Journal of Robotics and Control (JRC)
Volume 6, Issue 2, 2025
ISSN: 2715-5072, DOI: 10.18196/jrc.v6i2.25572

Improved Path Planning for Multi-Robot Systems Using a Hybrid Probabilistic Roadmap and Genetic Algorithm Approach

Thanushika Jathunga [1*], Samantha Rajapaksha [2]
[1] Faculty of Engineering, Sri Lanka Institute of Information Technology, Malabe, Sri Lanka
[2] Faculty of Computing, Sri Lanka Institute of Information Technology, Malabe, Sri Lanka
Email: [1] thanushika.j@sliit.lk, [2] samantha.r@sliit.lk
*Corresponding Author

***Abstract*—This study focuses on the development and application of an improved Probabilistic Roadmap (PRM) algorithm enhanced with Genetic Algorithms (GA) for multi-robot path planning in dynamic environments. Traditional PRM-based methods often struggle with optimizing path length and minimizing turns, particularly in complex, multi-agent scenarios. To address these limitations, we propose a hybrid PRM-GA approach that incorporates genetic operators to evolve optimal paths for multiple robots in real-time.The research contribution is an enhanced PRM-GA framework that improves efficiency in multi-robot navigation by integrating evolutionary techniques for dynamic obstacle handling and optimized path generation.The research methodology involves testing the algorithm in various environments, including varying robot numbers and environmental complexities, to evaluate its scalability and effectiveness. Our results demonstrate that the PRM-GA algorithm successfully reduces both path lengths and turn counts compared to standard PRM-based methods, ensuring collision-free and smooth paths. The algorithm showed robust performance across different scenarios, effectively handling dynamic obstacles and multi-agent coordination. However, in highly dynamic environments with rapidly changing obstacles and constraints, the algorithm may occasionally produce paths with turn counts and distances similar to or slightly higher than those of simpler approaches due to the need for frequent re-optimization. Future research can explore incorporating additional factors such as energy consumption and time optimization, alongside distance and turns, to further enhance the algorithm's efficiency in real-world applications. Overall, the PRM-GA approach advances the state of the art by offering a more adaptable and scalable solution for multi-robot path planning, with applications in logistics, industrial automation, and autonomous robotics.**

Keywords*—Probabilistic Roadmap (PRM); Genetic Algorithms (GA); PRM-GA Hybrid Method; Multi-Robot Path Planning.

I. INTRODUCTION

Efficient path planning is a fundamental problem in robotics, particularly in environments with multiple robots operating simultaneously. Multi-robot systems are increasingly being used in applications such as automated warehouses, search-and-rescue operations, and autonomous vehicle coordination. In such scenarios, generating optimal paths for robots to navigate from start to goal positions without collisions while minimizing travel cost and complexity is critical [1].

Traditional methods for robot path planning, such as Dijkstra's and A* algorithms, often struggle with scalability and computational overhead in multi-robot environments. Probabilistic Roadmap (PRM) is a widely used sampling-based approach for motion planning due to its ability to efficiently handle high-dimensional spaces [2][3]. However, PRM-generated paths may not always be optimal, especially in terms of path smoothness and minimal distance. Moreover, PRM does not inherently address multi-robot coordination, making collision avoidance challenging in dynamic environments.

On the other hand, Genetic Algorithms (GAs) are powerful optimization tools inspired by natural selection and genetic evolution [4]. GA excels in refining solutions by iteratively improving a population of candidate paths using genetic operations such as selection, crossover, and mutation. Despite the optimization capabilities, GA is computationally intensive and requires effective initialization to avoid convergence to suboptimal solutions. Furthermore, GAs alone do not explicitly incorporate real-time adaptability, which is crucial for multi-robot coordination in dynamic environments.

To address these challenges, this paper proposes a hybrid approach that leverages the strengths of both PRM and GA while mitigating their limitations**.** The PRM is employed to generate feasible initial paths, which serve as the initial population for the GA. The GA then optimizes these paths by considering two critical fitness objectives: minimizing total path distance and reducing the number of sharp turns. By integrating PRM's efficient path generation with GA's optimization capability, this hybrid approach enhances both computational efficiency and path quality. More importantly, it provides an adaptive framework suitable for real-time multi-robot coordination, reducing potential collisions and improving navigation in complex environments.

The proposed approach is particularly relevant in real-world applications such as autonomous warehouse logistics, where multiple robots must navigate efficiently in confined spaces, or in search-and-rescue missions where robots need to coordinate movement in uncertain terrain. By ensuring smooth and computationally feasible path planning, the hybrid PRM-GA method enhances practical deployment in such scenarios.

The key research contributions of this study are as follows. This paper presents a novel hybrid PRM-GA approach that optimizes path planning for multi-robot systems by leveraging PRM's efficiency in path generation and GA's adaptability in path optimization. The study develops a framework that enhances real-time adaptability and collision avoidance in dynamic multi-robot environments. Furthermore, experimental validation is conducted to demonstrate the computational benefits and effectiveness of the hybrid approach in reducing travel cost and improving path smoothness.

The rest of this paper is organized as follows. Section II presents a comprehensive overview of background research. Section III details the proposed methodology, including the implementation of PRM and GA and how they are adapted for multi-robot systems. Section IV presents experimental results and analysis, highlighting the advantages of the proposed method. Finally, Section V concludes with a discussion of findings and potential future work.

II. Background Research

Path planning is essential for autonomous mobile robots to efficiently navigate in various environments such as industrial, agricultural, healthcare, and service sectors. The goal is to find an optimal or near-optimal path from a start point to a target point, avoiding obstacles, minimizing travel distance, energy consumption, and ensuring real-time feasibility. Literature discusses a range of path planning techniques developed in recent years, including Genetic Algorithms (GA), Probabilistic Roadmaps (PRM), Particle Swarm Optimization (PSO), and hybrid approaches.

Several recent review papers provide an overview of existing path planning techniques, highlighting emerging trends. These reviews also discuss key performance metrics such as path smoothness, real-time applicability, and energy efficiency. Liu et al. [5] classify path planning approaches into global and local methods based on environmental knowledge. Their review categorizes algorithms into classical, bionic, and artificial intelligence-based techniques, offering a comprehensive overview of path evaluation strategies and emerging trends. Similarly, another study [6] emphasizes learning-based approaches for robotic manipulators, highlighting the adaptability of reinforcement learning and deep learning methods in complex, high-dimensional spaces. The challenges of path planning in dynamic and multi-objective scenarios are discussed in [7] emphasizing the need for robust autonomous navigation strategies. In [8], authors discuss obstacle avoidance algorithms, demonstrating the potential of hybrid meta heuristic approaches like PSO-GA and fuzzy-neural models in overcoming limitations of classical techniques. A comparative analysis of classical and heuristic algorithms are presented in [9] focusing on dynamic clutter navigation and moving target tracking. Another study [10] addresses Coverage Path Planning (CPP) issues, emphasizing the importance of minimizing travel time, energy consumption, and overlapping. The integration of optimization algorithms with classical methods is proposed as a solution to the challenges posed by limited battery capacities and low planning efficiency. Further, in [11] authors examine the evolution of path planning algorithms, discussing the potential of cloud computing technologies to enhance algorithm deployment and scalability. A performance analysis of various path planning algorithms are presented in [12] with simulation results indicating the superiority of the Rapidly Exploring Random Tree (RRT) algorithm in deterministic environments. Mustafa et. al [13] review optimization methods used in path planning, offering a concise overview of recent advancements. In [14], the study provides insights into motion planning algorithms, with an emphasis on machine learning and reinforcement learning methods, which show promise for improving convergence speed and stability.

Particle Swarm Optimization (PSO) is a population-based optimization algorithm that is often combined with other methods to improve real-time adaptation in dynamic environments. PSO has been shown to perform well in both static and dynamic scenarios. In the literature, researchers have introduced a hybrid Particle Swarm Optimization (PSO) and Simulated Annealing (SA) algorithm for Autonomous Guided Vehicle (AGV) path planning. This algorithm aims to enhance scalability and performance by minimizing local optima and reducing computation time. The hybrid PSO-SA algorithm outperforms other heuristic algorithms, showing faster convergence and significantly better performance in path length minimization and smoothness. Experimental results demonstrate its effectiveness in optimizing AGV path planning tasks [15]. A hybrid approach combining Improved Particle Swarm Optimization (IPSO) and the Improved Dynamic Window Approach (IDWA) is proposed in [16] for mobile robot path planning in dynamic environments. IPSO enhances search accuracy and exploration capabilities, while IDWA improves dynamic obstacle avoidance by integrating the velocity obstacle (VO) concept. The combination of these methods helps the robot effectively navigate through dynamic obstacles while maintaining path smoothness and safety. The APSO (A* and PSO) hybrid algorithm, which integrates A* for initial path planning and PSO for optimization, enhances search ability and reduces runtime by employing redundant point removal and stochastic inertia weight adjustments. The proposed method improves path efficiency, with simulation results showing a reduction of up to 18.97 percent in running time compared to traditional A* algorithms [17]. Faiza et al. present a multi-objective path planning algorithm using a hybrid Grey Wolf Optimization (GWO) and PSO approach. The algorithm addresses the NP hard nature of path planning by optimizing path distance, smoothness, and collision avoidance in three distinct steps. The approach incorporates evolutionary mutation operators to further improve path safety and reduce path length. Simulations demonstrate that this method outperforms conventional techniques, providing feasible, collision-free paths [18]. In another study, authors explore a bio-inspired hybrid algorithm that combines GWO and PSO with a mutation operator for multi-constraint path planning and collision avoidance. This algorithm accelerates convergence through frequency-based modifications and integrates a more efficient collision avoidance strategy, transforming non-feasible points into optimal solutions. The method shows superior performance in path smoothness and safety across various simulation environments [19]. A hybrid FA and

Modified Chaotic PSO (MCPSO) algorithm, called HFAMCPSO is proposes in [20]. This hybrid algorithm significantly outperforms others like Radial Cell De composition (RCD) and A* in terms of path length, reducing path length by 43.3 percent and 25.5 percent, respectively. The hybrid approach also improves efficiency and path smoothness, demonstrating its superiority in navigating static obstacles. Another study addresses global path planning under kinematic constraints using a modified PSO (MPSO) algorithm combined with η3-splines for path smoothing. The MPSO algorithm incorporates adaptive random fluctuations to overcome local convergence issues and improve optimization. The integration of η3-splines ensures smoother trajectories while respecting kinematic limitations. Simulation results show that this method outperforms other PSO-based algorithms in path quality and smoothness [21]. With the integration of subsystems such as motion planning, localization, and map generation, this study offers a PSO-based autonomous robot navigation system for unknown environments that guarantees effective path-following and collision avoidance. The PSO algorithm minimizes the travel distance between several waypoints to improve the robot's route. Through successful testing with the Gazebo simulator in a variety of dynamic situations, the system showed strong navigation and obstacle avoidance capabilities [22]. Another study introduces the Bi-Population Particle Swarm Optimization with a Random Perturbation Strategy (BPPSO) for mobile robot path planning. BPPSO divides particles into two subpopulations to enhance global and local search capabilities. The random perturbation strategy boosts diversity, preventing premature convergence and improving search quality. Experimental results show that BPPSO outperforms conventional PSO algorithms in terms of path quality and runtime efficiency [23].

The A* algorithm is another foundational path-planning method that employs a heuristic-based approach to find the shortest path between two points. While simple and efficient, its high memory consumption limits its scalability. To address this, one of the studies have proposed an improved A* algorithm, introducing bidirectional search, a guideline for path optimization, and a key point list to reduce memory usage. MATLAB simulations demonstrated a 60 percent reduction in memory footprint, highlighting the algorithm's practicality in constrained environments [24]. Building on A*, [25] introduced the Adaptive Bidirectional A* (ABA*) algorithm for aerial robot path planning in 3D environments. By integrating adaptive techniques for 3D variables and employing bidirectional search, ABA* achieved a 91.2 percent reduction in planning time compared to traditional A*, showcasing its potential for computationally limited 3D applications. Similarly, another study has presented a hybrid RRT-A*-BT approach that combines Rapidly Exploring Random Trees (RRT) with A* and Back-Tracking (BT). This method, supported by vision-based environment modeling, demonstrated superior performance in cluttered indoor settings, outperforming Genetic Algorithms and standard RRT in path length and computational efficiency [26]. To address the challenges of dynamic obstacles and real time planning, [27] proposed a fusion algorithm combining an improved A* for global planning and the Dynamic Window Approach (DWA) for local planning. Enhancements to A* included sub-node selection and path smoothness, while fuzzy control was integrated into DWA. Experiments using a TurtleBot3 robot showed a 9.6 percent reduction in path length and a 29 percent reduction in planning time, emphasizing the algorithm's adaptability and efficiency in dynamic environments. Focusing on two-wheel mobile robots, another study has integrated RRT, BiRRT, and HA* for path planning and trajectory tracking. The study used a kinematic Differential Drive Mobile Robot (DDMR) model with a PID controller to minimize tracking deviations. Optimization techniques such as Particle Swarm Optimization (PSO) and Hybrid Butterfly Particle Optimization (HBPO) were applied to fine-tune PID parameters. Simulation results showed that HA* generated smoother and shorter paths compared to RRT and BiRRT, while HBPO outperformed other methods by achieving faster convergence [28].

Ant Colony Optimization (ACO) has emerged as a powerful algorithm for solving path-planning problems, inspired by the natural behavior of ants in finding the shortest paths to food. ACO-based techniques are particularly effective in complex environments due to their ability to balance exploration and exploitation. However, traditional ACO faces challenges such as slow convergence, redundant paths, and inefficiency in high-dimensional spaces. To address these limitations, researchers have proposed several enhancements and hybrid approaches. Basic ACO often struggles with inefficiency, particularly during early iterations, where ants explore redundant or suboptimal paths. Wang et al. proposed a Monte Carlo based Improved ACO (MC-IACO) for the path planning of welding robots. By integrating Monte Carlo sampling, this method evaluates path quality during each iteration using a sigmoid-based Monte Carlo factor, enabling ants to prioritize nodes likely to form optimal paths. A feedback adjustment factor further enhances decision-making by considering the impact of subsequent nodes. Simulations demonstrated that MC-IACO significantly outperforms basic ACO in terms of pathfinding efficiency, making it suitable for applications like welding robot path planning [29]. In [30], authors have proposed an improved ACO algorithm for intelligent warehouse robot path planning, modeling storage shelves with Poisson Distribution for unknown factors and using a three-color raster map. An optimized pheromone mechanism evaluates paths based on safety, length, and turns. Simulations showed the algorithm required fewer iterations, turns, and runtime, effectively addressing blind search and deadlock issues, high lighting its efficiency for warehouse logistics. To overcome the limitations of ACO and enhance its convergence speed and solution quality, hybrid approaches combining ACO with Genetic Algorithms (GA) have been explored. Kangkang Ma et al. introduced a fusion algorithm that integrates ACO and GA through two key strategies: the Optimal Strategy, which selects high-quality parent paths using a roulette mechanism, and the Genetic Region Strategy, which restricts offspring ants to search within defined regions. This method reduces the search area, accelerates convergence, and stabilizes performance. Simulation results showed that the fusion algorithm achieves faster and more reliable convergence than standalone ACO or other variations [31]. Tsagaris and Mansour have further extended the application of ACO-GA hybrids for mechatronic systems, such as robotic arms and CNC machines, requiring optimization of large point sets. The hybrid approach resulted in up to a 40 percent reduction in trajectory length and a 20 percent decrease in path planning time, significantly outperforming traditional GA methods. Although real-world applications slightly reduced these improvements, the method demonstrated its efficacy for high-dimensional path planning tasks [32].

