JOURNAL OF SOFT COMPUTING AND DATA MINING
e-ISSN: 2716-621X

JSCDM

Vol. 6 No. 3 (2025) 183-199
https://publisher.uthm.edu.my/ojs/index.php/jscdm

Intelligent Trajectory Planning Algorithm for Autonomous Mobile Robots

Shaymaa Alzubairi[1]*, Alexander Petunin[2]

[1] *Institute of New Materials and Technologies, Department of IT and Computer-Aided Design, Laboratory of Optimal Cutting of Industrial Materials and Optimal Routing Technologies, Ural Federal university, Mira Street, 19, Yekaterinburg, 620002, RUSSIA*
[2] *Institute of New Materials and Technologies, Department of IT and Computer-Aided Design, Faculty of Machine Building, Laboratory of Optimal Cutting of Industrial Materials and Optimal Routing Technologies, Ural Federal University, Mira Street, 19, Yekaterinburg, 620002, RUSSIA*

*Corresponding Author: Shaymaaalzubairi77@gmail.com
DOI: https://doi.org/10.30880/jscdm.2025.06.03.013

Article Info

Received: 21 August 2025
Accepted: 12 December 2025
Available online: 30 December 2025

Keywords

Autonomous mobile robot, AMR, trajectory planning, nonholonomic wheeled mobile robot, genetic algorithm

Abstract

Given the widespread utilization of autonomous mobile robots in the past two decades, the path planning problem has received considerable attention from researchers. The goal is to find a collision-free path from a start point to a target point in an environment full of obstacles whilst satisfying some criteria, such as time, distance, and safety.
This study proposes an intelligent hybrid optimisation method for autonomous mobile robots (AMRs) to reach a target or multiple targets in environments filled with obstacles. The proposed method combines two algorithms. The first algorithm is the probabilistic roadmap algorithm, which tries to explore the search space and generate several free collision paths to be utilised as a population for the second algorithm. The second algorithm is the enhanced genetic algorithm (EGA), which is utilised to plan the shortest and smoothest path depending on the population generated by PRM. The EGA presents an effective and accurate fitness function, improves the genetic operators of a conventional genetic algorithm (GA), and proposes a new genetic modification operator.The efficiency of the proposed method is verified by several simulations in various environments and real-time experiments on a nonholonomic wheeled mobile robot. Results show that the proposed hybrid approach outperforms the conventional GA, probabilistic roadmap, ant colony optimisation, the Bezier smoothing algorithm, and other algorithms in planning a smooth, near-optimal, collision-free path in competitive time.

1. Introduction

Autonomous mobile robots are utilised in various applications such as danger seeking, exploration, security patrol, mining, space missions, and education [1], [2]. Requirements for AMR navigation include path planning, localizing the robot location, and describing the robot environment [3]. AMR path planning entails determining a feasible path between the starting point and the target point, or several targets, taking into account a variety of objectives, including avoiding obstacles, determining the shortest route, reaching the target in the shortest amount of time, using less energy, and achieving path smoothness [4].

Based on the availability of prior knowledge about the environment, robot path planning may generally be classified into two types: offline and online. Online path planning does not have any prior knowledge of the environment; instead, the robot learns it through a variety of sensors whenever the environment changes. In offline path planning, on the other hand, the environment is known beforehand. Further, robot path planning can be divided into two categories based on the type of obstacles: (1) planning in a static environment, where the obstacles are fixed during the entire process, and (2) planning in a non-static environment. For example, throughout time, the existing obstacles may shift positions. These kinds of adjustments take place in a dynamic setting [5].

Furthermore, the path of AMR may contain single or multiple targets. Multiple targets can be of three types: (a) dependent targets, (b) independent targets, and (c) a combination of types a and b, where part of the targets are dependent on each other while the rest are independent. Substantial research has focused on the path planning problem, and various solutions have been produced. Path planning problem solutions can be classified as traditional methods, such as cell decomposition methods and roadmap approaches. The approaches were widely used because artificial intelligence technologies were not yet developed. Using these techniques might or might not resolve the issue of path planning [6]. Another classification is graph search methods [7], such as the Dijkstra algorithm, the A* algorithm, and the Dhouib-Matrix-SPP (DM-SPP) method[8]. DM-SPP provides quick and precise path planning, producing the shortest paths in any graph without collisions with obstacles. Improved performance and capabilities are provided by DM-SPP-24, an upgraded version designed to operate with 24 movement directions in square-grid situations with static impediments. However, these methods have drawbacks that affect how well they perform in complicated environments, such as high computational resource requirements and limited adaptability. Over time, path planning techniques have been expanded by optimization strategies that focused on reducing the AMR's travel distance, time, energy consumption, and other criteria, leading to the development of heuristic methods [9], the ant colony algorithm, the grey wolf algorithm, and the bee algorithm, amongst many others. Each method offers advantages over the others in specific aspects.

For the last two decades, the use of genetic algorithms (GAs) for path planning problems has gained widespread interest. Compared with traditional search and optimisation methods, the GA is a powerful parallel search algorithm based on the mechanism of natural selection and uses operations of reproduction, crossover, and mutation on a population of solutions to find an optimum path in a large workspace. A GA based on an effective motion planning technique for the static environment with a polygonal obstacle has been proposed in [10]. [11] developed a mobile robot using a GA-based search technique for real-time path planning in a planned terrain. Multiple objective optimisation problems by GA are solved in [12]. [13] enhanced GA for MRN across a wide range of search spaces in order to accomplish a multi-objective task. [14] introduced an enhanced GA for mobile robots' global path planning. It employed the coevolution methods to get several mobile robots to cooperate. A novel mutation operator for the GA was presented in [15] and used to solve the mobile robots' path planning problem in dynamic environments. The path planning problem with single and multiple independent targets has been proposed to be solved by domain knowledge-based GAs in [16]. Despite their benefits, GAs still have problems with blind population creation, slow speed, prematurity leading into the local optimum path, and mass computation.

Aiming at overcoming these shortcomings, this study presents a modified GA hybrid with PRM for planning optimal or near-optimal paths between a single start and a single target or multiple independent targets. This proposed approach adopts a new method to generate the initial population, which can generate paths that are diverse and feasible by utilizing PRM. In addition, it uses three traditional genetic operators (selection operator, crossover operator, and mutation operator) and a fitness function to find the best path from the population. Some changes to the mutation operator have been performed to avoid prematurity. The crossover operator was adapted for paths with multiple targets. An enhanced delete operator was developed to smooth the produced path. Furthermore, the proposed hybrid algorithm was applied to a nonholonomic wheeled mobile robot to navigate from the start destination to the final destination through various obstacle-filled environments.

The remainder of the paper is structured as follows: Section 2 describes the problem. Section 3 describes the ARM modelling. Section 4 introduces the proposed methodology. Section 5 analyses the experimental findings. Section 6 concludes the study and presents suggestions for future work.

2. Problem Description

In this study, two problems are addressed. AMR path planning with a single target is the first type, in which every potential path is made up of a start node, a target node, and a number of intermediate nodes. The second is AMR path planning for multiple independent targets, in which a potential path consists of at least one start node, several independent targets, and a few intermediate nodes between the start node and the first target node, as well as between any two target nodes in succession. Because the target nodes are independent of one another, they can also appear in any order. Finding the optimal or near-optimal path is the objective, and the robot's distance travelled, time, and path smoothness are all considered optimization criteria.

The following assumptions were considered during research:

1. All environments are static.
2. All environmental information is assumed to be known beforehand.
3. In this study, the mobile robot has been viewed as a point, and the obstacle boundary is made up of the real boundaries of the obstacles as well as the minimal distance required for the robot to be safe.
4. The robot's velocity is fixed.

