Misan Journal of Engineering Sciences ISSN: 2957-4250
Vol. 2, No. 2, December 2023 ISSN-E: 2957-4242

Comparative Analysis of GA-PRM Algorithm Performance in Simulation and Real-World Robotics Applications

Sarah Sabeeh *[1], Israa S. Al-Furati [2]

[1] *Computers Engineering Department, University of Basrah, Basrah, Iraq*
[2] *Electrical Engineering Department, University of Basrah, Basrah, Iraq*

**Corresponding author E-mail: israa.sabri@uobasrah.edu.iq*

(Received 5 Oct 2023, Revised 25 Oct 2023, Accepted 26 Oct 2023)

Abstract: This paper presents a comprehensive analysis of the performance of the Genetic Algorithm Probabilistic Roadmap (GA-PRM) algorithm in both simulated and real-world robotic environments. The GA-PRM algorithm is a promising approach for robot path planning, and understanding its behavior in different settings is crucial for its practical applications. In simulations, we explore the advantages of controlled and reproducible test conditions, allowing for extensive parameter tuning and algorithm improvement. Real-world testing is employed to validate the algorithm's performance in actual robotic environments, taking into account the inherent complexities and uncertainties present. In our comparative analysis, we found that the GA-PRM algorithm demonstrates significant improvements in real-world scenarios compared to simulations. Specifically, the algorithm produced shorter paths in real-world robot testing, with an average length of 21.428 cm, as opposed to 25.6235 units in simulations. Moreover, the computational efficiency of the algorithm was notably enhanced in the real-world environment, where it took only 0.375 seconds on average to plan paths, compared to 0.6881 seconds in simulations. The algorithm also exhibited higher path smoothness in the real world, with an average smoothness score of 0.432, compared to 0.3133 in simulations. These results underscore the algorithm's adaptability to real-world conditions and its potential for efficient navigation in practical healthcare and automation applications. Our research bridges the gap between simulation and reality, facilitating the development of more reliable and adaptable robotic systems. The insights gained from this comparative evaluation contribute to a deeper understanding of the GA-PRM algorithm's behavior and its potentials.

Keywords: Algorithm Performance, Collision Avoidance, Dynamic Obstacles, GA-PRM, Genetic Algorithm, Healthcare Robots, Path Planning, Real-World Testing, Robot Navigation, Simulation

1. Introduction

Robotic systems have garnered significant attention in various fields, from healthcare to manufacturing, owing to their versatility and potential to solve complex tasks. An essential component of robotic autonomy is path planning, the ability to determine efficient routes from a starting point to a destination while avoiding obstacles. The Genetic Algorithm-based Probabilistic Roadmap (GA-PRM) algorithm is one such path planning approach known for its effectiveness. In the world of robotics research, evaluating the performance of path planning algorithms is crucial. It allows to assess their suitability for real-world applications. Traditionally, researchers have relied on simulations to validate and fine-tune algorithms before deploying them on physical robots. However, the gap between simulation and reality poses challenges in ensuring that algorithms perform as expected in practical scenarios.

This paper examines a fundamental aspect of robotics research—comparing how the GA-PRM algorithm behaves in simulated environments and when deployed on a physical robot. Through conducting a comparative analysis, we aim to shed light on the algorithm's strengths, limitations, and its adaptability to real-world challenges.

Our study is motivated by the growing need for reliable robotic systems in diverse applications, including autonomous vehicles, medical robots, and industrial automation. The ability to navigate safely and efficiently in dynamic environments is paramount for such systems.

Throughout this paper, we present the results of experiments conducted both in simulation and on a real-world robotic platform. We analyze the algorithm's performance metrics, discuss observed differences, and explore the implications of these findings. Our goal is to contribute valuable insights that inform the deployment of path planning algorithms in practical, real-world settings. As we begin this comparative analysis endeavor, we will explore the methodology used, the criteria for selecting scenarios, the presentation of simulation and real-world results, and a comprehensive discussion of our findings. We hope this research will guide future efforts in enhancing the alignment between simulation and real-world robotics applications.