Genetic Algorithms (GAs) are one of the most popular optimization methods used in the domain of path planning due to their ability to handle complex, multidimensional solution spaces. They evolve a population of candidate solutions iteratively to find the optimal or near-optimal path, often balancing efficiency with the ability to navigate through environments with numerous obstacles. Multiple studies have contributed enhancements to traditional GAs, incorporating problem-specific knowledge and introducing new hybrid methods. In [33], authors have proposed a hybrid Genetic Algorithm-Bezier Curve (GA-BZ) method for mobile robot path planning, optimizing performance in environments with varying obstacle densities. Simulation results highlighted the superior efficiency of GA-BZ compared to other algorithms like A-Star and Dijkstra's. Similarly, Mohanraj et al. [34] developed a hybrid algorithm to balance global path planning and real-time navigation, reducing computation time and enhancing path cost efficiency, particularly for larger grids. Multi-objective optimization methods have also been explored. A Multi Objective Genetic Algorithm (MRPS-MOGA) that prioritizes safety, smoothness, and trip duration is introduced in [35]. The algorithm outperformed conventional methods in both time complexity and path efficiency. In a similar way, another study has addressed the limitations of the Firefly Algorithm by hybridizing it with Genetic Algorithm operations, mitigating the issue of local optima and improving global path planning accuracy [36]. Fang and Liang have presented an intelligent obstacle avoidance system combining a potential field method with Genetic Algorithm and Reinforcement Learning to improve manipulator efficiency. This system significantly reduced path length, nodes, and working time in simulation tests [37]. A simulated annealing-based genetic algorithm for evacuation route planning is proposed in [38]. The study proved effective in avoiding local optima while maintaining population diversity through adaptive genetic operators. In [39], authors have proposed a modified Genetic Algorithm designed for industrial robots to address path planning in confined workspaces. The system successfully handled arc welding-specific technology limitations and collision avoidance. Similar to this, Sarkar et al. [40] have improved the performance of traditional genetic algorithms by lowering path complexity with the introduction of domain knowledge-based operators for single and multi-target path design. Hybrid fuzzy-logic-based GA approaches have also demonstrated well performance in selecting collision free paths. In [41] authors have used a combined fuzzy computing with a genetic algorithm to evaluate and select collision-free paths, relying on fuzzy logic when paths were blocked. Rath et al. [42] extended this by integrating Genetic Algorithms with Neural Networks for humanoid robots, achieving minimal navigation errors in cluttered environments. In another study, Rath et al. [43] proposed a hybrid fuzzy-genetic algorithm for path planning, combining fuzzy logic for initial calculations and GA for optimization. In the realm of three-dimensional path planning, one of the studies [44] proposed a Hybrid Genetic Cuckoo Search Algorithm, which combined the global search capability of GA with the adaptive local search ability of Cuckoo Search. This approach effectively handled multi constrained environments and improved efficiency. Similarly, Sriniketh et al. [45] proposed a Genetic Algorithm (GA) based approach to optimize evacuation routes during fire drills, significantly reducing path distances compared to conventional algorithms.

Probabilistic Road Maps (PRMs) are another commonly used global path planning method, particularly in environments with many obstacles. PRM constructs a graph of randomly sampled points, connecting them using simple local planners to form a roadmap that the robot can follow. Often combined with other optimization techniques like GAs or PSO, PRMs provide a powerful tool for handling complex and high-dimensional environments. Pohan and Utama introduced Smart-PRM, a fast and asymptotically optimal path planning algorithm, to overcome the limitations of traditional PRM. Smart-PRM incorporates five smart sampling strategies, including informed search, incremental dense sampling, and sampling near obstacles, to enhance efficiency and path quality. Comparative studies with PRM and other advanced algorithms like Informed RRT*-connect highlighted Smart PRM's superior speed and accuracy. Its ability to consistently generate optimal paths quickly makes it suitable for real-world applications requiring high computational performance [46]. Similarly, another study has proposed a hybrid method that enhances population initialization and reduces infeasible paths by incorporating fitness-based initialization techniques. This method also introduces a combination of genetic operators to optimize path quality. Experimental results demonstrated the effectiveness of the CBPRM-GA approach in reducing computational time and improving path feasibility and quality, showcasing its potential for solving the MRGPP problem [47]. Another promising method integrates Probabilistic Roadmaps (PRM) with Modified Ant Colony Optimization (ACO) for path planning in complex environments. Raheem proposed a three-stage framework that first generates a random roadmap using PRM, followed by path optimization using modified ACO, and concludes with path smoothing using third-order B-spline curves. This approach addresses the need for smooth, continuous, and efficient paths, ensuring feasibility and safety in dynamic settings. The method demonstrated significant improvements in path quality and smoothness compared to traditional techniques [48].

Over the years, numerous other methods and algorithms have also been developed to address challenges such as obstacle avoidance, efficiency, and adaptability to complex scenarios. These include classical algorithms, optimization techniques, hybrid methods, and AI-driven approaches. Li et al. [49] leveraged Deep Reinforcement Learning (DRL) with reflective reward design to enhance multi-agent and multi-task learning, enabling higher success rates in dynamic environments. Building on the limitations of traditional methods, another study has combined the Whale

Optimization Algorithm (WOA) and Adaptive Genetic Algorithm (AGA) into the WOA-AGA model, achieving improved efficiency and shorter paths in complex scenarios [50]. Addressing dynamic environments, researchers have enhanced the Simulated Annealing (SA) algorithm with improved path selection and deletion operations, demonstrating superior adaptability [51]. Meanwhile, another research has introduced the Pelican Optimization Algorithm (POA) and compared it to Particle Swarm Optimization (PSO), achieving lower trajectory errors for obstacle avoidance [52]. To further refine navigation strategies, another study has utilized the Gray Wolf Optimizer (GWO), which outperformed PSO, ABC, and other heuristic methods by maintaining an effective balance between exploration and exploitation [53]. In another hybrid approach, Awei et al. [54] combined the Mayfly Optimization Algorithm with the Dynamic Window Approach, enhancing real-time obstacle avoidance and reducing path lengths. The Quantum-inspired Evolutionary Algorithm (QEA) proposed by Gao et al. [55] significantly improves runtime efficiency and accuracy in large-scale optimization problems. On the other hand, Elmi and Efe [56] developed a sensory-driven grasshopper algorithm for crowded and dynamic environments, ensuring collision-free paths with superior stability. Another study [57] introduced the Adaptive Parallel Arithmetic Optimization Algorithm (APAOA), incorporating novel parallel communication strategies to prevent local optima and enhance convergence. Meanwhile, Kumar and Sikander [58] improved the Artificial Bee Colony Algorithm by integrating B. Units Evolutionary Programming, achieving notable reductions in path length and search costs. In the realm of hybrid optimization, Zang has combined WOA with computer perception techniques, enabling robots to perceive and optimize paths in complex environments effectively [59]. Further exploration of swarm intelligence was undertaken by Seyedadhi and Chitra [60], who applied the Cuckoo Optimization Algorithm for dynamic obstacle navigation, ensuring smooth and collision free paths. Hybridization was also explored in [61], where the authors have merged the Cuckoo Search and Bat Algorithm (BA), demonstrating robust performance in unknown environments. In [62], Deep learning approaches were integrated and researchers have employed a GRU-RNN model with modified ant colony optimization techniques to enhance collision avoidance in dynamic scenarios. Another study [63] developed hybrid swarm intelligence algorithms combining Cuckoo Search with Bat Algorithm and Firefly Algorithm, achieving superior efficiency and success rates compared to standalone methods.

Some research works were related to robot path planning using machine learning algorithms. Many research works used supervised, unsupervised, and reinforcement learning. However, most of the implementations and evaluations were at the theoretical level instead of real-world applications [64]. Neural network and reinforcement learning are the main techniques used in machine learning for local path planning [65]. Deep reinforcement learning was used on path planning for dynamic environments with transfer learning [66][79]. There were enhanced reinforcement learning algorithms for autonomous path planning with fewer steps [67][68]. End-to-end and Imitation Learning were used to get more optimal performance in autonomous driving [69]. Machine learning and deep learning were used in Multi-Robot Path Planning to get the best solutions [70][71][72]. The path planning can be improved using a hybrid approach where two or more methods can be merged to get optimal results [73][74]. There were studies on web-based robot path planning with user commands [75][76][77]. Some research was done on computer vision and image processing to navigate robots [78]. Many research works used heuristics and metaheuristics to find optimal paths in unknown areas [80][81]. Artificial potential field and reinforcement learning were used in robot pathfinding with optimum results [82][83][86]. Static and dynamic rewarding techniques were used to find the optimal path in an unknown environment [84][85]. Posture rewarding and sparse rewarding techniques were used with reinforcement learning to find the best path in an unknown area for robots [86][87]. Energy efficient path planning was developed using deep meta learning algorithms to get the optimal results [88]. Potential fields with population-based meta-heuristics is a technique used in real time path planning for robots [89].

Together, these studies highlight the diverse and innovative solutions developed for addressing the challenges of robot path planning, ensuring safer, more efficient, and adaptive navigation in complex environments. Based on the reviewed literature, the choice of combining the Probabilistic Roadmap (PRM) with Genetic Algorithm (GA) for multi-robot path planning is well-supported by several key factors. PRM efficiently handles high-dimensional environments by generating collision-free paths, while GA optimizes these paths for criteria such as energy efficiency, smoothness, and collision avoidance. Literature shows that PRM alone can create feasible paths, but struggles with optimizing them for multiple objectives, especially in dynamic environments. GA addresses this by refining initial PRM generated paths, ensuring optimal routes for multiple robots in complex, real-time scenarios. The research gap lies in the limited exploration of hybrid methods like PRM-GA specifically for multi-robot systems. While both techniques are well-studied individually, their combined potential to enhance path planning in dynamic, real-world settings require further investigation. Thus, selecting PRM-GA fills this gap by offering a scalable and adaptable solution for efficient, coordinated robot navigation.

III. METHODOLOGY

The methodology behind the Genetic Algorithm (GA) used in this multi-robot path finding approach follows a structured, phase-based approach aimed at achieving efficient and collision-free navigation in a grid environment with fixed obstacles. Fig. 1 illustrates the process of the proposed path planning algorithm.

The process begins with environmental modeling, where the grid and obstacles are mapped, laying the foundation for planning. Following this, the initial population of potential paths for each robot is generated using the Probabilistic Roadmap (PRM) method, which samples 1,000 random points within the workspace and connects them based on predefined distance criteria. This generates a diverse set of paths, representing various possible routes from each robot's starting point to its target destination. Next, in the genetic operations phase, each path is evaluated using a fitness function. This function is designed to minimize the path length and the number of turns. The selection process ensures that the best-performing paths (individuals) are given higher chances to reproduce and pass on their traits to the next generation. Elitist selection is used, where the top-performing paths are directly carried over to the next generation to maintain solution quality. Crossover operation combines sections from two well-performing paths to create new offspring paths, increasing diversity in the population and enhancing overall fitness. After crossover, mutation introduces minor changes, such as slight deviations in cell selection, helping the algorithm avoid local optima and improving path exploration. The mutation process accounts for collision avoidance by ensuring minimum separation between robots. GA iterates through these genetic operations until it either converges on an optimal solution or reaches a predefined stopping criterion, such as a maximum number of iterations or a minimum error threshold. This iterative process refines the paths, balancing efficiency and safety. By the end of the process, the GA produces an optimized set of collision-free paths, enabling effective multi-robot navigation within the fixed-obstacle environment.

A. Environmental Modelling

The environmental model for this simulation utilized a binary occupancy grid with a 10x10 unit grid structure, representing the operational area for the robots. In this setup, each cell represents a specific spatial location within the grid, allowing for a straightforward distinction between navigable and obstructed spaces. Obstacles were strategically positioned at pre-determined coordinates within the grid, and these locations were marked with a binary value of 1 to indicate non-traversable areas. The rest of the cells were assigned a value of 0, representing open and accessible paths for the robots.

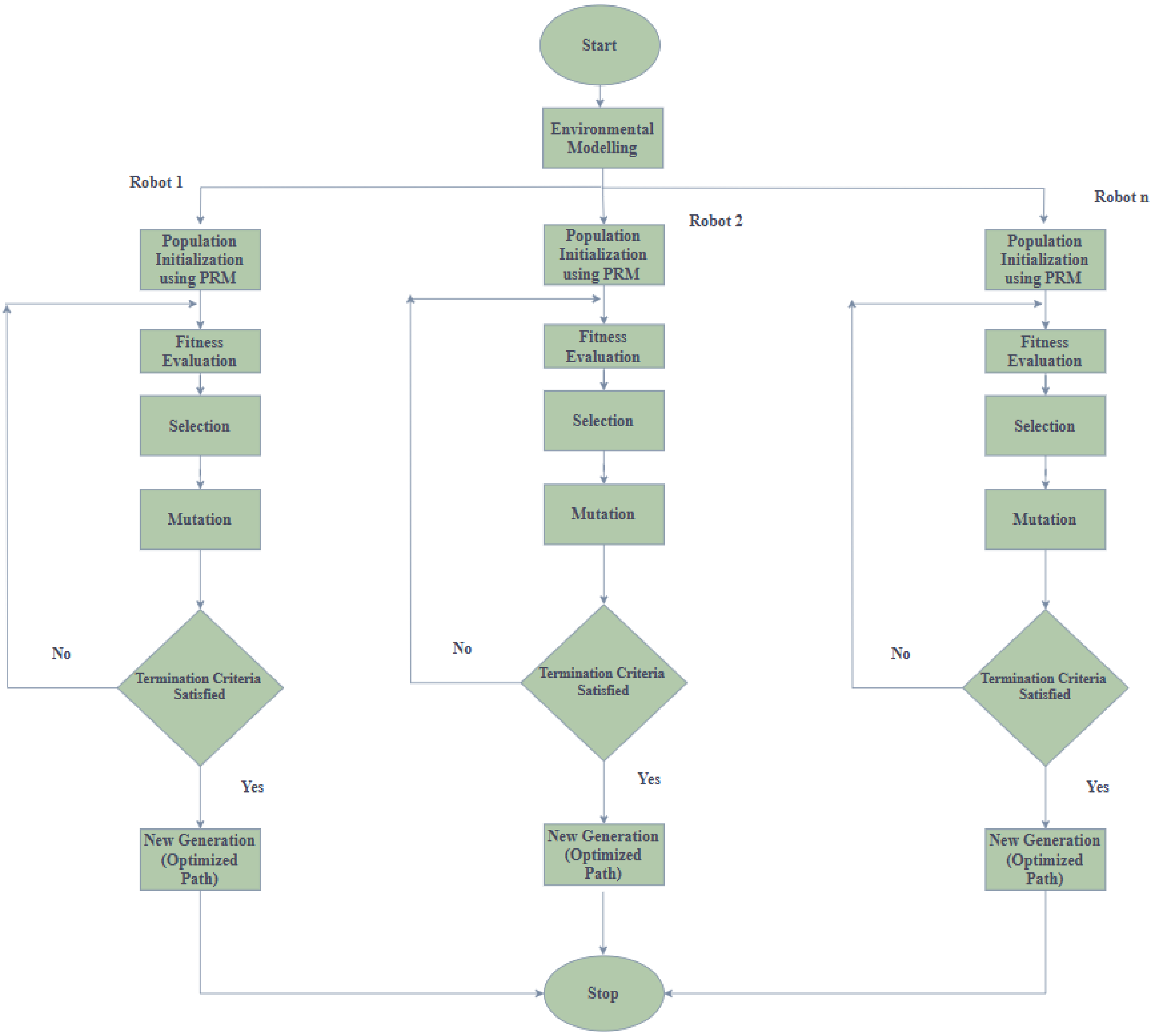

Fig. 1. Process of the proposed PRM-GA algorithm

To effectively assign these obstacle locations within the grid, their coordinates were transformed into linear indices, accurately marking the designated cells as occupied. This conversion allowed for efficient mapping of each obstacle within the structured environment, enhancing the simulation's spatial awareness by clearly defining boundaries and barriers. This modeling approach emphasized both the spatial distribution and density of obstacles, providing the robots with a detailed, realistic environment for pathfinding and collision avoidance, and forming a foundation for optimizing their movements within the structured grid. Fig. 2 illustrates the simulation environment, highlighting the static obstacles within the workspace.