3. Autonomous Mobile Robot Modelling

Drive systems can be divided into holonomic and nonholonomic drives according to the robot's movement [17]. Holonomic and nonholonomic drives are the relation between a robot's-controlled degree of freedom and its overall degrees of freedom. A robot with a holonomic drive has controlled degrees of freedom equal to its total number of degrees of freedom. Similarly, the robot used in this work is referred to as having a nonholonomic drive if its controllable degrees of freedom are less than its total degrees of freedom. Figure 1 shows a wheeled AMR system that is nonholonomic.

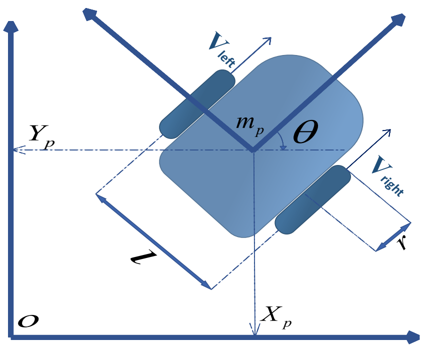


Fig. 1 *Nonholonomic wheeled AMR*

For the stability of the AMR, it has one caster wheel at the front or rear of the platform and two drive wheels on the same axis. Two separate analogue direct current motors are used to drive the wheeled robot's left and right wheels for platform steering and motion. The two driving wheels are attached to the axis center, and the AMR center mass is situated at point (m_p)[18]. As the AMR model is a multi-input multi-output system, it has two input states (the velocities of the left and right wheels), but three output states depending on where it is in the global coordinate frame [θ, X-axis, and Y-axis]; the pose surface is X_p and Y_p representing the point's coordinates m_p. In addition to having an underactuated model, the AMR platform's kinematics equation features output states that are highly nonlinear and time-variant. It is therefore possible to determine the configuration of the mobile robot using these three generalized coordinates. Conclusion The following are the equations for the computer simulation [19].

$$X_p(k) = \left[\frac{1}{2}(V_{left} + V_{right}) \times \cos(\theta(k)) \times T\right] + X_p(k-1), \tag{1}$$

$$Y_p(k) = \left[\frac{1}{2}(V_{left} + V_{right}) \times \sin(\theta(k)) \times T\right] + Y_p(k-1), \tag{2}$$

$$\theta(k) = \left[\frac{1}{l}(V_{left} + V_{right})T\right] + \theta(k-1), \tag{3}$$

Parameters of equations and their definitions	
V_{Right}	Right wheel velocity
V_{Left}	Left wheel velocity
r	Radius of the left and the right wheels
L·	Distance between the two wheels
X_p	Coordinate of the point m_p on X-axis
Y_p	Coordinate of the point m_p on Y-axis
θ	Orientation of the motion of the point m_p
T	Sampling time of the mathematical calculation

4. Proposed Method: Hybrid PRM and Enhanced GA

4.1 Representation of Environment

Traditional GAs utilize cell-based methods to represent environments. These methods require a delicate balance between speed and accuracy, which is time consuming, especially in large and complex environments. In this work, image processing and morphological operations were used to read variant maps with different sizes and degrees of complexity to compose the environments for search algorithms that were used for robot path planning [20], i.e., PRM and enhanced genetic algorithm (EGA). Pseudocode 1 describes the steps for map processing.

```
Pseudocode1
Input: 2D map image
Output: Processed map
FUNCTION ProcessImage (2D map image):
  img = LoadImage (2D map image)
  grayscale_img = ConvertImageToGrayscale(img)
  binary_img = ConvertImageToBinary(grayscale_img)
  objects_in_img =FindObjects(binary_img)
  refined_img = ApplyErosion(objects_in_img, kernel_size)
  refined_img = ApplyDilation(refined_img, kernel_size)
  outboard = FindOuterBoundaries(refined_img)
  inbord = FindInnerBoundaries(refined_img)
  object_borders= SubtractImages (outbord, inbord)
  features_map = ExtractRegionFeatures(refined_img,object_borders)
  occupancy_map = ConvertToOccupancyMap(features_map)
  RETURN occupancy_map
END FUNCTION
```

4.2 Path Representation (Chromosome Representation)

Potential solutions to a problem are encoded as chromosomes. These chromosomes form a population of the GAs. In path planning problems, the potential solutions are paths. In the proposed method, a path can either contain a single start and a single target, as shown in Fig. 2(a), or a single start and a number of n targets that can appear in any order, as shown in Fig. 2(b). Paths are of different lengths. Here, the genes of the chromosome are path nodes represented as Cartesian floating points.

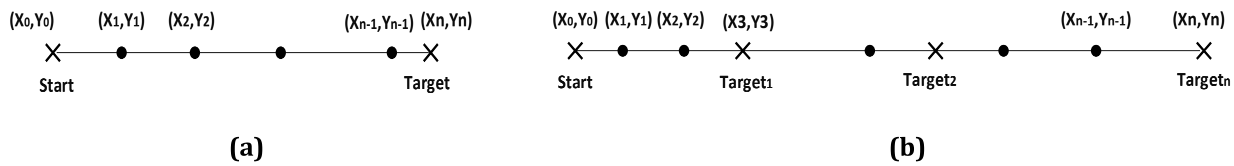


Fig. 2 *Figure description (a) Path with a single start and a single target; (b) Path with a single start and multiple targets*

4.3 Initial Population

PRM has been used to create feasible paths [21-22]. To build paths that would later be used as the GA population, PRM has been run several times with varying numbers of nodes. When there is only one target, PRM creates paths between the start point and the target point. However, when there are several targets, PRM creates paths between the start point and all targets and between the targets with each other. Then, a special technique is employed to join the paths generated by PRM to create paths that contain the starting point and all target points without duplicating any target in paths. Although the paths are collision free, they may have hard sharps and be unsuitable for real-world applications. Therefore, these paths must be refined using genetic operators to become optimal or near optimal. Pseudocode 2 describes the steps of generating path population.

Pseudocode2
Input: Start point and targets Output: Population paths

FUNCTION population generation (start point, target points):
S = (start point) M= Number (targets) T(m) = (target points)
max= number of paths (population)
Initialize PRM parameters
If M=1
For i=1 to max
For each pair of points (S,T(m))
Find feasible path using PRM and store it in pstore
End
Return pstore
Else
For i=1 to M
For each pair of points (S,T(i))
Find paths between using PRM store path in pstore
End
For j=2 to M-1
For each pair of points (t(j),t(j+1))
Find paths between using PRM store path in pstore
End
For z=1 to max
Connect paths in pstore start with S and contain all T(m) without duplicated and put in pstore1
End
Return pstore1
Endif
End function

4.4 Fitness Function

A fitness function is utilised to assess the quality of paths. Given that the population paths generated by PRM are feasible, attention must be paid to the length and smoothness of the paths. Moreover, given that length and the number of hard angles are regarded as optimisation criteria, the objective function for a path should be minimized, as shown in Eq. (4).

$$\int = \frac{1}{\sum_{i=1}^{n} d_i} + \frac{1}{Q}, \tag{4}$$

where d_i is the path length. The Euclidean distance equation is used to calculate it, as shown in Eq. (5) [23].

$$d_i = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}, \tag{5}$$

In Eq. (5), x_i and x_{i+1} are the X-coordinates of the nodes ith and $(i+1)th$, respectively. Similarly, y_i and y_{i+1} are the Y-coordinates of the nodes ith and$(i+1)th$, respectively. Q is a number of hard angles in the path. The ($CosTheta$) function was used to calculate the angles between the segments of a path (Eqs. [6], [7] and [8]), and the ($acosd$) function was used to calculate the degrees of that angle (Eq. [9]).

