planning Based on Artificial Potential Field with Deterministic Annealing,” ISA Transactions, vol. 138, pp. 74-87, 2023.
[8] X. Diao, W. Chi, and J. Wang, “Graph Neural Network Based Method for Robot Path Planning,” Biomimetic Intelligence and Robotics, vol. 4, no. 1, article no. 100147, 2024.
[9] S. Lin, A. Liu, J. Wang, and X. Kong, “An Intelligence-Based Hybrid PSO-SA for Mobile Robot Path Planning in Warehouse,” Journal of Computational Science, vol. 67, article no. 101938, 2023.
[10] R. Sarkar, D. Barman, and N. Chowdhury, “Domain Knowledge Based Genetic Algorithms for Mobile Robot Path Planning Having Single and Multiple Targets,” Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 7, pp. 4269-4283, 2022.
[11] M. N. Ab Wahab, A. Nazir, A. Khalil, W. J. Ho, M. F. Akbar, M. H. M. Noor, et al., “Improved Genetic Algorithm for Mobile Robot Path Planning in Static Environments,” Expert Systems with Applications, vol. 249, part C, article no. 123762, 2024.
[12] H. Heng and W. Rahiman, “ACO-GA-Based Optimization to Enhance Global Path Planning for Autonomous Navigation in Grid Environments,” IEEE Transactions on Evolutionary Computation, pp. 1-15, 2025.
[13] J. Li, Y. Hu, and S. X. Yang, “A Novel Knowledge-Based Genetic Algorithm for Robot Path Planning in Complex Environments,” IEEE Transactions on Evolutionary Computation, vol. 29, no. 2, pp. 375-389, 2025.
[14] L. Liu, X. Wang, X. Yang, H. Liu, J. Li, and P. Wang, “Path Planning Techniques for Mobile Robots: Review and Prospect,” Expert Systems with Applications, vol. 227, article no.120254, 2023.
[15] S. Bandi and D. Thalmann, “Space Discretization for Efficient Human Navigation,” Computer Graphics Forum, vol. 17, no. 3, pp. 195-206, 1998.
[16] H. Mahjoubi, F. Bahrami, and C. Lucas, “Path Planning in an Environment with Static and Dynamic Obstacles Using Genetic Algorithm: A Simplified Search Space Approach,” IEEE International Conference on Evolutionary Computation, pp. 2483-2489, 2006.
[17] S. Alarabi, C. Luo, and M. Santora, “A PRM Approach to Path Planning with Obstacle Avoidance of an Autonomous Robot,” 8th International Conference on Automation, Robotics and Applications, pp. 76-80, 2022.
[18] Q. Li, Y. Xu, S. Bu, and J. Yang, “Smart Vehicle Path Planning Based on Modified PRM Algorithm,” Sensors, vol. 22, no. 17, article no. 6581, 2022.
[19] Y. Hu and S. X. Yang, “A Knowledge Based Genetic Algorithm for Path Planning of a Mobile Robot,” IEEE International Conference on Robotics and Automation, pp. 4350-4355, 2004.
[20] Y. Huang, H. Wang, L. Han, and Y. Xu, “Robot Path Planning in Narrow Passages Based on Improved PRM Method,” Intelligent Service Robotics, vol. 17, no. 3, pp. 609-620, 2024.
[21] M. Yu, Q. Luo, H. Wang, and Y. Lai, “Electric Logistics Vehicle Path Planning Based on the Fusion of the Improved A-Star Algorithm and Dynamic Window Approach,” World Electric Vehicle Journal, vol. 14, no. 8, article no. 213, 2023.
[22] J. Liu, M. Fu, A. Liu, W. Zhang, and B. Chen, “A Homotopy Invariant Based on Convex Dissection Topology and a Distance Optimal Path Planning Algorithm,” IEEE Robotics and Automation Letters, vol. 8, no. 11, pp. 7695-7702, 2023.
[23] G. Chen, Y. Du, X. Xi, K. Zhang, J. Yang, L. Xu, et al., “Improved Genetic Algorithm Based on Bi-Level Co-Evolution for Coverage Path Planning in Irregular Region,” Scientific Reports, vol. 15, no. 1, article no. 10047, 2025.
[24] T. Feng, J. Li, H. Jiang, S. X. Yang, P. Wang, Y. Teng, et al., “The Optimal Global Path Planning of Mobile Robot Based on Improved Hybrid Adaptive Genetic Algorithm in Different Tasks and Complex Road Environments,” IEEE Access, vol. 12, pp. 18400-18415, 2024.
[25] A. Alabbadi and A. Kanan, “Genetic Algorithm-Based Path Planning for Autonomous Mobile Robots,” IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology, pp. 177-180, 2023.
[26] Z. Yao and Y. Xu, “An Improved Genetic Algorithm for Robot Path Planning,” Journal of Computational Methods in Sciences and Engineering, vol. 24, no. 3, pp. 1331-1340, 2024.
[27] C. A. Murthy and N. Chowdhury, “In Search of Optimal Clusters Using Genetic Algorithms,” Pattern Recognition Letters, vol. 17, no. 8, pp. 825-832, 1996.

[28] H. Qu, K. Xing, and T. Alexander, "An Improved Genetic Algorithm with Co-Evolutionary Strategy for Global Path Planning of Multiple Mobile Robots," Neurocomputing, vol. 120, pp. 509-517, 2013.
[29] D. Agarwal and P. S. Bharti, "Implementing Modified Swarm Intelligence Algorithm Based on Slime Moulds for Path Planning and Obstacle Avoidance Problem in Mobile Robots," Applied Soft Computing, vol. 107, article no. 107372, 2021.
[30] P. B. Fernandes, R. C. L. Oliveira, and J. V. Fonseca Neto, "Trajectory Planning of Autonomous Mobile Robots Applying a Particle Swarm Optimization Algorithm with Peaks of Diversity," Applied Soft Computing, vol. 116, article no. 108108, 2022.