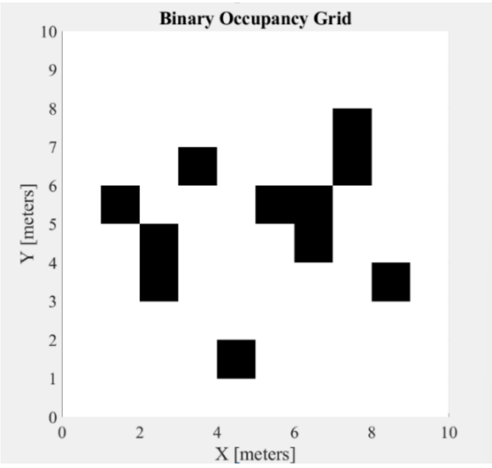

Fig. 2. Simulation environment

B. Probabilistic Road Map Initialization

The initial path generation employed the Probabilistic Roadmap (PRM) method. A total of 1,000 random points were sampled within the workspace, and connections were established between points that fell within a predefined connection distance of 1 unit. Each connection was validated to ensure it did not intersect with obstacles, thereby guaranteeing collision-free paths. Once the roadmap was constructed, each robot was assigned start and goal positions, and the PRM computed an initial feasible path. Although these paths avoided obstacles, they did not always optimize path length or minimize the number of turns.

C. Fitness Function

In the genetic algorithm's selection phase, each candidate path is evaluated using a fitness function designed to minimize energy expenditure, which is influenced by two key factors: path length and the number of turns. Longer paths require more energy for traversal, making them less efficient, while sharp or frequent turns increase energy consumption due to changes in momentum and direction. The Fitness Score (FS) used in this implementation is given by:

$$FS = \frac{1}{(Path\ length * Number\ of\ Turns\ )^2} \tag{1}$$

The path length (L) is computed by summing the Euclidean distances between consecutive waypoints.

$$L = \sum_{i=1}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_1)^2} \tag{2}$$

Here, n is the total number of waypoints in the path, and the sum represents the total distance covered by the path.

The angle θ between two consecutive segments of a path is the angle formed between two vectors that represent the direction of travel along each segment. For three consecutive points, $P_{(i-2)}, P_{(i-1)}, P_i$, the angle θ is formed by the segments, $P_{(i-2)} \rightarrow P_{(i-1)}$, and $P_{(i-1)} \rightarrow P_i$.

$$\theta = Cos^{-1}\left(\frac{(x_{i-1} - x_{i-2})(x_i - x_{i-1}) + (y_{i-1} - y_{i-2})(y_i - y_{i-1})}{\sqrt{(x_{i-1} - x_{i-2})^2 + (y_{i-1} - y_{i-2})^2}.\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}}\right) \tag{3}$$

D. Selection

The selection process ensures that the best-performing paths (individuals) are given higher chances to reproduce and pass on their traits to the next generation. After evaluating the fitness scores of all paths in the population, the paths are ranked in descending order of their fitness. The selection method used here is elitist selection, where the top paths with the highest fitness scores are selected for reproduction. These "elite" paths represent the most efficient routes based on their distance and number of turns. By focusing on these top paths, the algorithm ensures that the next generation is more likely to inherit favorable traits that result in better performance.

E. Crossover

The crossover phase then combines sections from two well-performing paths to create new offspring paths. This phase increases diversity in the population, potentially producing paths that inherit advantageous traits from each parent, thereby improving the overall fitness of the population. A crossover point is randomly chosen, and segments from two parent paths are swapped to generate new solutions.

F. Mutation

After crossover, mutation introduces minor changes to some paths, including slight deviations in cell selection. This helps the algorithm avoid local optima, exploring new routes that could yield more efficient paths. Mutation involves slightly perturbing waypoint coordinates and adding or removing intermediate waypoints. The mutation process accounts for collision avoidance by ensuring minimum separation between robots.

G. New Population

After the crossover and mutation steps, a new population is created, consisting of the offspring generated from the previous population. This new generation replaces the old one, and the process repeats. The genetic algorithm continues to iterate for a predefined number of generations (in this case, 2000), gradually improving the quality of the population with each cycle.

H. Parameter Tuning

The GA parameters, including population size, mutation rate, and crossover rate, were tuned using a grid search method combined with cross-validation. Different parameter combinations were tested, and the best set was selected based on path length and computational efficiency. This ensures an optimal balance between exploration and exploitation, allowing the algorithm to find efficient paths without excessive computational costs.

To fine-tune the GA parameters, the population size was set to 250, the mutation rate to 0.01, and the crossover rate to 1.0. These values were selected based on preliminary experiments that balanced convergence speed with solution quality.

I. Performance Evaluation

The performance of the hybrid approach is evaluated in comparison to standalone PRM using two key metrics: total path distance and turn count. Total path distance is calculated as the sum of all waypoint-to-waypoint distances along the path, representing the overall length of the journey. Turn count measures the total number of angular deviations in the path, indicating the complexity and smoothness of navigation. The results demonstrate the effectiveness of the genetic algorithm (GA) in enhancing path quality by optimizing both distance and turn count, leading to more efficient and energy-saving paths.

Additionally, the number of robots in the hybrid approach was varied from 2 to 8 to analyze the scalability and effectiveness of the system under different levels of complexity. By increasing the number of robots, the approach was tested for its ability to handle multiple simultaneous path optimizations while maintaining efficient navigation. This variation allowed for a comprehensive evaluation of the algorithm's performance in scenarios with varying coordination demands, further demonstrating its robustness and adaptability.

The proposed hybrid PRM-GA approach offers several key advantages in multi-robot path planning. PRM efficiently generates diverse initial paths, while GA refines them to minimize path length and reduce unnecessary turns, leading to optimized and efficient navigation. By leveraging PRM for initialization and GA for optimization, the method balances solution quality and computational efficiency, ensuring feasibility for real-time applications. Furthermore, the hybrid approach is robust in structured and semi-structured environments, effectively navigating static obstacle configurations while dynamically optimizing paths. This combination of advantages makes the PRM-GA hybrid approach a promising solution for multi-robot path planning in complex environments.

IV. Results and Discussion

The results of the study provide a comprehensive analysis of the hybrid approach's performance in multi-robot path planning. Key findings include the significant improvements achieved in total path distance and turn count when compared to standalone PRM. The hybrid approach demonstrated its effectiveness in optimizing paths for multiple robots, showcasing enhanced efficiency and smoother navigation. Additionally, the scalability of the system was validated by varying the number of robots from 2 to 8, revealing consistent performance across different levels of complexity. These outcomes highlight the robustness of the hybrid method and its potential for real-world applications in dynamic and multi-agent environments.

A. Path Planning for Two Robots

The improved PRM-GA algorithm was initially applied to two robots, each given a distinct color to simplify visual tracking. Robot 1, which started at position (9, 1) and aimed for a goal at (1, 9), was colored red, while Robot 2, starting at (1, 7) with a goal at (7, 1), was colored blue. Fig. 3 displays the paths generated for each robot using both the PRM method (represented by dotted lines) and the optimized PRM-GA method (represented by solid lines), allowing for clear comparison between initial and refined paths.

Table I outlines the positions of each robot at every time step, providing a clear record of their movements. By monitoring the trajectories of Robot 1 and Robot 2 over time, it becomes evident that their paths avoid obstacles and do not overlap, ensuring safe and well-coordinated navigation across the grid.

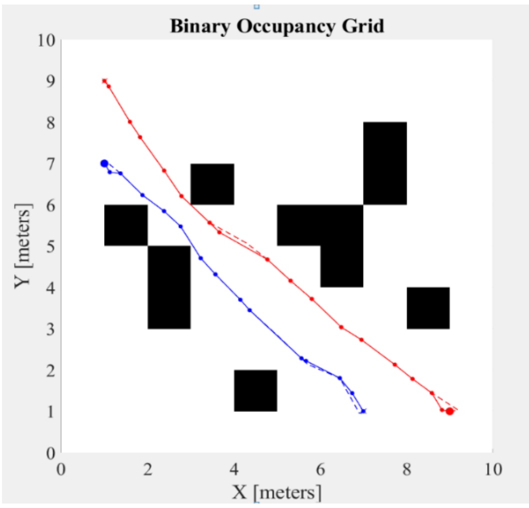

Fig. 3. Path comparison of PRM-GA and Standalone PRM method for two robots path planning

TABLE I. TIMELY POSITIONS OF TWO ROBOTS AFTER APPLYING PRM-GA ALGORITHM

Time Frame	Robot 1		Robot 2	
	x	*y*	*x*	*y*
1	9.0	1.0	1.0	7.0
2	8.7	1.2	1.1	7.0
3	8.3	1.6	1.4	6.7
4	7.6	1.9	2.6	5.7
5	7.2	2.1	2.9	5.4
6	6.6	2.5	3.5	4.7
7	5.8	2.9	3.8	4.4
8	5.7	3.1	4.4	3.7
9	4.9	3.7	5.2	3.0
10	4.4	4.3	5.4	3.0
11	3.7	4.8	6.5	1.8
12	3.6	4.8	6.6	1.6
13	2.6	6.1	7.0	1.3
14	2.3	6.6	7.0	1.0
15	2.0	7.0	7.0	1.0
16	1.4	7.8	7.0	1.0
17	1.3	8.2	7.0	1.0
18	1.1	9.0	7.0	1.0
19	1.0	9.0	7.0	1.0

The bar charts shown in Fig. 4 present a comparison of the total distance and number of turns for the PRM-GA and PRM paths of Robot 1 and Robot 2. For Robot 1, the PRM-GA path has a total distance of 11.7946 units and 11 turns, while the PRM path covers 11.8176 units and involves 14 turns. For Robot 2, the PRM-GA path has a total distance of 8.6621 units with 9 turns, while the PRM path covers 8.9185 units and requires 11 turns. This chart clearly highlights the differences in path length and turns between the two algorithms for each robot, showcasing the efficiency of the PRM-GA paths in terms of both distance and number of turns.

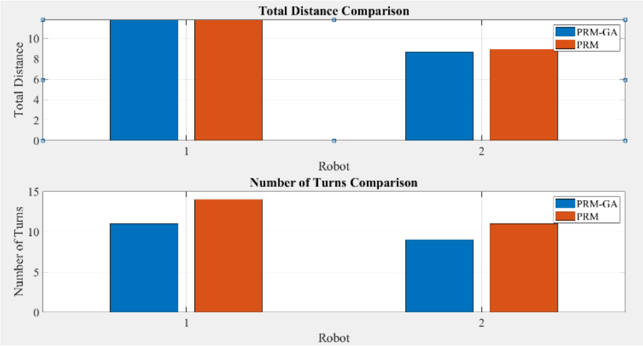

Fig. 4. Performance Comparison of PRM-GA and Standalone PRM method for two robots path planning

B. Path Planning for Three Robots

The enhanced PRM-GA algorithm was applied to three robots. Robot 1, starting at position (9, 1) and aiming for a goal at (1, 9), was colored red. Robot 2, starting at (1, 9) with a goal at (9, 1), was colored blue. Robot 3, beginning at (1, 7) and targeting a goal at (7, 1), was colored magenta. Fig. 5 illustrates the paths generated for each robot using both the PRM method (shown by dotted lines) and the optimized PRM-GA method (represented by solid lines). This allows for a clear comparison between the initial and refined paths for all three robots.

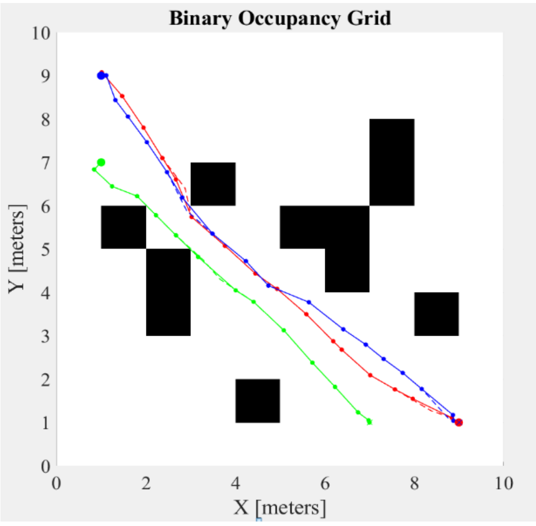

Fig. 5. Path comparison of PRM-GA and Standalone PRM method for three robots path planning

Table II details the positions of each robot at every time step, offering a comprehensive record of their movements. By observing the trajectories of Robot 1, Robot 2, and Robot 3 over time, their paths successfully avoid obstacles and remain non-overlapping, ensuring safe and coordinated navigation across the grid

TABLE II. TIMELY POSITIONS OF THREE ROBOTS AFTER APPLYING PRM-GA ALGORITHM

Time Frame	Robot 1		Robot 2		Robot 3	
	x	y	x	y	x	y
1	9.0	1.0	1.0	9.0	1.0	7.0
2	8.8	1.1	1.1	9.0	0.8	6.8
3	8.0	1.5	1.3	8.4	1.2	6.4
4	7.6	1.8	1.6	8.1	1.8	6.2
5	7.0	2.1	2.0	7.5	2.2	5.8
6	6.4	2.7	2.5	6.8	2.7	5.3
7	6.2	2.9	2.8	6.2	3.2	4.8
8	5.6	3.5	3.5	5.4	4.0	4.0
9	4.9	4.1	4.2	4.7	4.4	3.8
10	4.4	4.4	4.7	4.2	5.1	3.1
11	3.8	5.1	5.6	3.8	5.7	2.4
12	3.0	5.7	6.4	3.1	6.2	1.8
13	2.7	6.6	6.9	2.8	6.7	1.2
14	2.4	7.1	7.3	2.5	7.0	1.0
15	1.9	7.8	7.7	2.1	7.0	1.0
16	1.5	8.5	8.2	1.8	7.0	1.0
17	1.0	9.1	8.9	1.2	7.0	1.0
18	1.0	9.0	8.9	1.0	7.0	1.0
19	1.0	9.0	9.0	1.0	7.0	1.0

Fig. 6 compares the total distance and number of turns for the PRM-GA and PRM paths for three robots. Robot 1's PRM-GA path has 11.6754 units with 10 turns, while the PRM path covers 11.7095 units with 12 turns. Robot 2's PRM-GA path achieves a distance of 11.6172 units with 12 turns, slightly better than the PRM path's 11.6178 units and 13 turns. Robot 3's PRM-GA path is the most efficient, with a distance of 8.7805 units and 8 turns, compared to the PRM path's 8.7869 units and 9 turns. The chart highlights the PRM-GA's overall advantage in optimizing paths.