$$u = x(i+1) - x(i), \tag{6}$$

$$v = x(i+2) - x(i+1), \tag{7}$$

$$\text{Cos}\,Theta = \frac{dot(u,v)}{(norm(u) * norm(v))}, \tag{8}$$

$$th = a\cos d(\text{Cos}\,Theta), \tag{9}$$

$$Q = \begin{cases} \text{If} \quad th < 90° \quad \text{then} \quad Q = Q + 1 \\ \text{Else} \quad Q = 0 \end{cases} \tag{10}$$

4.5 Operators of GA

The primary genetic operators of traditional GAs—selection, crossover and mutation—are used to simulate the process of natural selection, which gives them their potent search capabilities [24]. The genetic operators are improved in this section to preserve and improve the population's feasible paths and achieve the optimal or near-optimal one. In addition, a delete operator was developed to increase the quality and smoothness of the produced paths. Figure 3 depicts the flowchart for the proposed method.

a. Selection Operator

The most crucial factor that could affect the performance of GAs is the selection operator [16]. Selection operators are based on the principle of 'survival of the fittest'. Amongst various selection techniques, this work employed roulette wheel selection. Wheel selection is the process by which members of a population are chosen for reproduction according to their fitness ratings. Chromosomes with a high fitness value are more likely to be chosen by this operator for the following generation.

b. Crossover Operator

In this study, each path may include a start node and several target nodes arranged in a random order. In this instance, before the crossover operation can be performed, the path must be segmented into the same number of targets. Next, if the segments of the two selected paths fall into the first or second case listed in [23], a one-point crossover is performed between them. Finally, the segments of the newly formed paths are combined and reorganized in accordance with the original sequence of the paths. Conversely, one-point crossing occurs instantly, without division or merging, when a path has a single target.

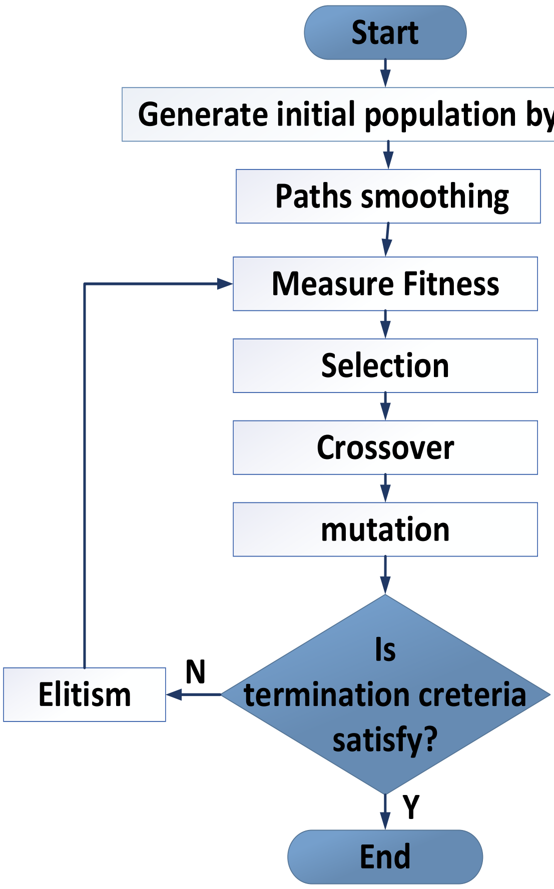


Fig. 3 *Flowchart of the proposed method*

c. Mutation Operator

The mutation operator introduces genetic diversity to the population to explore the solution space and prevent being trapped at local optima [24]. In this work, the mutation operator randomly selects a node from the path, but not the start or target nodes, and generates six points around the selected node, after which the operator selects one of the generated nodes that yields the best fitness value for the path.

d. The Elitist Methodology

Chromosomes may be altered by crossover or mutation; the best chromosome from the prior generation may be lost. Thus, the best chromosomes are passed down through the generations using the elitist method. Approximately 10% of the finest chromosomes are retained in this work [25]. Pseudocode 3 describes the steps for the elitist strategy.

Pseudocode3 Elitist_Strategy
Input: Population_Size, Generation_Count, Elitism_Rate (10%) Output: Best_Solution
Initialize Population using a specific method Determine Pop_Size, Gen_Count, and Elit_Rate For Generation = 1 to Gen_Count Evaluate Fitness of each individual in Population using Fitness Function Elit_Count = Pop_Size * Elit_Rate (10%) Best_Chromosomes = Select_Top(Elit_Count, Population) New_Population = Best_Chromosomes While Size(New_Population) < Pop_Size Parent1 = Select_Parent(Population) using (roulette wheel) Parent2 = Select_Parent(Population) using (roulette wheel) Child = Crossover(Parent1, Parent2) Child = Mutate(Child) Add Child to New_Population Population = New_Population End For

e. Enhanced Delete Operator

PRMEGA produces paths with a high fitness value and no collisions. However, given the randomization of PRM and GA, the paths have sharp turns, especially after the start point and before the target point if the path has a single target and before each target if the path has many targets. This renders the paths unfit for practical use. A delete operator is developed to smooth out the sharp curves and improve the quality of PRMEGA's paths. Focusing on the two nodes following a path's start point and the final two nodes before each target in the path, the operator utilizes the slope equation (Eq. 11).

$$Slope = \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1}, \tag{11}$$

$$\theta = \tan^{-1}\left(\frac{\Delta y}{\Delta x}\right) = \tan^{-1}\left(\frac{y_2 - y_1}{x_2 - x_1}\right), \tag{12}$$

where Δx = change in X, and Δy = change in Y. 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 (Fig. [4]). The delete operator determines in which quarter the first and second nodes are located in accordance with the starting point of the path by applying Eq. (12) between the start point and the first point and the start point and the second point to calculate the value of θ each point, respectively. The four cases are as follows:

Case 1: The delete operator does nothing if the two nodes are in the same quarter.

Case 2: If the nodes are in opposite quarters, the delete operator eliminates the first node and immediately connects the start point to the second node because this region is free of obstructions.

Case 3: When two nodes are located in adjacent horizontal quarters, the delete operator eliminates the first node and generates a new node, where x_2 is equal to x_{start} and y_2 remains unchanged. Because of the possibility of an obstruction between the start node and the second node, the value of the first node is (x_{start},y_2).

Case 4: If the nodes are located in two adjacent vertical quarters, then the operator eliminates the first node and creates a new node, where y_2 is equal to y_{start}, and x_2 remains unchanged. Thus, the value of the first node is (x_2,y_{start}), this arises from the possibility that there is an obstacle between the start node and the second node.

For each target in the path, the same process is carried out, but for the last two nodes before the target, it is done in reverse. The steps of the delete operator are described by Psedocode4.