1.1 GA-PRM algorithm and its significance

The significance of GA-PRM stems from its ability to harness the strengths of genetic algorithms and probabilistic roadmaps. They generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover [1]. Probabilistic roadmaps, on the other hand, are excellent at finding obstacle-free paths [2]. Through merging these techniques, the algorithm equips robots to navigate complex environments effectively, even in demanding situations. Additionally, the algorithm's real-time decision-making and dynamic obstacle handling capabilities place it at the forefront of modern research, enhancing the autonomy and efficiency of robots. As robotic technology continues to integrate into various industries, GA-PRM is expected to play a pivotal role in shaping the future of autonomous systems. Its capacity to enhance path planning and manage dynamic terrains has far-reaching implications, not only for robotics but also for broader technological advancements.

1.2 Importance of Dual Evaluation: Simulated and Real-World Environments

Path planning algorithms serve as the backbone of robotics, enabling autonomous robots to navigate complex and diverse environments. However, assessing the true potential and reliability of these algorithms necessitates a dual evaluation approach, encompassing both simulated and real-world settings. This approach holds substantial importance for several compelling reasons.

Firstly, it bridges the gap between theoretical prowess and practical applicability. Simulated environments offer a controlled and cost-effective platform for initial algorithm testing and development. Yet, the real world introduces a host of unpredictable challenges, including sensor inaccuracies, environmental variations, and dynamic obstacles. Relying solely on simulation may lead to solutions that excel in theory but struggle when confronted with real-world complexities [3].

Secondly, it emphasizes the critical traits of robustness and adaptability. Real-world environments are inherently capricious, demanding adaptability from robotic systems. Dual evaluation enables a comprehensive assessment of how path-planning algorithms react to dynamic changes and unexpected obstacles. Algorithms highlighting resilience in real-world trials are better suited to address unforeseen challenges and uphold safety [4].

Additionally, it places a spotlight on the pivotal role of sensors and their integration. Sensors are instrumental in a robot's perception of its surroundings. Real-world evaluation aids in refining sensor integration and calibration, addressing challenges stemming from sensor noise, occlusions, and inherent limitations. Algorithms proficient at harnessing sensor data in real-world scenarios are better prepared for success in practical applications [5].

Lastly, it facilitates benchmarking and comparative analysis. Comparing results between simulation and real-world experimentation offers valuable insights into discrepancies in algorithm performance. Such insights guide researchers in refining simulation models, enhancing sensor accuracy, and optimizing algorithmic efficiency. This iterative process leads to the development of more reliable and practical robotic solutions [6].

1.3 Objectives and Contributions

The primary goal of this research is to comprehensively assess the performance of the GA-PRM path-planning algorithm in two distinct environments: simulation and real-world robotics applications. To achieve this overarching goal, the following specific objectives have been set:

- **Objective 1:** To assess the performance of the GA-PRM algorithm in simulated environments.
- **Objective 2:** To evaluate how well the GA-PRM algorithm operates in real-world robotic applications.

To achieve these objectives, we aim to:

- Compare the algorithm's performance in simulated scenarios with its performance in real-world settings.
- Identify any differences or challenges encountered when deploying the algorithm in the physical world.

In terms of research contributions, this research makes several noteworthy contributions to the field of robotics and path planning algorithm evaluation:

1. **Insight into Algorithm Behavior:** We provide valuable insights into how the GA-PRM algorithm behaves in controlled simulations and real-world situations.
2. **Robustness Assessment:** Through conducting comparative analyses, we assess the algorithm's robustness and its ability to adapt to dynamic environments.
3. **Practical Guidance:** Researchers and practitioners can benefit from our findings to make informed decisions when implementing path-planning algorithms in real robots.

This research seeks to advance our understanding of path planning algorithms by conducting a thorough evaluation that encompasses both simulation and real-world robotics applications. The insights gained from this study contribute to the ongoing efforts to develop more effective and reliable robotic navigation solutions.

1.4 Related Work

Path planning is a fundamental component of robotics, playing a pivotal role in enabling robots to autonomously navigate and complete tasks in various environments. This section provides an overview of key path planning algorithms and discusses their significance in the field of robotics. The review is structured to offer insights into the evolution of path planning techniques, their applications, and their relevance to the GA-PRM algorithm.

1.4.1 Historical Evolution of Path Planning Algorithms

Path planning algorithms have witnessed significant advancements over the years, driven by the continuous growth of robotics research and applications. Early path planning algorithms primarily focused on deterministic methods, such as Dijkstra's algorithm and A* search, which guarantee optimal paths in structured environments [7].