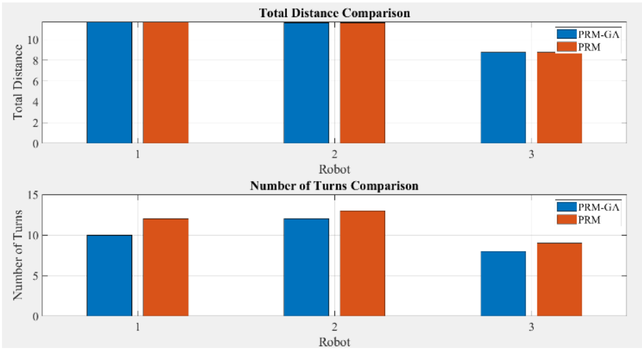

Fig. 6. Performance Comparison of PRM-GA and Standalone PRM method for three robots path planning

C. Path Planning for Four Robots

The improved PRM-GA algorithm was implemented for four robots, each with distinct start and goal positions. Robot 1, starting at (9, 1) and moving toward (1, 9), was represented in red. Robot 2, with a start position of (1, 9) and a goal at (9, 1), was depicted in blue. Robot 3 began at (1, 7) and aimed for (7, 1), shown in green. Finally, Robot 4, starting at (9, 9) and targeting (1, 1), was illustrated in magenta. Fig. 7 showcases the paths generated for each robot using the PRM method (indicated by dotted lines) alongside the optimized PRM-GA paths (shown as solid lines), effectively highlighting the differences between the initial and enhanced paths for all four robots.

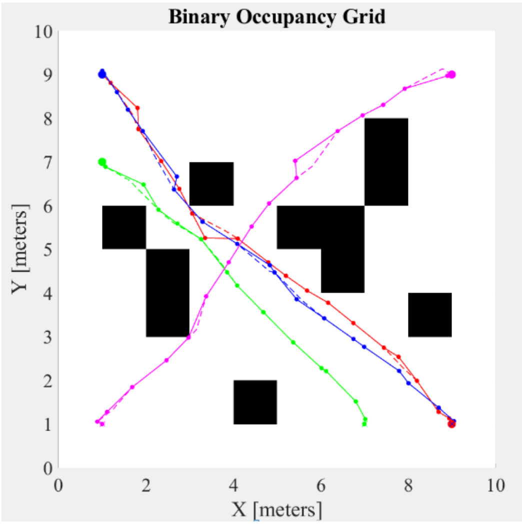

Fig. 7. Path comparison of PRM-GA and Standalone PRM method for four robots path planning

Table III presents the positions of each robot at every time step, offering a detailed record of their movements. By tracking the trajectories of Robots 1, 2, 3, and 4 over time, it is clear that their paths successfully avoid obstacles and remain non-overlapping, ensuring safe and well-coordinated navigation across the grid for all four robots.

TABLE III. TIMELY POSITIONS OF FOUR ROBOTS AFTER APPLYING PRM-GA ALGORITHM

Time Frame	Robot 1		Robot 2		Robot 3		Robot 4	
	x	y	x	y	x	y	x	y
1	9.0	1.0	1.0	9.0	1.0	7.0	9.0	9.0
2	8.9	1.1	1.0	9.1	1.1	6.9	8.9	9.0
3	8.7	1.3	1.3	8.6	1.9	6.5	7.9	8.7
4	8.2	2.0	1.6	8.2	2.3	5.9	7.4	8.3
5	7.8	2.5	1.9	7.7	2.7	5.6	7.0	8.1
6	7.4	2.7	2.7	6.7	3.3	5.2	6.4	7.7
7	6.7	3.3	2.6	6.4	3.9	4.5	5.4	7.0
8	6.2	3.8	3.3	5.6	4.1	4.2	5.5	6.6
9	5.7	4.1	4.1	5.1	4.7	3.6	4.8	6.0
10	5.2	4.4	4.8	4.6	5.4	2.9	4.4	5.5
11	4.8	4.7	4.9	4.5	6.0	2.3	3.9	4.7
12	4.1	5.2	5.5	3.9	6.1	2.2	3.4	3.9
13	3.4	5.3	6.1	3.4	6.8	1.5	3.0	3.0
14	3.1	5.8	6.8	2.9	7.0	1.1	2.5	2.5
15	2.8	6.4	7.0	2.8	7.0	1.0	1.7	1.8
16	2.4	7.0	7.8	2.2	7.0	1.0	1.1	1.3
17	1.8	7.7	8.0	1.9	7.0	1.0	0.9	1.1
18	1.8	8.2	8.7	1.4	7.0	1.0	1.0	1.0
19	1.2	8.8	9.1	1.1	7.0	1.0	1.0	1.0
20	1	9	9	1	7.0	1.0	1.0	1.0

The performance comparison between the PRM and PRM-GA algorithms for Robots 1, 2, 3, and 4 is shown in Fig. 8 highlighting total path distances and the number of turns. For Robot 1, the PRM-GA path covers a distance of 11.8345 units with 11 turns, while the PRM path is slightly shorter at 11.5426 units but involves 14 turns. Similarly, Robot 2's PRM-GA path spans 11.8193 units with 10 turns, compared to the PRM path's 11.7188 units and 13 turns. Robot 3's PRM-GA path covers 8.6697 units with 9 turns, while the PRM path measures 8.6193 units with 11 turns. Lastly, Robot 4 achieves a PRM-GA path of 11.9474 units with 12 turns, compared to the PRM path's 11.9413 units and 15 turns.

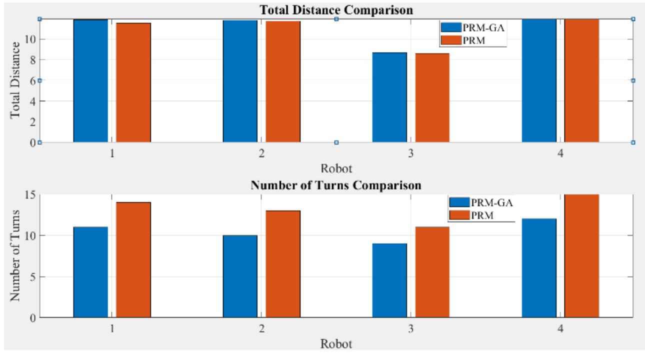

Fig. 8. Performance Comparison of PRM-GA and Standalone PRM method for four robots path planning

D. Path Planning for Five Robots

The enhanced PRM-GA algorithm was implemented for five robots, each with distinct starting positions, goals, and assigned colors. Robot 1, starting at (9, 1) and heading to (1, 9), was assigned red, while Robot 2 began at (1, 9) with a target of (9, 1) and was colored blue. Robot 3 started at (1, 7) and aimed for (7, 1), represented in green. Robot 4 began at (9, 9) and moved towards (1, 1), marked in magenta. Lastly, Robot 5, starting at (7, 2) with a goal at (1, 4), was depicted in cyan. Fig. 9 illustrates the paths generated by both the PRM method (dotted lines) and the optimized PRM-GA method (solid lines), showcasing a clear comparison of the original and improved paths for all five robots.

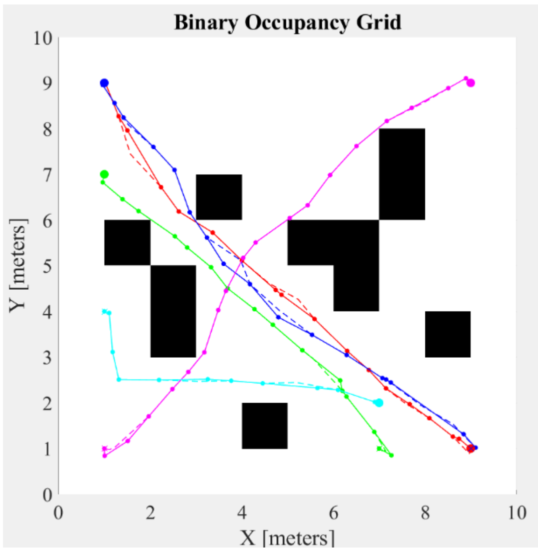

Fig. 9. Path comparison of PRM-GA and Standalone PRM method for five robots path planning

Table IV provides a comprehensive record of the positions of all five robots at each time step, detailing their movements throughout the grid. By analyzing the trajectories of Robots 1, 2, 3, 4, and 5 over time, it is evident that their paths effectively avoid obstacles and remain distinct, ensuring collision-free and well-coordinated navigation for all robots.

The bar charts given in Fig. 10 provide a comparative analysis of the total distances and the number of turns for five robots using the PRM and PRM-GA algorithms. For Robot 1, the PRM-GA method achieved a total distance of 11.5543 with 11 turns, compared to the PRM method with a slightly higher distance of 11.7826 and 16 turns. Similarly, Robot 2 traveled 11.8111 with 11 turns under PRM-GA, whereas PRM recorded 11.8337 with 15 turns. Robot 3 showed notable observations with PRM-GA, completing the path in 9.2706 with 8 turns, while PRM required 8.9328 and 10 turns. Robot 4 experienced a trade-off where PRM-GA covered 12.0445 with 13 turns, and PRM completed the path in 11.8208 but with 15 turns. Finally, Robot 5 demonstrated minimal differences, achieving a distance of 7.3231 with 6 turns under PRM-GA and 7.3309 with 9 turns using PRM.

TABLE IV. Timely Positions of Five Robots After Applying PRM-GA Algorithm

Time Frame	Robot 1		Robot 2		Robot 3		Robot 4		Robot 5	
	x	y	x	y	x	y	x	y	x	y
1	9.0	1.0	1.0	9.0	1.0	7.0	9.0	9.0	7.0	2.0
2	8.7	1.2	1.0	9.0	1.0	6.8	8.9	9.1	6.9	2.0
3	8.6	1.3	1.2	8.6	1.4	6.5	8.5	8.9	6.1	2.3
4	8.1	1.7	1.4	8.2	1.7	6.2	7.7	8.5	5.7	2.3
5	7.7	2.0	2.1	7.6	2.5	5.6	7.2	8.2	4.5	2.4
6	7.1	2.3	2.5	7.1	2.8	5.4	6.5	7.6	3.8	2.5
7	6.8	2.7	2.9	6.2	3.3	5.0	5.9	7.0	3.3	2.5
8	6.3	3.1	3.2	5.6	3.7	4.5	5.4	6.3	2.2	2.5
9	5.6	3.8	3.6	5.0	4.3	4.0	5.0	6.0	1.3	2.5
10	4.9	4.4	4.2	4.6	4.7	3.7	4.3	5.5	1.2	3.1
11	4.7	4.5	4.8	3.9	5.3	3.2	4.0	5.2	1.1	4.0
12	4.0	5.1	5.5	3.5	6.2	2.5	3.7	4.4	1.0	4.0
13	3.4	5.7	6.3	3.0	6.2	2.3	3.5	4.0	1.0	4.0
14	2.6	6.2	7.1	2.5	6.3	2.1	3.2	3.1	1.0	4.0
15	2.2	6.7	7.2	2.5	6.9	1.4	2.8	2.7	1.0	4.0
16	1.5	8.0	7.3	2.4	7.3	0.9	2.5	2.3	1.0	4.0
17	1.3	8.3	8.8	1.3	7.0	1.0	2.0	1.7	1.0	4.0
18	1.0	9.0	9.1	1.0	7.0	1.0	1.5	1.2	1.0	4.0
19	1.0	9.0	9.0	1.0	7.0	1.0	1.0	0.8	1.0	4.0

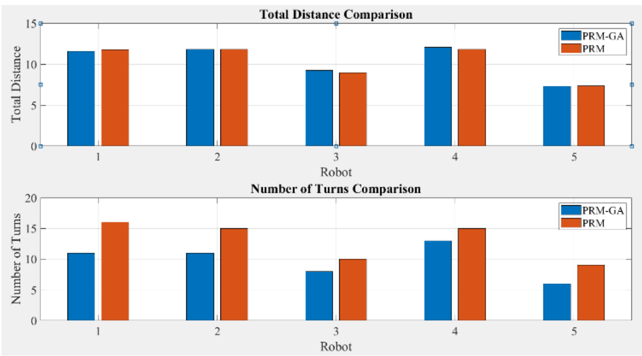

Fig. 10. Performance Comparison of PRM-GA and Standalone PRM method for five robots path planning

E. Path Planning for Six Robots

The enhanced PRM-GA algorithm was applied to six robots. Robot 1, starting at position (9, 1) with a target at (1, 9), is represented in red. Robot 2, beginning at (1, 9) and moving toward (9, 1), is shown in blue. Robot 3, originating from (1, 7) and heading to (7, 1), is depicted in green. Robot 4, starting at (9, 9) and aiming for (1, 1), is illustrated in magenta. Robot 5, starting from (7, 2) and targeting (1, 4), is displayed in cyan. Robot 6, starting at (7, 1) with a goal at (1,7), is highlighted in yellow. Fig. 11 presents the paths generated for all six robots, comparing the PRM method (dotted lines) with the optimized PRM-GA method (solid lines), enabling a clear comparison of the initial and refined paths.

Table V outlines the positions of all six robots at each time step. The analysis of the trajectories for Robots 1, 2, 3, 4, 5 and 6 demonstrates that their paths effectively avoid obstacles and remain distinct, ensuring collision paths.