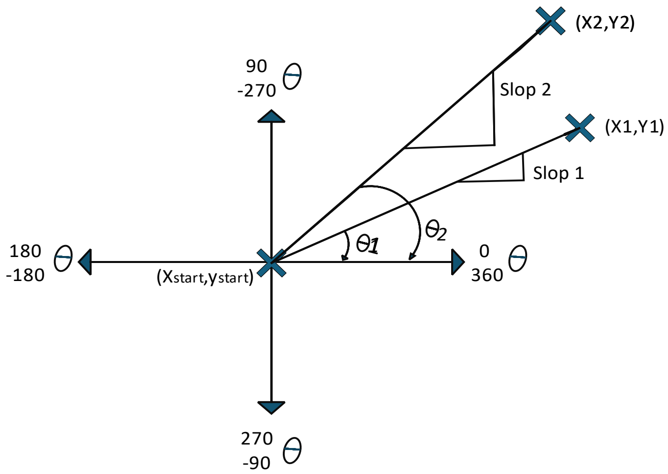


Fig. 4 *First and second point and calculated angle relative to the starting node*

```
Pseudocode4
Input: Path
Output: An Enhanced path
FUNCTION enhanced path (path):
  p = (path)   leg = Length(path)
    if leg > 6
    S = p(start point)  p1= p(first point)    p2= p(second point)
    th1 = angle between( S, p1)  th2 = angle between( S, p2)
    determined quadrant of th1 and th2
    if same quadrant  then  no modification
    elseif opposite quadrant  then  delete p1
   elseif th1 and th2 are different quadrants  then
    Modify (p1) according to the quadrant difference
    (Delete( p1)   Update x-coordinate(p1)    Update y-coordinate(p1))
 leg = Length(path)
 if leg > 5
    L = p(last point)  p1= p(last point-1)  p2= p(last point-2)
    th1 = angle between( L, p1)  th2 = angle between( L, p2)
    determined quadrant of th1 and th2
    if same quadrant then no modification
    elseif opposite quadrant  then delete p1
    elseif  th1 and th2 are different quadrants  then
    Modify (p1) according to the quadrant difference
    (Delete( p1)   Update x-coordinate(p1)    Update y-coordinate(p1))
    Return p
   END FUNCTION
```

4.6 Termination Condition

GA has no general stopping standards. The proposed method stops when the number of generations exceeded 100.

5. Experimental Results

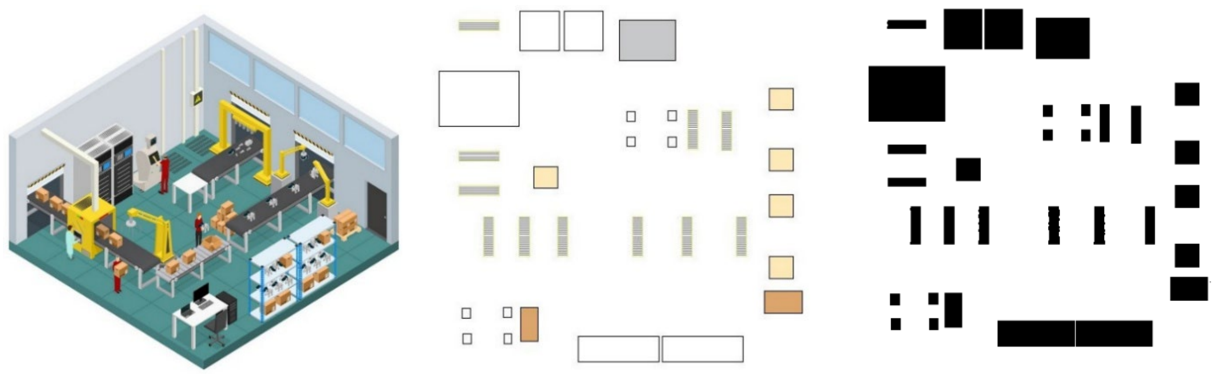
In this section, the efficiency of the proposed approach for the path planning problem with a single target and multiple targets is investigated, and the experimental results are discussed. The experimental results for a single target are described in Section 5.1, along with a performance comparison of the proposed approach versus several algorithms in different cases. Section 5.2 displays the experimental results for multiple targets. All experiments were performed on a laptop computer equipped with a Core (TM) i7-11800H CPU and 16 GB using MATLAB language.

5.1 Results of Experiments for the Robot's Single-Target Path

Previous research works that used different path planning algorithms with varied static environments (simple, complex, cluttered, corridors, real-like) containing diverse obstacle forms and sizes were compared with the proposed approach to verify that it delivered the shortest path in comparative time.

Case 1:

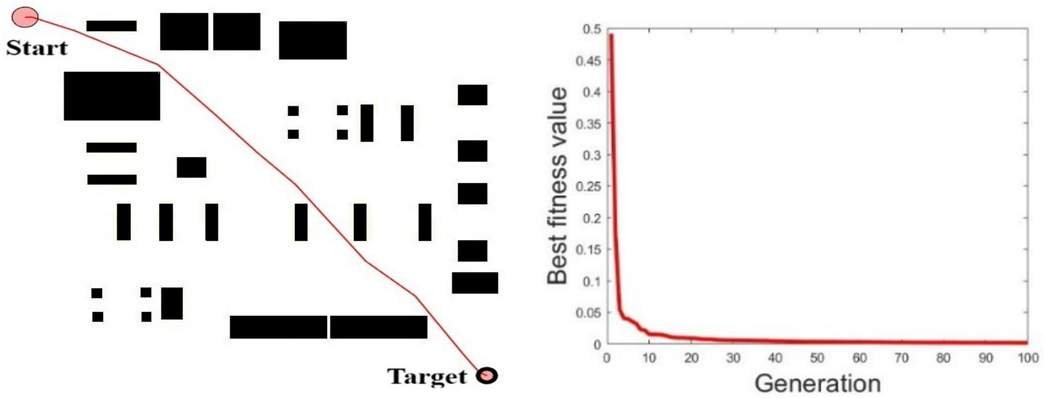
The proposed approach was utilised to plan a feasible path from a start point [20, 20] to a target point [490, 490] in a realistic static environment with an area of [500 × 500] cm. In the beginning, a realistic map image (Fig. 5[a]) was drawn as a 2D image by using an AutoCAD program (Fig. 5[b]), and then it was read and processed in a manner that makes it recognizable for search algorithms by utilizing morphological operations and image processing, as shown in Fig. 5(c).



(a) **(b)** **(c)**

Fig. 5 *Realistic original map (a) Original map image; (b) 2D map image with .jpg extension; (c) Processed image*

Then, the proposed approach planned a path between the start point and the target whilst maintaining a safety space between the robot and the obstacles, as depicted in Fig. 6(a) and the fitness function value in Fig. 6(b).



(a) **(b)**

Fig. 6 *Proposed approach path planning (a) Path planned by PRMEGA; (b) Fitness value*

The result was compared with the results of the PRM and GA algorithms in the same environment to verify the effectiveness of the proposed approach. All algorithms were executed 20 times, and the average values were utilized.

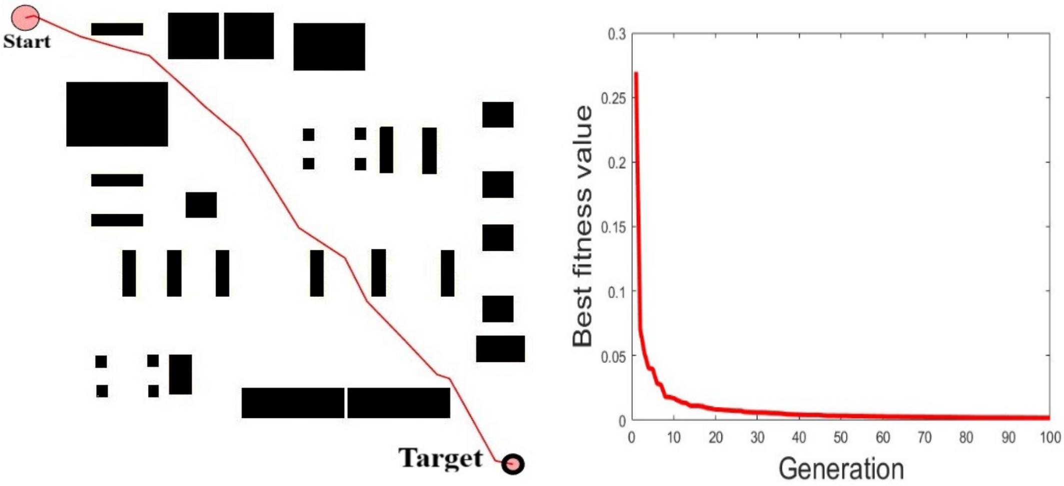
The algorithms’ parameter values are mentioned in Table 1. To have a robust and trusted comparison, equal parameters’ values were assigned to both the proposed approach and GA. To achieve better algorithms' performance, a high crossover rate was assigned to better explore the solutions and a low mutation rate to provide diversity and avoid falling into a local optimum. The population size was chosen to be not so big, which would cause the GA to slow down, and not so small that it might not be enough for a good mating pool. The number of iterations was chosen in the usual range of values.