As robotic systems expanded into unstructured and dynamic environments, probabilistic algorithms gained prominence. Techniques like Rapidly-exploring Random Trees (RRT) [8] and Probabilistic Roadmap Methods (PRM) [9] emerged to address the challenges posed by complex spaces and dynamic obstacles. These probabilistic approaches revolutionized path planning by offering solutions that balance efficiency and optimality.

1.4.2 Advancements in Path Planning Algorithms

The field of path planning has witnessed remarkable growth, primarily driven by the integration of artificial intelligence (AI) techniques and advancements in sensor technologies. These developments have led to significant improvements in the capabilities of path planning algorithms, making them increasingly essential in the realm of robotics.

Sampling-based Algorithms, like RRT* [10] and PRM* [3], extend the capabilities of RRT and PRM by ensuring asymptotic optimality. These methods are particularly valuable when optimality is crucial.

Machine Learning-based Approaches, such as Deep reinforcement learning (DRL) techniques, have been applied to path planning, enabling robots to learn adaptive navigation policies [11]. DRL-based algorithms demonstrate promise in complex, real-world scenarios.

Hybrid algorithms combine traditional planning methods with learning-based approaches to harness the strengths of both. These methods aim to achieve the best of both worlds in terms of efficiency and adaptability. GA-PRM algorithm is considered a type of this class of path planning algorithms.

These advancements collectively stress the dynamic nature of path planning algorithms, with each approach offering unique advantages and applications. As the GA-PRM algorithm integrates genetic algorithms with probabilistic roadmap methods, it contributes to this evolving landscape by addressing the challenges posed by modern robotics applications.

1.4.3 Path Planning in Healthcare

Path planning algorithms find applications in various domains, including healthcare robotics. In the medical field, robots are deployed for tasks such as surgery, patient care, and drug delivery. For example, autonomous robotic surgery—removing the surgeon’s hands—promises enhanced efficacy, safety, and improved access to optimized surgical techniques [12]. Path planning plays a critical role in ensuring the safety and precision of these robotic systems.

For instance, surgical robots require precise path planning to navigate within a patient's body, avoiding vital structures and minimizing tissue damage. Similarly, robots used in drug dispensing and patient assistance must plan collision-free paths in cluttered healthcare environments [13].

The ability of path planning algorithms to adapt to dynamic scenarios and ensure collision avoidance is of utmost importance in healthcare robotics.

1.4.4 Importance of Path Planning in Robotics

Path planning algorithms serve as the backbone of autonomous robotic systems. Their significance lies in their ability to enable robots to:

- Navigate complex and dynamic environments safely.
- Optimize trajectories to achieve task objectives efficiently.
- Adapt to changes in the environment in real time.
- Minimize energy consumption and wear-and-tear on robotic hardware.
- Enhance the overall autonomy and reliability of robotic systems.

As the GA-PRM algorithm combines genetic algorithms with probabilistic roadmap methods, it represents an innovative approach to address the challenges posed by modern robotics applications. The review of path planning algorithms provides essential context for understanding the contributions and advantages of the GA-PRM algorithm in the subsequent sections of this paper.

1.4.5 Prior Research on Simulated vs. Real-World Robotics Performance

Prior research has acknowledged the utility of simulation-based testing for initial algorithm validation while emphasizing the need for real-world evaluations to assess the adaptability and robustness of robotics algorithms. These studies provide valuable lessons on the challenges and opportunities in bridging the gap between simulated and real-world performance.

A comparative study by Acosta et al. [14] has highlighted differences between simulation and real-world results. These discrepancies often arise from differences in sensor accuracy, environmental dynamics, and the adaptability of algorithms to unforeseen scenarios. Such research underscores the importance of addressing these disparities for reliable real-world robotic applications.

This section discusses key studies that have contributed to the understanding of the challenges and insights gained from comparing simulation and real-world evaluations.

1) Simulation-Based Testing:

Simulation-based testing is a common approach for evaluating robotics algorithms due to its cost-effectiveness and safety. Researchers have often leveraged simulators like Gazebo and V-REP to emulate various robotic scenarios. For instance, Mirella Santos Pessoa de Melo et al. [15] explored the use of 3D simulators in validating robotics projects, aiming to make the process more cost-effective. The authors analyzed and compared three popular 3D simulators (V-Rep, Gazebo, and Unity) concerning their usability and features for robotics simulation. They intended to analyze these softwares focusing on robotics simulation, proposing to clarify some questions and guide the simulator choice. Through literature reviews and experiments, they found that V-Rep and Unity offer greater scene editing freedom and user-friendly interfaces, while Gazebo is comparatively more limited.