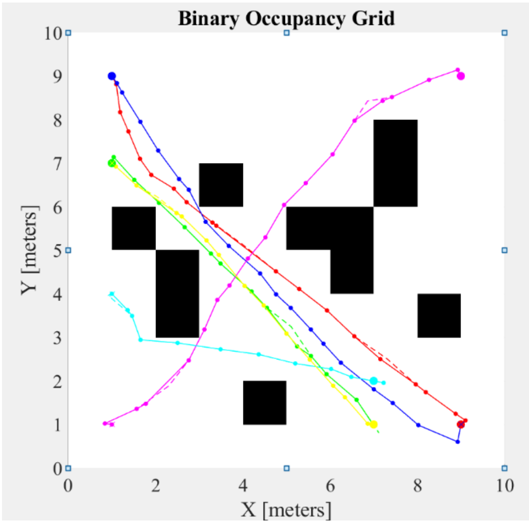

Fig. 11. Path comparison of PRM-GA and Standalone PRM method for six robots path planning

The comparison between the improved PRM-GA algorithm and the standalone PRM shown in Fig. 12 highlights notable improvements in path smoothness and efficiency for multi-robot navigation.

For Robot 1, PRM-GA reduced turns to 8 from 9, with distances remaining comparable (11.8682 vs. 11.588). Robot 2 achieved 9 turns in PRM-GA compared to 10 in PRM, with similar distances (11.7555 vs. 11.7025). Robot 3 saw turns reduced to 8 from 9, with minimal distance change (9.1698 vs. 9.1581). Robot 4 showed a significant reduction in turns (11 vs. 14) with nearly identical distances (11.8622 vs. 11.8946). Robot 5 reduced turns to 7 from 8, with slightly higher distance (6.8447 vs. 6.7471). Robot 6 achieved fewer turns (9 vs. 10) with similar distances (9.1057 vs. 9.0179). These findings demonstrate the PRM-GA algorithm's ability to produce smoother paths with fewer turns while maintaining efficient path lengths, enhancing navigation and coordination in multi-robot systems.

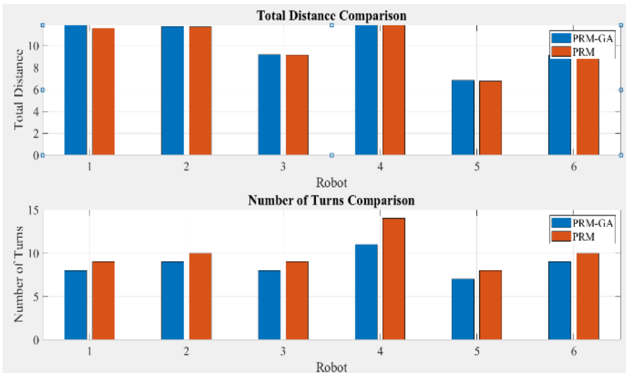

Fig. 12. Path comparison of PRM-GA and Standalone PRM method for six robots path planning

F. Path Planning for Seven Robots

The enhanced PRM-GA algorithm was tested on seven robots. Robot 1, starting at (9, 1) and targeting (1, 9), wacolored red. Robot 2, starting at (1, 9) and aiming for (9, 1), was colored blue. Robot 3, beginning at (1, 7) with a goal at (7, 1), was colored green. Robot 4, starting at (9, 9) and heading to (1, 1), was colored magenta. Robot 5, starting at (7, 2) and aiming for (1, 4), was colored cyan. Robot 6, starting at (7, 1) with a goal at (1, 7), was colored yellow. Lastly, Robot 7, beginning at (1, 4) with a goal at (7, 2), was colored black. Fig. 13 illustrates the paths generated by both the PRM method (dotted lines) and the optimized PRM-GA method (solid lines), providing a clear comparison of the initial and optimized paths for all seven robots.

Table VI outlines the positions of all five robots at each time step, providing a clear view of their movements across the grid. The analysis of the trajectories for Robots 1, 2, 3, 4, 5,6 and 7 demonstrates that their paths effectively avoid obstacles and remain distinct, ensuring safe, collision-free, and coordinated navigation for all robots.

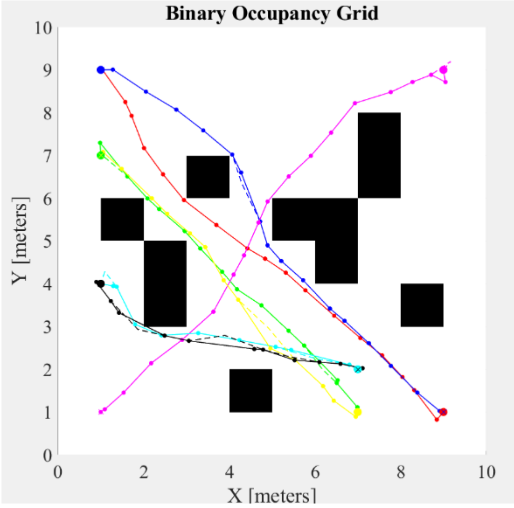

Fig. 13. Path comparison of PRM-GA and Standalone PRM method for seven robots path planning

TABLE V. TIMELY POSITIONS OF SIX ROBOTS AFTER APPLYING PRM-GA

Time Frame	Robot 1		Robot 2		Robot 3		Robot 4		Robot 5		Robot 6	
	x	*y*	*x*	*y*	*x*	*y*	*x*	*y*	*x*	*y*	*x*	*y*
1	9.0	1.0	1.0	9.0	1.0	7.0	9.0	9.0	7.0	2.0	7.0	1.0
2	9.1	1.1	1.1	8.8	1.0	7.1	8.9	9.1	7.2	2.0	6.9	1.0
3	8.9	1.2	1.2	8.6	1.5	6.6	8.3	8.9	6.5	2.1	6.3	1.6
4	8.2	1.7	1.7	7.9	2.1	6.1	7.4	8.5	6.0	2.3	6.1	1.9
5	8.0	1.9	2.1	7.3	2.7	5.5	7.2	8.4	5.2	2.4	5.5	2.5
6	7.2	2.5	2.5	6.6	3.3	4.9	6.6	8.0	4.4	2.6	5.0	3.1
7	6.6	3.0	2.8	6.4	3.5	4.7	6.1	7.2	3.5	2.7	4.5	3.7
8	5.9	3.6	3.2	5.7	4.2	4.1	5.4	6.5	2.5	2.9	4.0	4.2
9	5.3	4.1	3.7	5.1	4.6	3.7	4.9	6.0	1.7	2.9	3.5	4.9
10	4.8	4.5	4.4	4.5	5.2	2.8	4.5	5.3	1.5	3.5	3.2	5.2
11	3.4	5.6	4.8	4.0	5.6	2.6	4.1	4.8	1.4	3.6	2.6	5.8
12	3.3	5.6	5.1	3.7	5.9	2.2	3.7	4.2	1.0	4.0	2.5	5.9
13	2.7	6.1	5.6	3.2	6.6	1.6	3.4	3.9	1.0	4.0	1.6	6.5
14	2.4	6.4	5.9	2.9	7.0	1.0	3.1	3.2	1.0	4.0	1.1	6.9
15	1.9	6.7	6.2	2.4	7.0	1.0	2.8	2.5	1.0	4.0	1.0	7.0
16	1.6	7.1	7.0	1.8	7.0	1.0	1.8	1.5	1.0	4.0	1.0	7.0
17	1.4	7.7	7.4	1.5	7.0	1.0	1.6	1.4	1.0	4.0	1.0	7.0
18	1.2	8.2	8.0	1.0	7.0	1.0	0.8	1.0	1.0	4.0	1.0	7.0
19	1.1	8.8	8.9	0.6	7.0	1.0	1.0	1.0	1.0	4.0	1.0	7.0
20	1.0	9.0	9.0	1.0	7.0	1.0	1.0	1.0	1.0	1.0	1.0	7.0

TABLE VI. TIMELY POSITIONS OF SEVEN ROBOTS AFTER APPLYING PRM-GA

Time Frame	Robot 1		Robot 2		Robot 3		Robot 4		Robot 5		Robot 6		Robot 7	
	x	*y*	*x*	*y*	*x*	*y*	*x*	*y*	*x*	*y*	*x*	*y*	*x*	*y*
1	9.0	1.0	1.0	9.0	1.0	7.0	9.0	9.0	7.0	2.0	7.0	1.0	1.0	4.0
2	8.8	0.8	1.3	9.0	1.0	7.3	9.0	8.7	6.8	2.1	7.0	0.9	0.9	4.0
3	8.3	1.5	2.1	8.5	1.6	6.5	8.7	8.9	5.4	2.5	6.4	1.3	1.2	3.6
4	8.0	1.8	2.8	8.1	2.1	6.0	8.3	8.7	5.1	2.5	6.2	1.6	1.4	3.3
5	7.6	2.3	3.4	7.6	2.4	5.7	7.8	8.5	4.2	2.7	5.5	2.2	2.5	2.8
6	7.1	2.7	4.1	7.0	3.0	5.2	6.9	8.2	3.3	2.8	5.0	2.5	3.1	2.7
7	6.4	3.3	4.3	6.6	3.3	4.8	6.4	7.5	2.4	2.8	4.2	3.6	4.6	2.5
8	5.8	3.8	4.7	5.5	3.8	4.3	5.9	7.0	1.8	3.0	3.9	4.1	4.8	2.5
9	5.3	4.3	4.9	4.9	4.2	3.9	5.4	6.5	1.4	3.9	3.4	4.9	5.5	2.2
10	4.8	4.6	5.2	4.5	4.8	3.5	4.9	5.9	1.3	3.9	3.1	5.2	6.1	2.2
11	4.4	4.8	5.7	4.1	5.4	2.9	4.7	5.4	1.0	4.0	2.6	5.6	6.6	2.1
12	3.7	5.4	6.4	3.4	5.8	2.6	4.3	4.7	1.0	4.0	2.2	6.0	7.1	2.0
13	2.9	6.0	6.7	3.1	6.5	1.7	4.1	4.2	1.0	4.0	1.5	6.7	7.0	2.0
14	2.5	6.6	7.3	2.6	6.5	1.7	3.6	3.3	1.0	4.0	1.1	7.1	7.0	2.0
15	2.0	7.2	7.8	2.1	7.0	1.1	2.9	2.7	1.0	4.0	1.0	7.0	7.0	2.0
16	1.7	7.9	8.4	1.5	7.0	1.0	2.2	2.1	1.0	4.0	1.0	7.0	7.0	2.0
17	1.6	8.2	8.9	1.0	7.0	1.0	1.5	1.5	1.0	4.0	1.0	7.0	7.0	2.0
18	1.0	9.0	9.0	1.0	7.0	1.0	1.1	1.1	1.0	4.0	1.0	7.0	7.0	2.0
19	1.0	9.0	9.0	1.0	7.0	1.0	1.0	1.0	1.0	4.0	1.0	7.0	7.0	2.0

The performance of the PRM-GA algorithm was compared to the standard PRM method for seven robots, highlighting improvements in path distance and the number of turns. The comparison between the improved PRM-GA algorithm and the standalone PRM shown in Fig. 14.

Robot 1 achieved a distance of 11.8536 with 10 turns using PRM-GA, compared to 11.8587 with 12 turns for PRM. Robot 2 followed a path of 11.9419 with 8 turns using PRM GA, versus 11.8221 with 10 turns for PRM. efficiency in multi-robot navigation. Robot 3 completed its route in 8.7879 with 8 turns under PRM-GA, compared to 8.9234 with the same number of turns for PRM. Robot 4 maintained identical results, with a distance of 11.7514 and 10 turns in both methods. Robot 5 saw a reduction to 7.0824 with 8 turns in PRM-GA, from 7.0834 with 9 turns in PRM. Similarly, Robot 6 achieved 8.5616 with 10 turns in PRM-GA, compared to 8.5694 with 11 turns in PRM. Lastly, Robot 7 followed a path of 6.7007 with 9 turns under PRM-GA, slightly outperforming the PRM path of 6.6854 with 10 turns.

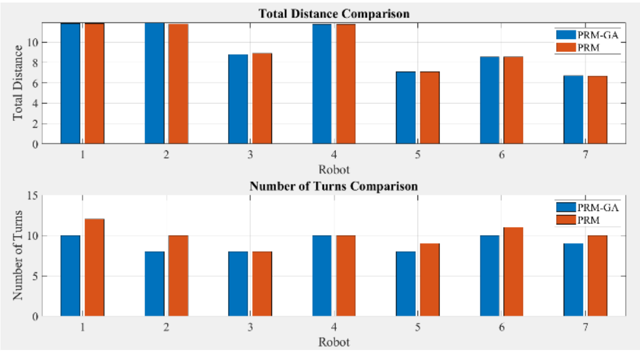

Fig. 14. Performance Comparison of PRM-GA and Standalone PRM method for seven robots path planning

G. Path Planning for Eight Robots

The enhanced PRM-GA algorithm was applied to eight robots, each with a unique start and goal position, represented by distinct colors. Robot 1 traveled from (9, 1) to (1, 9) in red, while Robot 2 moved from (1, 9) to (9, 1) in blue. Robot 3, starting at (1, 7) and heading to (7, 1), was green, and Robot 4, from (9, 9) to (1, 1), was magenta. Robot 5 went from (7, 2) to (1, 4) in cyan, and Robot 6, from (7, 1) to (1, 7), was yellow. Robot 7 traveled from (1, 4) to (7, 2) in black, while Robot 8 moved from (3, 1) to (6, 9) in red. Fig.15 compares the initial PRM paths, shown as dotted lines, with the optimized PRM-GA paths, depicted as solid lines, highlighting the improvements.

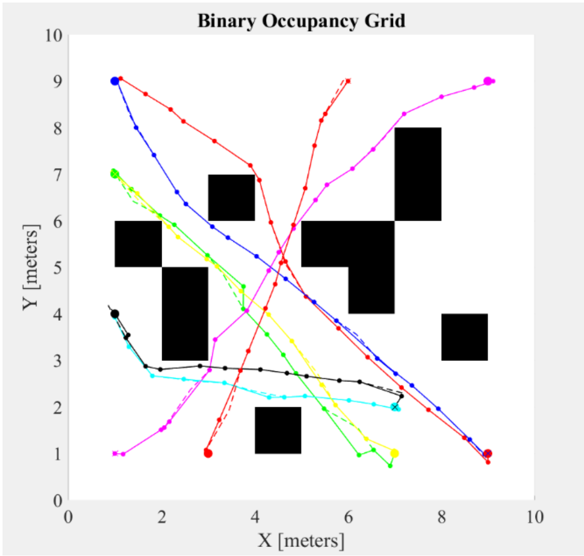

Fig. 15. Path comparison of PRM-GA and Standalone PRM method for eight robots path planning

Table VII presents the positions of each robot at every time step, offering a detailed record of their movements. By tracking the trajectories of Robots 1, 2, 3,4,5,6,7 and 8 over time, it is clear that their paths successfully avoid obstacles and remain non-overlapping, ensuring safe and well-coordinated navigation across the grid for all four robots.