Computation time is significantly impacted by the number of PRM nodes and edges. An optimal path is more likely to be found when there are more nodes and edges in the search space. But it also significantly lengthens computation time. On the other hand, fewer nodes and edges shorten computation times but may produce fewer effective paths. Consequently, choosing the right number of nodes and edges is crucial.

The averages for the optimal path length of the algorithms, the number of iterations, and the execution time are listed in Table 2. Also, the standard deviations of the methods are mentioned. Fig. 7 depicts the path generated by the traditional GA and the best fitness function. The path generated by PRM is shown in Fig. 8.

Table 1 *Control parameters*

	Parameters		Values	
	Generation No.	Population size	Crossover probability	Mutation probability
Proposed approach	100	100	0.9	0.01
GA	100	100	0.9	0.01
	No. nodes		No. edges	
PRM	850		100	



(a) **(b)**

Fig. 7 *Traditional GA path planning (a) Path planned by GA; (b) Fitness value*

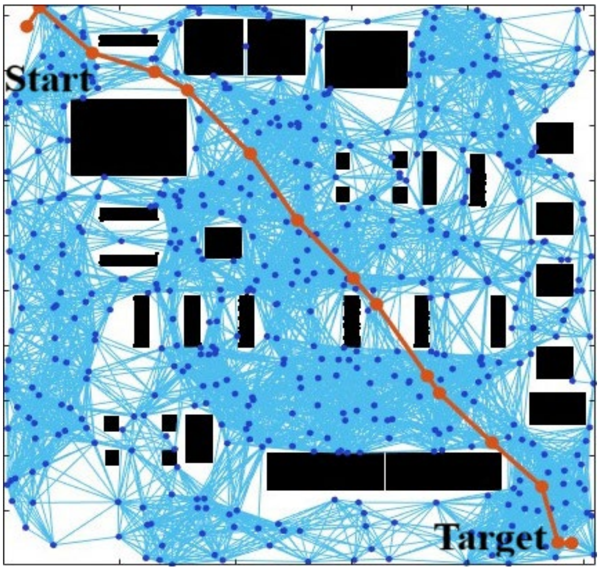


Fig. 8 *Path planned by PRM*

Table 2 *The algorithms' results*

	PRMEGA	GA	PRM
Length (CM)	555.56	556.11	556.37
Iteration No.	27	53	-
Time execution(sec)	11.61	13.96	2.87
The standard deviation	5.01	5.38	5.68

The results of the comparison in Table 2 show the superiority of the proposed approach, which planned a smooth short distance path equal to 555.56 cm in an iteration number of 27 and execution time of 11.61 s. By contrast, the GA algorithm planned a path with an acceptable length equal to 556.11 cm, but it required more iterations and a longer execution time, i.e., 53 and 13.96 s, respectively. Because the proposed approach uses PRM to produce feasible paths, next, it needs to focus only on selecting and refining the best path, increasing its quality; for that, it requires fewer iterations than traditional GA. As for the PRM algorithm, it planned a relatively short path of 556.37 cm with a comparative time of 2.87 seconds, but it had sharp turns, especially after the starting node and before the target node, due to the random nature of the PRM algorithm, making its generated paths unsuitable for real-life applications.

The standard deviation was 5.01 for the proposed approach, 5.38 for the standard algorithm, and 5.68 for the PRM algorithm. The standard deviation of 5.01 indicates that the paths generated by the proposed approach are more convergent and stable than the other two methods.

The path produced by the proposed approach's reference path equation is shown in Eq. (13):

$$y_{ref}(x_{ref}) = -3.153e-13\times x^6 + 6.063e-10\times x^5 - 4.343e-07\times x^4 + 0.0001415\times x^3 - 0.01904\times x^2 + 1.418\times x - 0.8837 \quad (13)$$

$$v_{ref} = \sqrt[2]{(\dot{x}_{ref})^2 + (\dot{y}_{ref})^2}, \quad (14)$$

$$w_{ref} = \frac{(\ddot{y}_{ref}\times\dot{x}_{ref} - \ddot{x}_{ref}\times\dot{y}_{ref})}{((\dot{x}_{ref})^2 + (\dot{y}_{ref})^2)}, \quad (15)$$

The AMR platform's reference linear velocity v_{ref} and reference angular velocity w_{ref} are determined using equations (14) and (15) based on the earlier reference path equation (13). As a result, the linear velocities of the left and right wheels V_L and V_R , and the angular velocities of the right and left wheels W_L and W_R can be calculated as follows:

$$V_R = \frac{(2v_{ref} + Lw_{ref})}{2}, \quad (16)$$

$$V_L = \frac{(2v_{ref} - Lw_{ref})}{2} \quad (17)$$

$$W_R = \frac{(2v_{ref} + Lw_{ref})}{2r}, \quad (18)$$

$$W_L = \frac{(2v_{ref} - Lw_{ref})}{2r}, \quad (19)$$

where r represents the AMR wheel's radius (0.075 m) and the separation distance between the two wheels (0.39 m), with a sampling time of 0.1s. The robot's reference angular and linear velocities are shown in Figure 9. Additionally, the linear velocity of the AMR's right and left wheels is shown in Fig. 10(a), while the robot's right and left wheels' angular velocity is shown in Fig. 10(b).

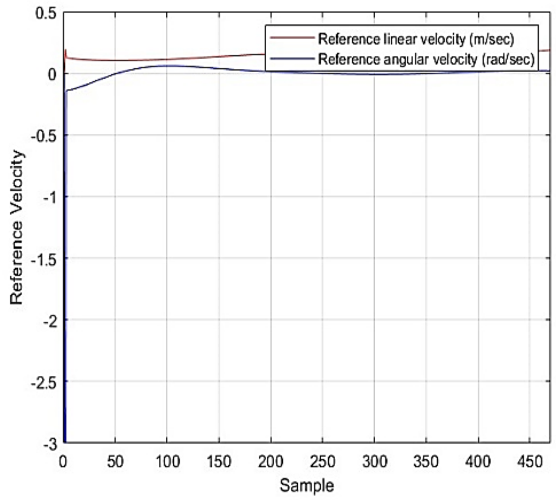
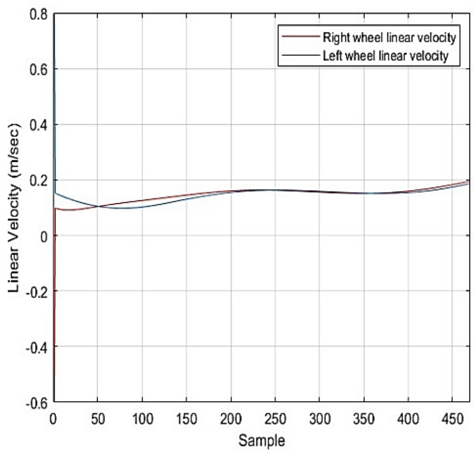
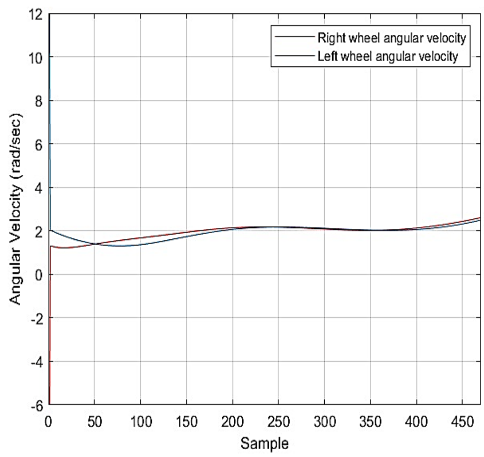


Fig. 9 *The robot's reference angular and linear velocities*



(a)



(b)

Fig. 10 *Left and right wheels' velocities (a) The left and right wheels' linear velocities; (b) The left and right wheels' angular velocities*

Case 2:

To prove that the proposed approach provides a shorter path with a comparative time, the proposed hybrid approach was compared with the ant colony optimization algorithm (ACO), the GA algorithm, ant colony optimization combined with the GA (ACO-GA), the Bezier curve smoothing algorithm (BCA), and the Bezier smoothing algorithm with increased safety distance (BCA-Q) that were proposed in [26], along with their results.

The proposed approach was executed 20 times, and the average result has been calculated, using a cluttered static environment with a workspace of [20 × 20] m, where the starting point was at (0, 0) and the target point was at (20, 20); the AMR was considered a point in the 2D workspace. As given in Table 3, the proposed approach produced a path of length equal to 28.571 m with a time execution of 17.121 s.

While ACO, BCA-Q, and GA planned a feasible path with path lengths of 45.998, 33.663, and 37.545 m, respectively, they are considered longer than that of PRMEGA (28.571 m) because the path contains several redundant nodes and infection locations. For the ACO-GA, the path length is 35.147 m, which is shorter than that of ACO and GA but longer than the proposed approach. The quality and length of the path produced by the BCA have been remarkably improved; the path length is 32.147 m. However, its length is still longer than the path generated by PRMEGA. Table 3 also shows that the PRMEGA is the better algorithm in terms of required execution time. Therefore, compared with the traditional algorithms, the proposed algorithm has better performance in terms of path length and smoothness with low time requirements in the cluttered environment. Fig. 11 displays the outcomes of the simulation with the environment using the proposed approach.

Table 3 *The algorithms' results*

Algorithms	Planning path length (m)	Time consumption (sec)
PRMEGA	28.571	17.121
ACO	45.998	32.547
GA	37.545	96.148
ACO-GA	35.147	121.117
BCA-Q	33.663	-
BCA	32.147	293.442

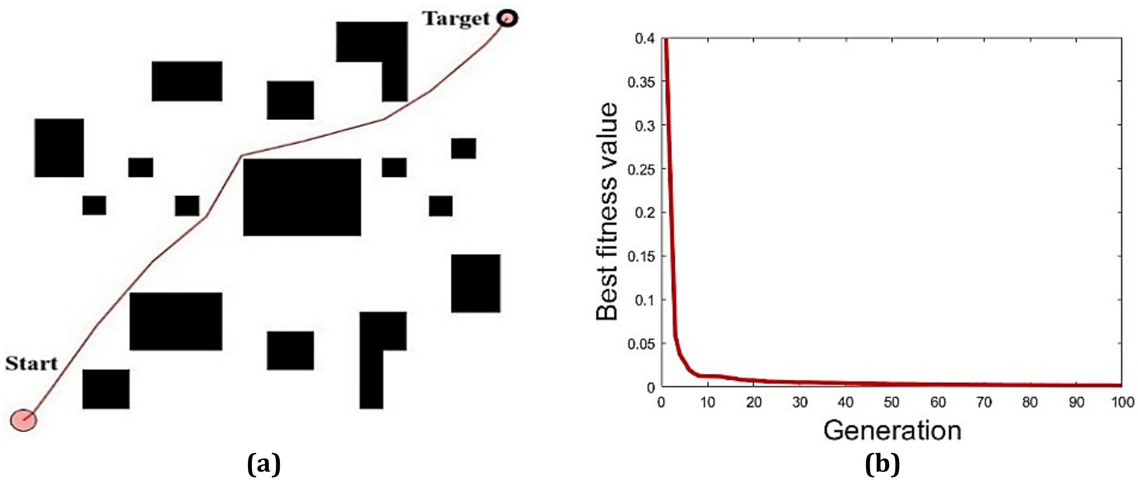


Fig. 11 *PRMEGA path planning on clutter map (a) Path planned by PRMEGA; (b) Fitness value*

Case 3:

In this case, the RPMEGA was compared with the fast-marching method (FMM), the fast-marching method hybridised with regression search (FMMHRS) methodology, and the artificial potential field method (APF) combined with particle swarm optimization (PSO) with a three-point smoothing method, which were developed in [19] and [27], along with their results.

The proposed approach was executed 20 times, and the average result has been calculated and is dependent on a simple static environment with a workspace of [500 × 500] cm. As shown in table 4, the proposed approach produced a path of length 664.16 cm with a time execution of 19.211 s. While the APF+PSO algorithm produced a path with a length of 819.87 cm, APF combined with PSO with a three-point smoothing method produced a path length of 753.26 cm.

Additionally, the FMMHRS and FMM algorithms produced paths with lengths of 664.26 cm and 676.08 cm, because they aim to create a straight line between the target point and intermediate points. In order to enable robot navigation without collisions, these algorithms then attempt to locate barriers and grant specific permissions around them. They so created a longer path and took a lot of time to get from the starting point to the target.

Finally, Table 4 demonstrates that the PRMEGA algorithm is better in terms of path length, path smoothness, and execution time, according to the results. Fig. 12 displays the outcomes of the simulation with the environment using the proposed approach.

Table 4 *The algorithms' results*

Algorithms	Planning path length (cm)	Time consumption (sec)
PRMEGA	664.16	19.211
FMM	676.08	30
FMMHRS	664.26	28
APF+PSO	819.87	46
APF+PSO+3-point	753.26	40

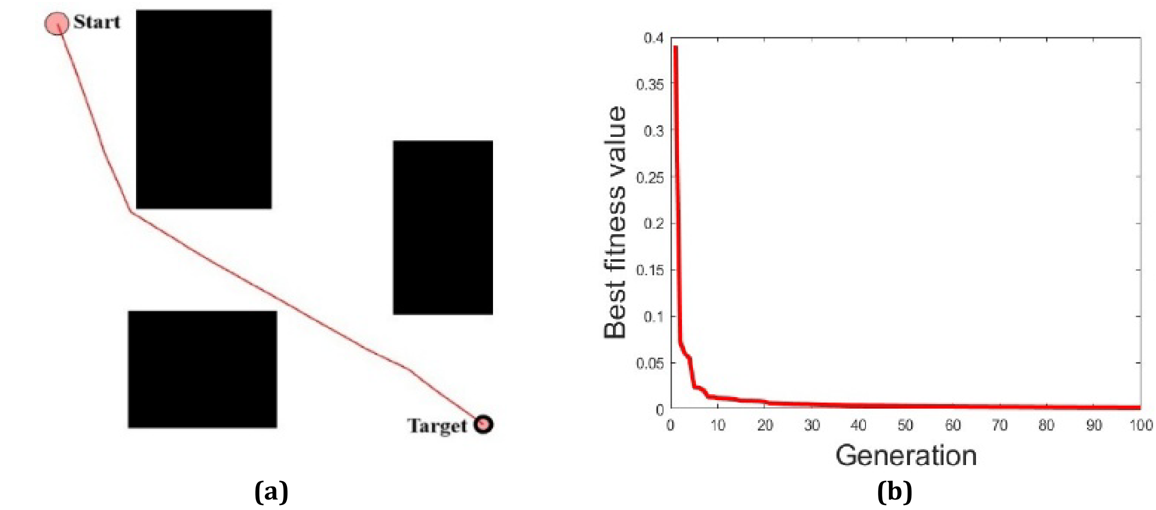


Fig. 12 *PRMEGA path planning on simple map (a) Path planned by PRMEGA; (b) Fitness value*

5.2 Results of Experiments of the Robot's Path Planning Have Multiple Targets

In this section, the proposed approach (i.e., PRMEGA) was applied on complex corridor maps for planning optimal or near-optimal paths for multiple independent targets; the values of the parameters are the same as indicated in Section 5.1.