TABLE VII. TIMELY POSITIONS OF SEVEN ROBOTS AFTER APPLYING PRM-GA ALGORITHM

Time Frame	Robot 1		Robot 2		Robot 3		Robot 4		Robot 5		Robot 6		Robot 7		Robot 8	
	x	y	x	y	x	y	x	y	x	y	x	y	x	y	x	y
1	9.0	1.0	1.0	9.0	1.0	7.0	9.0	9.0	7.0	2.0	7.0	1.0	1.0	4.0	3.0	1.0
2	9.0	0.8	1.0	9.0	1.0	7.1	9.1	9.0	7.1	1.9	7.0	1.1	1.3	3.5	3.0	1.1
3	8.5	1.3	1.5	8.0	1.4	6.7	8.7	8.9	6.6	2.1	6.4	1.3	1.2	3.5	3.2	1.7
4	7.7	1.9	1.8	7.4	2.0	6.1	8.0	8.7	6.0	2.1	5.7	2.0	1.7	2.9	3.7	2.8
5	7.1	2.4	2.3	6.6	2.3	5.9	7.2	8.3	5.1	2.2	5.4	2.5	2.0	2.8	3.9	3.2
6	6.4	3.1	2.5	6.4	3.0	5.3	6.5	7.5	4.6	2.2	4.8	3.4	2.8	2.9	4.2	4.1
7	5.8	3.7	3.1	5.9	3.8	4.6	6.1	7.1	4.3	2.2	4.3	4.0	3.4	2.8	4.4	4.6
8	5.1	4.4	3.4	5.6	3.8	4.1	5.5	6.8	3.3	2.5	3.7	4.5	4.1	2.8	4.6	5.1
9	4.7	5.1	4.0	5.2	4.3	3.6	5.3	6.4	2.5	2.6	3.2	5.0	4.7	2.7	4.8	5.9
10	4.3	6.0	4.7	4.8	4.6	3.1	4.8	5.8	1.8	2.7	3.0	5.2	5.1	2.7	5.1	6.7
11	4.1	6.9	5.3	4.3	4.9	2.7	4.5	5.3	1.3	3.3	2.4	5.7	5.8	2.6	5.3	7.6
12	3.9	7.2	5.8	3.9	5.5	2.0	4.3	4.9	1.0	3.9	2.2	5.9	6.3	2.5	5.4	8.2
13	3.1	7.7	7.0	2.7	6.2	1.0	3.8	4.1	1.0	4.0	1.5	6.6	7.2	2.2	5.5	8.3
14	2.5	8.1	6.6	3.0	6.5	1.1	3.2	3.4	1.0	4.0	1.0	7.0	7.0	2.0	6.0	9.0
15	2.2	8.4	7.4	2.5	6.9	0.7	3.0	2.8	1.0	4.0	1.0	7.0	7.0	2.0	6.0	9.0
16	1.7	8.7	7.9	2.0	7.0	1.0	2.2	1.7	1.0	4.0	1.0	7.0	7.0	2.0	6.0	9.0
17	1.1	9.1	8.6	1.3	7.0	1.0	2.1	1.6	1.0	4.0	1.0	7.0	7.0	2.0	6.0	9.0
18	1.0	9.0	8.9	1.0	7.0	1.0	2.0	1.5	1.0	4.0	1.0	7.0	7.0	2.0	6.0	9.0
19	1.0	9.0	9.0	1.0	7.0	1.0	1.2	1.0	1.0	4.0	1.0	7.0	7.0	2.0	6.0	9.0
20	1.0	9.0	9.0	1.0	7.0	1.0	1.0	1.0	1.0	1.0	1.0	7.0	7.0	2.0	6.0	9.0

The results highlight the performance differences between the PRM and the enhanced PRM-GA algorithms for eight robots. For Robot 1, the PRM-GA path covered a distance of 12.0531 with 10 turns, closely matching the PRM path at 12.0576 and 10 turns. Robot 2 saw a slight increase in distance with PRM-GA (12.5789) compared to PRM (11.658) but achieved fewer turns, 10 compared to 12. Robot 3 benefited from a reduced number of turns with PRM-GA (11 versus 14), though the distance slightly increased from 9.1868 to 9.4195. Robot 4 showed a marginal improvement in turns, with PRM-GA requiring 13 compared to 14 for PRM, and a nearly identical distance of around 12.02. For Robot 5, PRM-GA achieved a similar distance (7.0421 vs. 7.0303) with fewer turns (8 compared to 10). Robot 6 had comparable distances for both methods (8.7169 for PRM-GA and 8.7221 for PRM) but required fewer turns with PRM-GA (10 versus 11). demonstrated its capability to optimize paths by either reducing the number of turns or maintaining comparable distances, with improved coordination and efficiency across most cases. Robot 7 demonstrated a reduced distance with PRM-GA (7.2129 vs. 7.7196) while maintaining the same number of turns (10). Finally, Robot 8 achieved a notable improvement with PRM-GA, covering a slightly shorter distance (8.6249 vs. 8.6989) and requiring only 5 turns compared to 7 for PRM. Overall, the PRM-GA algorithm Fig. 16 illustrates the comparison of PRM – GA path and standalone PRM path in terms of total distance and total number of turns on the path for each robot.

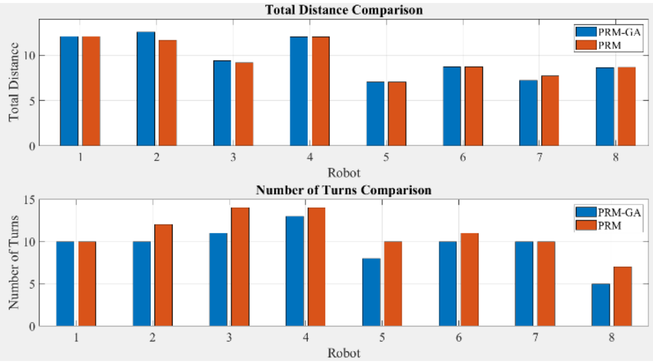

Fig. 16. Performance Comparison of PRM-GA and Standalone PRM method for eight robots path planning

H. *Statistical Performance Analysis of the Proposed PRM - GA Approach*

To evaluate the performance of the proposed PRM-GA approach, statistical comparisons were conducted on key metrics such as path length and turn count. A t-test was performed to assess whether the PRM-GA method significantly improves path efficiency compared to the standard PRM approach. Additionally, the impact of varying robots count on performance was considered to examine scalability and adaptability. The following analysis presents the results and their implications for multi-robot path planning.

Table VIII indicates t- test results to compare the mean path lengths of PRM and PRM-GA in two robot scenario. The p-value indicates no statistically significant difference, though PRM-GA shows a slight reduction in path length variance.

TABLE VIII. PATH LENGTH ANALYSIS FOR TWO ROBOTS

Parameter	PRM Path Length	PRM -GA Path Length
Mean	10.36805	10.22835
Variance	4.20239	4.90628
t Stat	1.19709	
p(T<=t) one-tail	0.22152	
t Critical one-tail	6.31375	
p(T<=t) two-tail	0.44305	
t Critical two-tail	12.70620	

The t-test for turn count shows a reduction in turns for PRM-GA. However, the p-value remains above the significance threshold, suggesting the improvement is not statistically confirmed. A summary of t-test results for turn count analysis for two robots is shown in Table IX.

TABLE IX. TURN COUNT ANALYSIS FOR TWO ROBOTS

Parameter	PRM Path Turn Count	PRM -GA Path Turn Count
Mean	12.50000	10.00000
Variance	4.50000	2.00000
t Stat	5.00000	
p(T<=t) one-tail	0.06283	
t Critical one-tail	6.31375	
p (T<=t) two-tail	0.12567	
t Critical two-tail	12.70620	

The P-values for both metrics were greater than 0.05, indicating no statistically significant difference. While PRM-GA showed reduced turn counts and slightly shorter path lengths, the results suggest that the improvements are not substantial enough to be statistically confirmed with the given data. Further testing with larger datasets is needed to validate the observed trends.

For the three-robot scenario, the analysis compares the PRM and PRM-GA methods based on path length and turn count, using **t-tests** to assess statistical significance. Table X indicates the statistical analysis of path length performance in the proposed approach.

TABLE X. PATH LENGTH ANALYSIS FOR THREE ROBOTS

Parameter	PRM Path Length	PRM -GA Path Length
Mean	10.70473	10.69103
Variance	2.76067	2.73845
t Stat	1.32540	
p(T<=t) one-tail	0.15809	
t Critical one-tail	2.91999	
p(T<=t) two-tail	0.31618	
t Critical two-tail	4.30265	

The mean length for PRM was 10.7047, while for PRM-GA, it was 10.6910, showing only a slight reduction. The t-statistic of 1.3254 and one-tailed p-value of 0.1581 indicate that this difference is not statistically significant, as the p-value is greater than the standard threshold (0.05). Similarly, the two-tailed p-value of 0.3162 confirms that PRM-GA does not significantly improve path length compared to PRM.

However, a notable improvement is observed in turn count. The mean turn count for PRM was 11.3333, while for PRM-GA, it reduced to 10.0000, indicating smoother navigation. The t-statistic of 4.0000 and one-tailed p-value of 0.0286 suggest that this reduction is statistically significant, as the p-value is below 0.05. The two-tailed p-value of

0.0572, though slightly above 0.05, still indicates a meaningful improvement. Table XI showcases the statistical analysis of three robot scenario turn count improvement in the proposed approach.

TABLE XI. TURN COUNT ANALYSIS FOR THREE ROBOTS

Parameter	PRM Path Turn Count	PRM -GA Path Turn Count
Mean	11.3333	10.0000
Variance	4.3333	4.0000
t Stat	4.0000	
p(T<=t) one-tail	0.0286	
t Critical one-tail	2.9200	
p(T<=t) two-tail	0.0572	
t Critical two-tail	4.3027	

For the four-robot scenario, the t-test results for path length are shown in Table XII. The results indicate that PRM-GA produces a slightly longer average path compared to PRM. The t-statistic of -1.7836 suggests that this difference is not statistically significant, as the one-tailed p-value of 0.0862 is greater than 0.05. Similarly, the two-tailed p-value of 0.1725 confirms that PRM-GA does not significantly impact path length in this case.

TABLE XII. PATH LENGTH ANALYSIS FOR FOUR ROBOTS

Parameter	PRM Path Length	PRM -GA Path Length
Mean	10.9555	11.0677
Variance	2.4523	2.5591
t Stat	-1.7836	
p(T<=t) one-tail	0.0862	
t Critical one-tail	2.3534	
p(T<=t) two-tail	0.1725	
t Critical two-tail	3.1824	

However, for turn count, PRM-GA shows a substantial improvement, reducing the mean turn count from 13.2500 (PRM) to 10.5000 (PRM-GA). The t-statistic of 11.0000 and a very low one-tailed p-value of 0.0008 indicate a highly significant reduction, emphasizing the effectiveness of PRM-GA in optimizing smoother paths with fewer turns. The two-tailed p-value of 0.0016 further confirms the significance of this result. The statistical analysis of four robot scenario turn count improvement is shown in Table XIII.

TABLE XIII. TURN COUNT ANALYSIS FOR FOUR ROBOTS

Parameter	PRM Path Turn Count	PRM -GA Path Turn Count
Mean	13.2500	10.5000
Variance	2.9167	1.6667
t Stat	11.0000	
p(T<=t) one-tail	0.0008	
t Critical one-tail	2.3534	
p(T<=t) two-tail	0.0016	
t Critical two-tail	3.1824	

For the five-robot scenario, the t-test results for path length indicate that PRM-GA results in a slightly longer average path compared to PRM. However, the difference is minimal, and the t-statistic of -0.6079 suggests that this variation is not statistically significant. The one-tailed p-value of 0.2880 is much greater than 0.05, and the two-tailed p-value of 0.5761 further confirms the lack of significant difference between the two methods in terms of path length. The t-critical values for both one-taile and two-tailed tests reinforce that the observed difference is within an expected range of variation. The statistical analysis of five robot scenario path length performance is shown in Table XIV.

TABLE XIV. PATH LENGTH ANALYSIS FOR FIVE ROBOTS

Parameter	PRM Path Length	PRM -GA Path Length
Mean	10.3402	10.4007
Variance	4.3850	4.1927
t Stat	-0.6079	
p(T<=t) one-tail	0.2880	
t Critical one-tail	2.1318	
p(T<=t) two-tail	0.5761	
t Critical two-tail	2.7764	

In contrast, for turn count, PRM-GA demonstrates a notable improvement in five robot scenario, reducing the mean turn count from 14 to 10.75. The t-statistic of 4.3333 and a one-tailed p-value of 0.0113 indicate a statistically significant reduction, emphasizing PRM-GA's ability to produce smoother paths with fewer turns. Additionally, the two-tailed p-value of 0.0227 further confirms the significance of the results. The t-critical values for one-tailed and two-tailed tests indicate that the observed difference is substantial. The statistical analysis of five robot scenario path turn count performance is shown in Table XV.

TABLE XV. TURN COUNT ANALYSIS FOR FIVE ROBOTS

Parameter	PRM Path Turn Count	PRM -GA Path Turn Count
Mean	14.0000	10.7500
Variance	7.3333	4.2500
t Stat	4.3333	
p(T<=t) one-tail	0.0113	
t Critical one-tail	2.3534	
p(T<=t) two-tail	0.0227	
t Critical two-tail	3.1824	

Table XVI shows the path length analysis of six robots scenario. In this case, PRM-GA achieves a slightly shorter average path compared to PRM. The t-statistic of 2.3701 suggests a moderate difference between the two methods. The one-tailed p-value of 0.0320 is less than 0.05, indicating a statistically significant improvement in path length using PRM-GA. The two-tailed p-value of 0.0639 is slightly above 0.05, suggesting that the difference may not be as strong when considering both positive and negative deviations. However, the t-critical values for one-tailed **(**2.0150**)** and two-tailed **(**2.5706**)** confirm that PRM-GA provides a meaningful reduction in path length.

TABLE XVI. PATH LENGTH ANALYSIS FOR SIX ROBOTS

Parameter	PRM Path Length	PRM -GA Path Length
Mean	10.1064	10.0126
Variance	4.3077	4.2284
t Stat	2.3701	
p(T<=t) one-tail	0.0320	
t Critical one-tail	2.0150	
p(T<=t) two-tail	0.0639	
t Critical two-tail	2.5706	

For turn count, PRM-GA significantly reduces the mean turn count from 10.0000 (PRM) to 8.6667 (PRM-GA), showing a smoother path with fewer abrupt directional changes. The t-statistic of 4.0000, along with a one-tailed p-value of 0.0052, indicates a highly significant difference in turn count reduction. The two-tailed p-value of 0.0103 further confirms this result. The t-critical values for one-tailed (2.0150) and two-tailed (2.5706) tests reinforce that PRM-GA substantially outperforms PRM in minimizing turns. The statistical analysis of six robot scenario turn count improvement is shown in Table XVII.

TABLE XVII. TURN COUNT ANALYSIS FOR SIX ROBOTS

Parameter	PRM Path Turn Count	PRM -GA Path Turn Count
Mean	10.0000	8.6667
Variance	4.4000	1.8667
t Stat	4.0000	
p(T<=t) one-tail	0.0052	
t Critical one-tail	2.0150	
p(T<=t) two-tail	0.0103	
t Critical two-tail	2.5706	

These findings for six robots suggest that PRM-GA effectively reduces path complexity while also slightly optimizing path length.

In seven robot scenario, for the path length, PRM-GA has a mean of 9.5299, which is very close to PRM's mean of 9.5235. The variances are also similar, with PRM having 5.2870 and PRM-GA having 5.1488. The t-statistic is -0.2297, and the p-value for the two-tailed test is 0.8259, which is much higher than typical significance levels of 0.05 or 0.01. Since the p-value is large, we fail to reject the null hypothesis, meaning there is no statistically significant difference in path length between PRM and PRM-GA. The statistical analysis of seven robot scenario path length performance is shown in Table XVIII.

TABLE XVIII. PATH LENGTH ANALYSIS FOR SEVEN ROBOTS

Parameter	PRM Path Length	PRM -GA Path Length
Mean	9.5235	9.5299
Variance	5.2870	5.1488
t Stat	-0.2297	
p(T<=t) one-tail	0.4130	
t Critical one-tail	1.9432	
p(T<=t) two-tail	0.8259	
t Critical two-tail	2.4469	

For turn count, PRM-GA achieves a lower mean turn count of 9.2857, compared to 10.1429 for PRM in the seven robot scenario. The variance for PRM is 1.8095, while for PRM-GA, it is lower at 0.9048. The t-statistic is 3.2863, and the p-value for the two-tailed test is 0.0167, which is below 0.05. This means that the null hypothesis is rejected, indicating that PRM-GA significantly reduces the number of turns compared to PRM. The statistical analysis of five robot scenario path turn count performance is shown in Table XIX.

In eight robot scenario, the mean path length for PRM is 9.7716, while for PRM-GA, it is slightly lower at 9.5736. The variances are 4.6325 for PRM and 4.3011 for PRM-GA, indicating similar levels of variability. The t-statistic is 1.6065, and the p-value for the two-tailed test is 0.1522, which is greater than the common significance levels of 0.05 or 0.01. This high p-value suggests that there is no statistically significant difference in path length between PRM and PRM-GA, meaning PRM-GA does not provide a meaningful reduction in path length. The statistical analysis of seven robot scenario path length performance is shown in Table XX.

TABLE XIX. TURN COUNT ANALYSIS FOR SEVEN ROBOTS

Parameter	PRM Path Turn Count	PRM -GA Path Turn Count
Mean	10.1429	9.2857
Variance	1.8095	0.9048
t Stat	3.2863	
p(T<=t) one-tail	0.0083	
t Critical one-tail	1.9432	
p(T<=t) two-tail	0.0167	
t Critical two-tail	2.4469	

TABLE XX. PATH LENGTH ANALYSIS FOR EIGHT ROBOTS

Parameter	PRM Path Length	PRM -GA Path Length
Mean	9.7716	9.5736
Variance	4.6325	4.3011
t Stat	1.6065	
p(T<=t) one-tail	0.0761	
t Critical one-tail	1.8946	
p(T<=t) two-tail	0.1522	
t Critical two-tail	2.3646	

For turn count, PRM-GA shows an improvement by achieving a lower mean turn count of 9.6250, compared to 11.0000 for PRM. The variance values are 5.4286 for PRM and 5.4107 for PRM-GA, indicating similar dispersion. The t-statistic is 3.6667, and the p-value for the two-tailed test is 0.0080, which is below 0.05. This indicates that the difference in turn count is statistically significant, suggesting that PRM-GA effectively reduces the number of turns required for navigation. The statistical analysis of five robot scenario path turn count performance is shown in Table XXI.

TABLE XXI. TURN COUNT ANALYSIS FOR EIGHT ROBOTS

Parameter	PRM Path Turn Count	PRM -GA Path Turn Count
Mean	11.0000	9.6250
Variance	5.4286	5.4107
t Stat	3.6667	
p(T<=t) one-tail	0.0040	
t Critical one-tail	1.8946	
p(T<=t) two-tail	0.0080	
t Critical two-tail	2.3646	

I. Computational Complexity

Table XXII shows the navigation times recorded for different numbers of robots to indicate how the computational complexity of the PRM-GA method scales as the number of robots increases. These values represent the average time taken to complete path navigation after running each scenario 10 times. Since only the number of robots varied while keeping other parameters constant, the observed trend directly reflects the impact of robot density on computational performance.

The computational time required for multi-robot path planning increases as the number of robots increases. The data reveals a non-linear growth in computational time, indicating that the complexity of planning paths for multiple robots significantly impacts processing time. For a small number of robots, the computational time remains relatively low at 16.67s and 26.12s, respectively. However, as the number of robots increases, the computational burden grows, reaching 42.72s for 5 robots and 60.29s for 7 robots. The highest recorded computational time is 1.14s for 8 robots, suggesting that beyond a certain number of robots, the increase in computational time becomes more pronounced.

TABLE XXII. TURN COUNT ANALYSIS FOR SEVEN ROBOTS

Number of Robots	Average Computational Time (s)
2	16.67
3	26.12
4	31.56
5	42.72
6	46.38
7	60.29
8	71.14

This trend is likely due to the increasing complexity of collision avoidance, path optimization, and interaction among multiple robots. The algorithm must account for more constraints and dependencies as the number of robots rises, leading to higher computational overhead. Optimizing the algorithm or leveraging parallel computing techniques may help mitigate this growing computational cost in large-scale multi-robot systems.

V. CONCLUSION

The study successfully demonstrated the effectiveness of the PRM-GA approach in optimizing multi-robot path planning by reducing turn counts and enhancing trajectory smoothness across diverse test scenarios. Statistical analysis confirmed that PRM-GA significantly improved path efficiency, particularly in environments with a higher number of robots, where smooth coordination is essential. While reductions in path length were observed in many cases, the improvements were not always consistent, indicating a need for further refinement.

Despite its advantages, the algorithm's performance in highly dynamic environments revealed certain challenges. Frequent obstacle changes sometimes necessitated continuous re-optimization, leading to occasional instances where turn counts and path lengths were comparable to conventional methods. This highlights the complexity of multi-robot coordination in real-time applications and underscores the need for additional enhancements to ensure more consistent path optimization without excessive computational costs.

Future research can expand upon these results by integrating energy and time as additional optimization criteria, alongside distance and turns, to enhance the efficiency and adaptability of path planning algorithms. This approach could directly address industry-specific challenges in logistics, robotics, and autonomous navigation, such as reducing operational costs, improving battery life in autonomous vehicles, and optimizing delivery routes. Highlighting these practical benefits further underscores the real-world significance of the study. Additionally, exploring hybrid approaches that combine PRM-GA with other metaheuristic techniques, such as Ant Colony Optimization or Reinforcement Learning, may lead to improved performance in highly dynamic environments. Further investigation into real-time implementations using parallel computing, GPU acceleration, and distributed systems could enhance computational efficiency, making the algorithm more viable for large-scale applications.

REFERENCES

[1] Y. Tang, S. Qi, L. Zhu, X. Zhuo, Y. Zhang, and F. Meng, "Obstacle avoidance motion in mobile robotics," *Journal of System Simulation*, vol. 36, no. 1, pp. 1–26, 2024.

[2] S. Sabeeh and I. S. Al-Furati, "Comparative analysis of ga-prm algorithm performance in simulation and real-world robotics applications," *Misan Journal of Engineering Sciences*, vol. 2, no. 2, pp. 12–37, 2023.

[3] M. K. Ouach, T. Eren, and E. Ozcan, "Prm path smoothening by circular arc fillet method for mobile robot navigation," *International Journal of Engineering Research and Development*, vol. 16, no. 1, pp. 1–19, 2021.

[4] Y. Yan, "Error recognition of robot kinematics parameters based on genetic algorithms," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 12, pp. 6167–6176, 2020.

[5] 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, p. 120254, 2023.

[6] M. G. Tamizi, M. Yaghoubi, and H. Najjaran, "A review of recent trend in motion planning of industrial robots," *International Journal of Intelligent Robotics and Applications*, vol. 7, no. 2, pp. 253–274, 2023.

[7] A. Loganathan and N. S. Ahmad, "A systematic review on recent advances in autonomous mobile robot navigation," *Engineering Science and Technology, an International Journal*, vol. 40, p. 101343, 2023.

[8] A. N. A. Rafai, N. Adzhar, and N. I. Jaini, "A review on path planning and obstacle avoidance algorithms for autonomous mobile robots," *Journal of Robotics*, vol. 2022, no. 1, p. 2538220, 2022.

[9] H. Hewawasam, M. Y. Ibrahim, and G. K. Appuhamillage, "Past, present and future of path-planning algorithms for mobile robot navigation in dynamic environments," *IEEE Open Journal of the Industrial Electronics Society*, vol. 3, pp. 353–365, 2022.

[10] C. S. Tan, R. Mohd-Mokhtar, and M. R. Arshad, "A comprehensive review of coverage path planning in robotics using classical and heuristic algorithms," *IEEE Access*, vol. 9, pp. 119310–119342, 2021.

[11] K. Sharma and R. Doriya, "Path planning for robots: An elucidating draft," *International journal of intelligent robotics and applications*, vol. 4, no. 3, pp. 294–307, 2020.

[12] J. Ding, Y. Zhou, X. Huang, K. Song, S. Lu, and L. Wang, "An improved rrt* algorithm for robot path planning based on path expansion heuristic sampling," *Journal of Computational Science*, vol. 67, p. 101937, 2023.

[13] M. S. Abed, O. F. Lutfy, and Q. F. Al-Doori, "A review on path planning algorithms for mobile robots," *Engineering and Technology Journal*, vol. 39, no. 5A, pp. 804–820, 2021.

[14] C. Zhou, B. Huang, and P. Fr¨anti, "A review of motion planning algorithms for intelligent robots," *Journal of Intelligent Manufacturing*, vol. 33, no. 2, pp. 387–424, 2022.

[15] 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, p. 101938, 2023.

[16] Z. Yang, N. Li, Y. Zhang, and J. Li, "Mobile robot path planning based on improved particle swarm optimization and improved dynamic window approach," *Journal of Robotics*, vol. 2023, no. 1, p. 6619841, 2023.

[17] C. Huang, Y. Zhao, M. Zhang, and H. Yang, "Apso: An a*-pso hybrid algorithm for mobile robot path planning," *IEEE Access*, vol. 11, pp. 43238–43256, 2023.

[18] F. Gul, W. Rahiman, S. Alhady, A. Ali, I. Mir, and A. Jalil, "Meta heuristic approach for solving multi-objective path planning for au

tonomous guided robot using pso–gwo optimization algorithm with evolutionary programming," *Journal of Ambient Intelligence and Hu manized Computing*, vol. 12, pp. 7873–7890, 2021.

[19] F. Gul, I. Mir, D. Alarabiat, H. M. Alabool, L. Abualigah, and S. Mir, "Implementation of bio-inspired hybrid algorithm with mutation op erator for robotic path planning," *Journal of Parallel and Distributed Computing*, vol. 169, pp. 171–184, 2022.

[20] N. A. K. Zghair and A. S. Al-Araji, "Intelligent hybrid path planning algorithms for autonomous mobile robots," *International Journal of Intelligent Engineering & Systems*, vol. 15, no. 5, 2022.

[21] Q. Yuan, R. Sun, and X. Du, "Path planning of mobile robots based on n improved particle swarm optimization algorithm," *Processes*, vol. no. 1, p. 26, 2022.

[22] S. Han, X. Shan, J. Fu, W. Xu, and H. Mi, "Industrial robot trajectory planning based on improved pso algorithm," in *Journal of Physics: Conference Series*, vol. 1820, no. 1, p. 012185, 2021.

[23] B. Tao and J.-H. Kim, "Mobile robot path planning based on bi population particle swarm optimization with random perturbation strat egy," *Journal of King Saud University-Computer and Information Sciences*, vol. 36, no. 2, p. 101974, 2024.

[24] T. X. Rong, Z. Yukun, and J. XinXin, "Improved a-star algorithm for robot path planning in static environment," in *Journal of Physics: Conference Series*, vol. 1792, no. 1, p. 012067, 2021.

[25] P. Wang, H. Mutahira, J. Kim, and M. S. Muhammad, "Aba*–adaptive bidirectional a* algorithm for aerial robot path planning," *IEEE Access*, vol. 11, pp. 103521-103529, 2023.

[26] J. O. Adibeli, Y.-k. Liu, A. Ayodeji, and N. J. Awodi, "Path planning in nuclear facility decommissioning: Research status, challenges, and opportunities," *Nuclear Engineering and Technology*, vol. 53, no. 11, pp. 3505–3516, 2021.

[27] P. Li, L. Hao, Y. Zhao, and J. Lu, "Robot obstacle avoidance optimization by a* and dwa fusion algorithm," *Plos one*, vol. 19, no. 4, p. e0302026, 2024.

[28] I. A. Hassan, I. A. Abed, and W. A. Al-Hussaibi, "Path planning and trajectory tracking control for two-wheel mobile robot," *Journal of Robotics and Control (JRC)*, vol. 5, no. 1, pp. 1–15, 2024.

[29] T. Wang, L. Wang, D. Li, J. Cai, and Y. Wang, "Monte carlo-based improved ant colony optimization for path planning of welding robot," *Journal of King Saud University-Computer and Information Sciences*, vol. 35, no. 7, p. 101603, 2023.

[30] Y. Chen, J. Wu, C. He, and S. Zhang, "Intelligent warehouse robot path planning based on improved ant colony algorithm," *IEEE Access*, vol. 11, pp. 12360–12367, 2023.

[31] K. Ma, L. Wang, J. Cai, D. Li, A. Wang, and T. Tan, "Robot path planning using fusion algorithm of ant colony optimization and genetic algorithm," *International Journal of Modeling, Simulation, and Scientific Computing*, vol. 14, no. 06, p. 2341032, 2023.

[32] T. Chen, S. Chen, K. Zhang, G. Qiu, Q. Li, and X. Chen, "A jump point search improved ant colony hybrid optimization algorithm for path planning of mobile robot," *International Journal of Advanced Robotic Systems*, vol. 19, no. 5, p. 17298806221127953, 2022.

[33] N. Abu, W. Bukhari, M. Adli, and A. Ma'arif, "Optimization of an autonomous mobile robot path planning based on improved genetic algorithms," *Journal of Robotics and Control (JRC)*, vol. 4, no. 4, pp. 557–571, 2023.

[34] T. Mohanraj, T. Dinesh, B. Guruchandhramavli, S. Sanjai, and B. She shadhri, "Mobile robot path planning and obstacle avoidance using hybrid algorithm," *International Journal of Information Technology*, vol. 15, no. 8, pp. 4481–4490, 2023.