Case 1:

In the first case, the proposed approach was utilised to plan a path from a single start point to two independent targets in a complex corridor map with a workspace of [50 × 50] m. Figure 13(a) depicts the optimal or near-optimal path produced by the proposed approach from the start point to the two targets. Figure 13(b) shows the number of iterations.

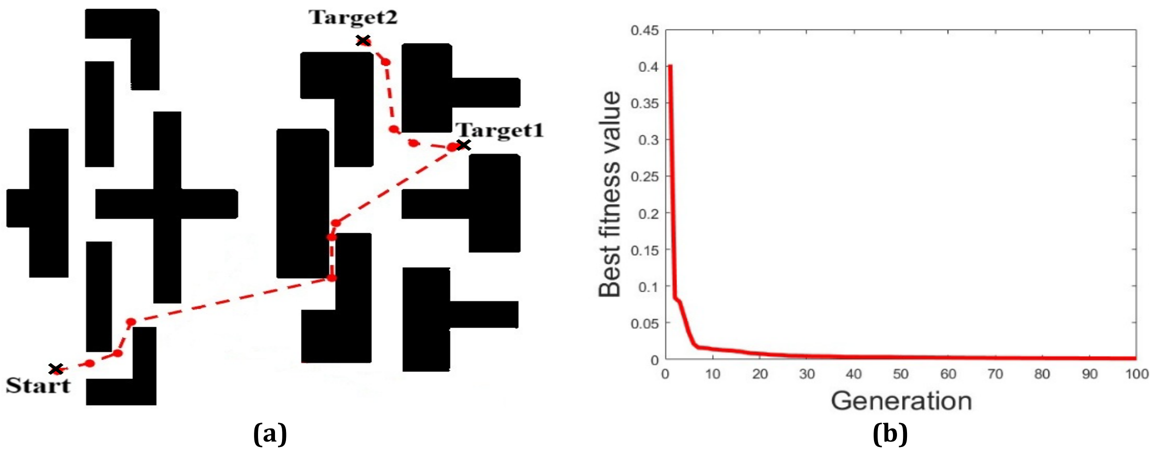
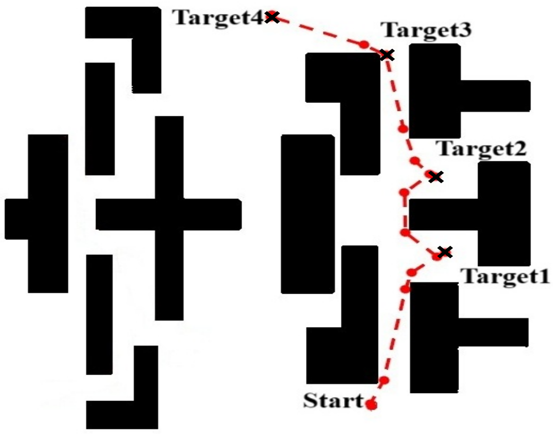


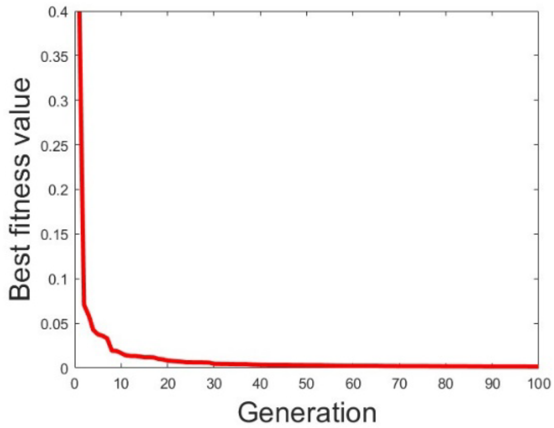
Fig. 13 *PRMEGA path planning on complex corridor map (a) Path planning by PRMEGA for two targets; (b) Number of iterations*

Case 2:

In the second case, the PRMEGA was utilised to plan a path from a single start point to four independent targets in the same complex corridor environment. Figure 14(a) depicts the optimal or near-optimal path produced by the proposed approach from the start point to the four targets. Figure 14(b) shows the number of iterations.



(a)



(b)

Fig. 14 *PRMEGA path planning on complex corridor map (a) Path planning by PRMEGA for four targets; (b) Number of iterations*

As shown in Figure 13(a), the proposed approach could plan an optimal or near-optimal path from the start point to the first target and then to the second target with a length of 105.128 m and a time execution of 59.985 sec. Also, the proposed PRMEGA planned an optimal or near-optimal path with a length of 78.985 m and time execution of 108.785 sec, from the start point to all four targets in Figure 14(a). The proposed approach first calculated the optimal sequence of targets, and then it planned the paths.

Experimental results demonstrate the effectiveness of the proposed approach in planning optimal or near-optimal paths for a single or multiple independent targets in a short execution time. In addition, the results depict the scalability of PRMEGA in terms of the size of the environments and their complexity (simple, complex, cluttered, corridor, and realistic environments). Nevertheless, when there are many targets, PRM needs more time to plan paths between targets and provide them for GA in order to determine the best path from the starting point to all targets.

6. Conclusion

This study proposed a hybrid PRMEGA algorithm for planning a collision-free, near-optimal smooth path from a starting point to a single target or multiple independent targets in various environments. The proposed approach utilised morphological operations and image processing for environment representation and thus eliminated the need to compromise between accuracy and speed when representing an environment. It used PRM for creating an initial population consisting entirely of feasible paths, which expedites the evolutionary process. An enhanced delete operator was developed to smooth the produced paths and increase their smoothness and make them more suitable in real-life applications. PRMEGA was compared with the PRM and traditional GA using a realistic map. Comparison results showed that the proposed approach planned the smoothest path in competitive time. Also, compared with the ACO, ACO-GA, ASFA-GA and BCA algorithms using a cluttered environment, PRMEGA enhanced path length, i.e., 37.9% compared with the ACO algorithm, 18.7% compared with the ACO-GA, 15.1% compared with the ASFA-GA and 11.1% compared with the BCA. Moreover, compared with the FMM, FMMHRS, PSO and APF+PSO+3-point algorithm using a simple environment, the PRMEGA enhanced path length by 1.76% compared with the FMM algorithm, 0.015% compared with the FMMHRS, 23.4% compared with the APS+PSO and 11.8% compared with the APF+PSO+3-point. The execution time of the proposed method was remarkably more competitive than those of all the other methods. Additionally, the proposed approach planned a feasible short path on a complex corridor map for multiple independent targets, i.e., two and four, in competitive execution time. The results clearly showed that the proposed hybrid algorithm produced the shortest collision-free smooth path without consuming a long time. In future work, the proposed approach can be adjusted to work with dynamic targets and in dynamic environments.