[35] K. Suresh, K. Ravichandran, and S. Venugopal, "Multi-objective genetic algorithm for mobile robot path planning in industrial automation," *Journal of Intelligent & Fuzzy Systems*, vol. 44, no. 4, pp. 6829–6842, 2023.

[36] T.-W. Zhang, G.-H. Xu, X.-S. Zhan, and T. Han, "A new hybrid algorithm for path planning of mobile robot," *The Journal of Super computing*, vol. 78, no. 3, pp. 4158–4181, 2022.

[37] Z. Fang and X. Liang, "Intelligent obstacle avoidance path planning method for picking manipulator combined with artificial potential field method," *Industrial Robot: the international journal of robotics research and application*, vol. 49, no. 5, pp. 835–850, 2022.

[38] L. Zhai and S. Feng, "A novel evacuation path planning method based on improved genetic algorithm," *Journal of Intelligent & Fuzzy Systems*, vol. 42, no. 3, pp. 1813–1823, 2022.

[39] K. Suresh, R. Venkatesan, and S. Venugopal, "Mobile robot path planning using multi-objective genetic algorithm in industrial automation," *Soft Computing*, vol. 26, no. 15, pp. 7387–7400, 2022.

[40] 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.

[41] A. Hentout, A. Maoudj, and M. Aouache, "A review of the literature on fuzzy-logic approaches for collision-free path planning of manipulator robots," *Artificial Intelligence Review*, vol. 56, no. 4, pp. 3369–3444, 2023

[42]] A. K. Rath, D. R. Parhi, H. C. Das, P. B. Kumar, and M. K. Mahto, "Design of a hybrid controller using genetic algorithm and neural network for path planning of a humanoid robot," *International Journal of Intelligent Unmanned Systems*, vol. 9, no. 3, pp. 169–177, 2021.

[43] N. AbuJabal, T. Rabie, M. Baziyad, I. Kamel, and K. Almazrouei, "Path planning techniques for real-time multi-robot systems: A systematic review," *Electronics*, vol. 13, no. 12, p. 2239, 2024.

[44] K. Sharma, S. Singh, and R. Doriya, "Optimized cuckoo search algorithm using tournament selection function for robot path planning," *International Journal of Advanced Robotic Systems*, vol. 18, no. 3, p. 1729881421996136, 2021.

[45] K. Sriniketh, A. V. Le, R. E. Mohan, B. J. Sheu, V. D. Tung, P. Van Duc, and M. B. Vu, "Robot-aided human evacuation optimal path planning for fire drill in buildings," *Journal of Building Engineering*, vol. 72, p. 106512, 2023.

[46] M. A. R. Pohan and J. Utama, "Smart probabilistic road map (smart prm): fast asymptotically optimal path planning using smart sampling strategies," *Jordanian Journal of Computers and Information Technology*, vol. 10, no. 2, 2024.

[47] M. N. Ab Wahab, A. Nazir, A. Khalil, W. J. Ho, M. F. Akbar, M. H. M. Noor, and A. S. A. Mohamed, "Improved genetic algorithm for mobile robot path planning in static environments," *Expert Systems with Applications*, vol. 249, p. 123762, 2024.

[48] S. Kumar and A. Sikander, "A modified probabilistic roadmap algorithm for efficient mobile robot path planning," *Engineering Optimization*, vol. 55, no. 9, pp. 1616–1634, 2023

[49] D. Li, L. Wang, J. Cai, A. Wang, T. Tan, and J. Gui, "Research on path planning of mobile robot based on improved genetic algorithm," *International Journal of Modeling, Simulation, and Scientific Computing*, vol. 14, no. 06, p. 2341030, 2023.

[50] Z. Jin, G. Luo, R. Wen, and J. Huang, "Woa-aga algorithm design for robot path planning," *International Journal Of Computers Communications & Control*, vol. 18, no. 5, 2023.

[51] K. Shi, Z. Wu, B. Jiang, and H. R. Karimi, "Dynamic path planning of mobile robot based on improved simulated annealing algorithm," *Journal of the Franklin Institute*, vol. 360, no. 6, pp. 4378–4398, 2023.

[52] R. Z. Khaleel, H. Z. Khaleel, A. A. Abdullah Al-Hareeri, A. S. Mahdi Al-Obaidi, and A. J. Humaidi, "Improved trajectory planning of mobile robot based on pelican optimization algorithm," *Journal Europ´een des Syst`emes Automatis´es*, vol. 57, no. 4, 2024.

[53] F. Hamad, H. N. Fakhouri, F. Alzghoul, and J. Zraqou, "Development and design of object avoider robot and object, path follower robot based on artificial intelligence," *Arabian Journal for Science and Engineering*, pp. 1–22, 2024.

[54] A. Zou, L. Wang, W. Li, J. Cai, H. Wang, and T. Tan, "Mobile robot path planning using improved mayfly optimization algorithm and dynamic window approach," *The Journal of Supercomputing*, vol. 79, no. 8, pp. 8340–8367, 2023.

[55] L. Gao, R. Liu, F. Wang, W. Wu, B. Bai, S. Yang, and L. Yao, "An advanced quantum optimization algorithm for robot path planning," *Journal of Circuits, Systems and Computers*, vol. 29, no. 08, p. 2050122, 2020.

[56]] Z. Elmi and M. ¨ O. Efe, "Online path planning of mobile robot using grasshopper algorithm in a dynamic and unknown environment," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 33, no. 3, pp. 467–485, 2021.

[57] R.-B. Wang, W.-F. Wang, L. Xu, J.-S. Pan, and S.-C. Chu, "An adaptive parallel arithmetic optimization algorithm for robot path planning," *Journal of advanced transportation*, vol. 2021, no. 1, p. 3606895, 2021.

[58] S. Kumar and A. Sikander, "Optimum mobile robot path planning using improved artificial bee colony algorithm and evolutionary program ming," *Arabian Journal for Science and Engineering*, vol. 47, no. 3, pp. 3519–3539, 2022.

[59] J. Zan, "Research on robot path perception and optimization technology based on whale optimization algorithm," *Journal of Computational and Cognitive Engineering*, vol. 1, no. 4, pp. 201–208, 2022.

[60] W. Min, L. Mo, B. Yin, and S. Li, "An improved cuckoo search algorithm and its application in robot path planning," *Applied Sciences*, vol. 14, no. 20, 2024.

[61] Z. Wang, Z. Wu, L. Si, K. Tong, and C. Tan, "A novel path planning method of mobile robots based on an improved bat algorithm," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 235, no. 16, pp. 3071–3086, 2021.

[62] Z. Chen, J. Walters, G. Xiao, and S. Li, "An enhanced gru model with application to manipulator trajectory tracking," *EAI Endorsed Transactions on AI and Robotics*, vol. 1, no. 1, 2022.

[63] O. A. Razzaq Abdul Wahhab and A. S. Al-Araji, "Path planning and control strategy design for mobile robot based on hybrid swarm optimization algorithm," *International Journal of Intelligent Engineering & Systems*, vol. 14, no. 3, 2021.

[64] A. M. Bacha, R. B. Zamoum, and F. Lachekhab, "Machine Learning Paradigms for UAV Path Planning: Review and Challenges," *Journal of Robotics and Control (JRC)*, vol. 6, no. 1, pp. 215-233, 2025.

[65] C. E. Okereke, M. M. Mohamad, N. H. A. Wahab, O. Elijah, A. Al-Nahari and S. Zaleha.H, "An Overview of Machine Learning Techniques in Local Path Planning for Autonomous Underwater Vehicles," in *IEEE Access*, vol. 11, pp. 24894-24907, 2023, doi: 10.1109/ACCESS.2023.3249966.

[66] A. A. N. Kumaar and S. Kochuvila, "Mobile Service Robot Path Planning Using Deep Reinforcement Learning," in *IEEE Access*, vol. 11, pp. 100083-100096, 2023, doi: 10.1109/ACCESS.2023.3311519.

[67] K. Yang and L. Liu, "An Improved Deep Reinforcement Learning Algorithm for Path Planning in Unmanned Driving," in *IEEE Access*, vol. 12, pp. 67935-67944, 2024, doi: 10.1109/ACCESS.2024.3400159.

[68] M. Muzammul, M. Assam, Y. Y. Ghadi, N. Innab, M. Alajmi and T. J. Alahmadi, "IR-QLA: Machine Learning-Based Q-Learning Algorithm Optimization for UAVs Faster Trajectory Planning by Instructed-Reinforcement Learning," in *IEEE Access*, vol. 12, pp. 91300-91315, 2024, doi: 10.1109/ACCESS.2024.3420169.

[69] M. Ganesan, S. Kandhasamy, B. Chokkalingam and L. Mihet-Popa, "A Comprehensive Review on Deep Learning-Based Motion Planning and End-to-End Learning for Self-Driving Vehicle," in *IEEE Access*, vol. 12, pp. 66031-66067, 2024, doi: 10.1109/ACCESS.2024.3394869.

[70] J. -M. Alkazzi and K. Okumura, "A Comprehensive Review on Leveraging Machine Learning for Multi-Agent Path Finding," in *IEEE Access*, vol. 12, pp. 57390-57409, 2024, doi: 10.1109/ ACCESS .2024.3392305.

[71] S. Y. Luis, D. G. Reina and S. L. T. Marín, "A Multiagent Deep Reinforcement Learning Approach for Path Planning in Autonomous Surface Vehicles: The Ypacaraí Lake Patrolling Case," in *IEEE Access*, vol. 9, pp. 17084-17099, 2021, doi: 10.1109/ACCESS.2021.3053348.

[72] A. M. Barrionuevo, S. Yanes Luis, D. Gutiérrez Reina and S. L. Toral Marín, "Informative Deep Reinforcement Path Planning for Heterogeneous Autonomous Surface Vehicles in Large Water Resources," in *IEEE Access*, vol. 12, pp. 71835-71852, 2024, doi: 10.1109/ACCESS.2024.3402980.

[73] W. A. H. Sandanika, S. H. Wishvajith, S. Randika, D. A. Thennakoon, S. K. Rajapaksha, and V. Jayasinghearachchi, "Ros-based multi-robot system for efficient indoor exploration using a combined path planning technique," *Journal of Robotics and Control (JRC)*, vol. 5, no. 5, pp. 1241-1260, 2024.

[74] J. Bodaragama and U. U. S. Rajapaksha, "Path Planning for Moving Robots in an Unknown Dynamic Area Using RND-Based Deep Reinforcement Learning," *2023 3rd International Conference on Advanced Research in Computing (ICARC)*, Belihuloya, Sri Lanka, 2023, pp. 13-18, doi: 10.1109/ICARC57651.2023.10145730.

[75] T. B. Ionescu, "Web-based simulation and motion planning for human-robot and multi-robot applications," *International Journal of Computer Integrated Manufacturing*, vol. 38, no. 2, pp. 177–199, 2024.

[76] U. S. Rajapaksha, C. Jayawardena, and B. A. MacDonald, "Design, Implementation, and Performance Evaluation of a Web-Based Multiple Robot Control System," *Journal of Robotics*, vol. 2022, no. 1, p. 9289625, 2022.

[77] J. M. Santana *et al*., "Design and Implementation of an Interactive System for Service Robot Control and Monitoring," *Sensors*, vol. 25, no. 4, p. 987, 2025.

[78] N. Jayasekara, B. Kulathunge, H. Premaratne, I. Nilam, S. Rajapaksha, and J. Krishara, "Revolutionizing Accessibility: Smart Wheelchair Robot and Mobile Application for Mobility, Assistance, and Home Management," *Journal of Robotics and Control (JRC)*, vol. 5, no. 1, pp. 27-53, 2024.

[79] V. D. Cong, "Path following and avoiding obstacle for mobile robot under dynamic environments using reinforcement learning," *Journal of Robotics and Control (JRC)*, vol. 4, no. 2, pp. 157-164, 2023.

[80] M. Panagoda *et al*., "Moving Robots in Unknown Environments Using Potential Field Graphs," *2022 2nd International Conference on Advanced Research in Computing (ICARC)*, pp. 96-101, 2022, doi: 10.1109/ICARC54489.2022.9754182.

[81] M. N. A. Wahab, C. M. Lee, M. F. Akbar, and F. H. Hassan, "Path Planning for Mobile Robot Navigation in Unknown Indoor Environments Using Hybrid PSOFS Algorithm," in *IEEE Access*, vol. 8, pp. 161805-161815, 2020, doi: 10.1109/ACCESS.2020.3021605.

[82] K. K. P. Gayashani, U. U. S. Rajapaksha, and C. Jayawardena, "Moving a Robot In Unknown Areas Without Collision Using Robot Operating System," *2022 2nd International Conference on Advanced Research in Computing (ICARC)*, pp. 84-89, 2022, doi: 10.1109/ICARC54489.2022.9754138.

[83] U. Orozco-Rosas, K. Picos, J. J. Pantrigo, A. S. Montemayor, and A. Cuesta-Infante, "Mobile Robot Path Planning Using a QAPF Learning Algorithm for Known and Unknown Environments," in *IEEE Access*, vol. 10, pp. 84648-84663, 2022, doi: 10.1109/ACCESS.2022.3197628.

[84] M. Zhao, H. Lu, S. Yang, and F. Guo, "The Experience-Memory Q-Learning Algorithm for Robot Path Planning in Unknown Environment," in *IEEE Access*, vol. 8, pp. 47824-47844, 2020, doi: 10.1109/ACCESS.2020.2978077.

[85] N. Abdalmanan *et al*., "2D LiDAR Based Reinforcement Learning for Multi-Target Path Planning in Unknown Environment," in *IEEE Access*, vol. 11, pp. 35541-35555, 2023, doi: 10.1109/ACCESS. 2023.3265207.

[86] J. Bodaragama and U. U. S. Rajapaksha, "Path Planning for Moving Robots in an Unknown Dynamic Area Using RND-Based Deep Reinforcement Learning," *2023 3rd International Conference on Advanced Research in Computing (ICARC)*, pp. 13-18, 2023, doi: 10.1109/ICARC57651.2023.10145730.

[87] T. Jiao, C. Hu, L. Kong, X. Zhao, and Z. Wang, "An Improved HM-SAC-CA Algorithm for Mobile Robot Path Planning in Unknown Complex Environments," in *IEEE Access*, vol. 13, pp. 21152-21163, 2025, doi: 10.1109/ACCESS.2025.3535728.

[88] M. Visca, R. Powell, Y. Gao, and S. Fallah, "Deep Meta-Learning Energy-Aware Path Planner for Unmanned Ground Vehicles in Unknown Terrains," in *IEEE Access*, vol. 10, pp. 30055-30068, 2022, doi: 10.1109/ACCESS.2022.3155161.

[89] M. Balza, M. A. Goldbarg, S. N. Silva, L. M. D. Silva, and M. A. C. Fernandes, "A Real-Time Safe Navigation Proposal for Mobile Robots in Unknown Environments Using Meta-Heuristics," in *IEEE Access*, vol. 13, pp. 23987-24013, 2025, doi: 10.1109/ACCESS.2025.3536081.