Acknowledgement

The authors would like to thank Ural Federal University for its support. The authors used QuillBot AI to assist in grammar checking, language editing. All content generated was reviewed and verified by the authors, who take full responsibility for the final submission.

Conflict of Interest

The authors declare that they have no conflict of interest regarding the publication of this paper.

Author Contribution

All the authors have designed the research, developed the methodology, performed the analysis, and written the manuscript. All authors have contributed equally.

References

[1] Geeta, S., Sanjeev, J., & Radhe, Sh. (2025). Path Planning for Fully Autonomous UAVs-A Taxonomic Review and Future Perspectives. *IEEE Access*, 13. DOI: 10.1109/ACCESS.2025.3529754.

[2] Saeid, N., Roohallah, A., Darius, N., Shady, M., Navid M., Mohammad, R., & Ibrahim, H. (2025). A comprehensive review on autonomous navigation. *ACM Computing Surveys*, 57(9), 1–67. https://doi.org/10.1145/3727642.

[3] Chen, W., & Jian, M. (2019). Summary of agv path planning. *2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE)*, 332–335. DOI: 10.1109/EITCE47263.2019.9094825.

[4] Ammar, A., Ahmed, AL., & Mohammed, N. (2022). Static and Dynamic Path Planning Algorithms Design for a Wheeled Mobile Robot Based on a Hybrid Technique. *International Journal of Intelligent Engineering and Systems*, 15(4), 167–181. DOI: 10.22266/ijies2022.0831.16.

[5] Giuseppe, F., René, B.M., Fabio, S., & Jan, O. S. (2021). Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda. *European Journal of Operational Research*, 294(2), 405–426. http://doi: 10.1016/j.ejor.2021.01.019.

[6] Krishna, T., Prases, K., & Shubhajit, D. (2025). Review on path planning methods for mobile robot. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 239(14). DOI.org/10.1177/09544062251330083.

[7] Gaojian, C., Yuxi, Y., Qipei, X., Chaolong, S., Guohua, L. & Shaosong, L. (2025). Efficient Path Planning for Automated Valet Parking: Integrating Hybrid A* Search with Geometric Curves. *International Journal of Automotive Technology*, 26(1), 243–253. https://doi.org/10.1007/s12239-024-00144-1.

[8] Soubail, D. (2025). Innovative technique with enriched movement directions to plan the trajectory for an autonomous Mobile robot. *Advances in Artificial Intelligence for Autonomous Robotic Applications,* 108(1), 1–19. DOI: 10.1177/00368504251321714.

[9] Semonti, B., Sajal, Ch. B., & Sarker, S. M. (2025). Path planning approaches in multi-robot system: A review. *Engineering Reports*, 7, e13035. https://doi.org/10.1002/eng2.13035.

[10] Takanori, Sh., & Toshio, F. (1993). Intelligent motion planning by genetic algorithm with fuzzy critic. *In Proceedings of 8th IEEE International Symposium on Intelligent Control*, *IEEE*, 565–570. DOI: 10.1109/ISIC.1993.397635.

[11] Man-Tak, Sh., & Gary, B. (1993). Genetic Algorithms for the Development of Real-Time Multi-Heuristic Search Strategies. *In ICGA*, 565–572. https://www.researchgate.net/publication/201976303.

[12] Fei, L., Shan, L., & Xiaodong, X. (2014). Optimal robot path planning for multiple goals visiting based on tailored genetic algorithm. *International Journal of Computational Intelligence Systems,* 7(6), 1109–1122. https://doi.org/10.1080/18756891.2014.963978

[13] Tüze, K., Ivan, T., & Katsunori, Sh. (2012). Incremental evolution of fast moving and sensing simulated snake-like robot with multiobjective GP and strongly-typed crossover. *Memetic Computing*, 4(3), 183–200. ttps://doi.org/10.1007/s12293-012-0085-z.

[14] Hong, Q., Ke, X., & Takacs, A. (2013). An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots. *Neurocomputing*, 120, 509–517. https://doi.org/10.1016/j.neucom.2013.04.020

[15] Chaymaa, L., Said, B., & Ali, E. (2018). Genetic algorithm-based approach for autonomous mobile robot path planning. *Procedia Computer Science*, 127, 180–189. https://doi.org/10.1016/j.procs.2018.01.113.

[16] Ritam, S., Debaditya, B., & Nirmalya, Ch. (2020). Domain knowledge based genetic algorithms for mobile robot path planning having single and multiple targets. *Journal of King Saud University - Computer and Information Sciences*, 34(7), 4269–4283. https://doi.org/10.1016/j.jksuci.2020.10.010.

[17] Sethakarn, P., & Suchada, S. (2021). Differential drive analysis of spherical magnetic robot using multi-single board computer. *International Journal of Intelligent Engineering and Systems*, 14(4), 264–275. DOI: 10.22266/ijies2021.0831.24.

[18] Noor, KH., & Ahmed, Al. (2022). Intelligent Hybrid Path Planning Algorithms for Autonomous Mobile Robots. *International Journal of Intelligent Engineering and Systems,* 15(5), 309–325. DOI: 10.22266/ijies2022.1031.28.

[19] Zainab, K., Ahmed, Al., & Mohammed, A. (2022). Enhancement of Cell Decomposition Path-Planning Algorithm for Autonomous Mobile Robot Based on an Intelligent Hybrid Optimization Method. *International Journal of Intelligent Engineering and Systems*, 15(3), 161–175. DOI: 10.22266/ijies2022.0630.14.

[20] Shaymaa, A., Alexander, P., Hussam, L., Mohammed, M., & Amjad, H. (2025). Dynamic Processing 2D Maps Method for Robot' s Trajectory Planning. *Proceedings of Engineering and Technology Innovation*, 30, 79-89. https://doi.org/10.46604/peti.2024.14508

[21] Xiaolu, M., Rui, G., Yibo, T., Hong, M., & Chengcheng, L. (2022). Path planning of mobile robot based on improved PRM based on cubic spline. *Wireless Communications and Mobile Computing*, 2022(1), 1632698. https://doi.org/10.1155/2022/1632698.

[22] Shengjin, C., Guangyong, Y., Guanghai, C., Shang, Y., & Lihuang, W. (2025) .Improved Path Planning and Controller Design Based on PRM. *IEEE Access*, 13(3), 44156–44168. DOI: 10.1109/ACCESS.2025.3548326.

[23] Areej, Al., & Awos, K. (2023). Genetic Algorithm-Based Path Planning for Autonomous Mobile Robots. *In 2023 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, *IEEE*,177–180. DOI: 10.1109/JEEIT58638.2023.10185855.

[24] Adem, T., & Mehmet, Y. (2012). Dynamic path planning of mobile robots with improved genetic algorithm. *Computers & Electrical Engineering*, 38(6),1564-1572. https://doi.org/10.1016/j.compeleceng.2012.06.016

[25] Zhifeng, Y., & Ye, X. (2024). An improved genetic algorithm for robot path planning. *Journal of Computational Methods in Sciences and Engineering,* 24(3), 1331–1340. https://doi.org/10.3233/JCM-247133.

[26] Jianwei, M., Yang, L., Shaofei, Z., & Lin, W. (2020). Robot Path Planning Based on Genetic Algorithm Fused with Continuous Bezier Optimization. *Computational Intelligence and Neuroscience*, 2020(1), 18–27. DOI: 10.1155/2020/9813040.

[27] Ravi, M., Katla, M., and Pandu, V. (2019). Dynamic motion planning algorithm for a biped robot using fast marching method hybridized with regression search. *Acta Polytechnica Hungarica*, 16(1), 189–208. DOI: 10.12700/APH.16.1.2019.1.10.