Ayala et al. [16] conducted a quantitative comparison of Gazebo, Webots, and V-REP, three simulators widely used by the research community to develop robotic systems. The comparison focused on key performance metrics such as CPU usage, memory footprint, and disk access. Through 20 executions of a robotic scenario featuring a NAO robot navigating to a goal while avoiding obstacles, the study found that Webots exhibits the most efficient resource utilization, followed by V-REP, which outperforms Gazebo, particularly in terms of CPU usage. While simulations offer controlled environments, they may not capture the complexities of the real world accurately. Challenges related to sensor fidelity and environmental modeling in simulations. This emphasized the need for realistic sensor models and physics engines to bridge the gap between simulation and reality.

2) Real-World Deployments:

Real-world deployments of robotic systems provide valuable insights into algorithm performance under genuine conditions. For example, Soh Chin Yun, S. Parasuraman, and Velappa Ganapathy [17] used Genetic Algorithm (GA) to assist mobile robot to move, identify the obstacles in the environment, learn the environment and reach the desired goal in an unknown and unrecognized environment. The primary focus is on developing an algorithm capable of avoiding sudden obstacles in the robot's path. In case the robot encounters dynamic obstacles while following its collision-free path, the controller can dynamically re-plan an alternative collision-free route. Fu et al. [18] proposed an improved A* algorithm to overcome inherent drawbacks of the original A* algorithm for the robot path planning. The key improvements included planning a local path between the current and goal nodes before the next search, adopting this path if it is safe, and utilizing post-processing to optimize the resulting path by straightening local segments. Through theoretical analysis and experiments in both virtual and real robot manipulator platforms, the authors demonstrated that their improved A* algorithm achieves a higher success rate in robot path planning while generating shorter and smoother paths compared to the original A* algorithm.

Section 2 of this study discusses the “Related Work”. Section 3 discusses the “Methodology” used in this research. Section 4 overviews the “Experimental Design”, Section 5 introduces the “Simulation Results, Section 6 presents the “Real-World Evaluation”, Section 7 gives the “Comparative Analysis”, and finally Section 8 provides the “Conclusions” of the study.

2. Methodology

2.1 GA-PRM algorithm and its implementation details

The GA-PRM algorithm is an innovative approach to solving robot path planning problems. This algorithm combines the strengths of two powerful techniques: Genetic Algorithms (GAs) and Probabilistic Roadmap (PRM) construction. In the GA-PRM algorithm, the process begins by randomly sampling configurations in the robot's workspace, including the start and goal positions. These sampled configurations form a roadmap that captures the connectivity between different points in the workspace. The GA component then optimizes paths within this roadmap using evolutionary principles. It does so by evolving a population of potential paths through generations, selecting the fittest paths based on a predefined fitness metric that considers both the path's length and collision avoidance. Through this process, the GA gradually refines the paths, eventually converging to a high-quality solution. Additionally, the algorithm incorporates sensor data to navigate around dynamic obstacles effectively, making it suitable for real-world environments.

2.2 Simulation Environment Used

The simulation environment (see Fig. 1) for GA-PRM algorithm validation was carefully designed to simulate real-world robot challenges and enable in-depth performance assessments. It featured a 2D virtual workspace closely resembling real environments with strategically placed static obstacles and designated start and end points. This setup allowed for robust evaluations, simulating scenarios comparable to real-world challenges. Dynamic obstacles, represented as virtual entities following random paths, were introduced to assess the algorithm's adaptability. They emulated moving objects and adhered to realistic speed limits, enabling comprehensive evaluations of obstacle detection and response. The primary challenge was guiding the robot around both static and dynamic obstacles to reach its destination efficiently, similar to real-world navigation.

The simulation also emphasized real-time execution, assessing the algorithm's speed and adaptability to changing scenarios, while rigorous testing covered diverse scenarios, including varying obstacle speeds and environment modifications. Quantitative metrics like path length and computation time were analyzed, offering numerical insights into the algorithm's performance. This well-structured simulation environment facilitated a comprehensive evaluation of the GA-PRM algorithm, providing insights on its strengths and areas for improvement in various real-world scenarios.

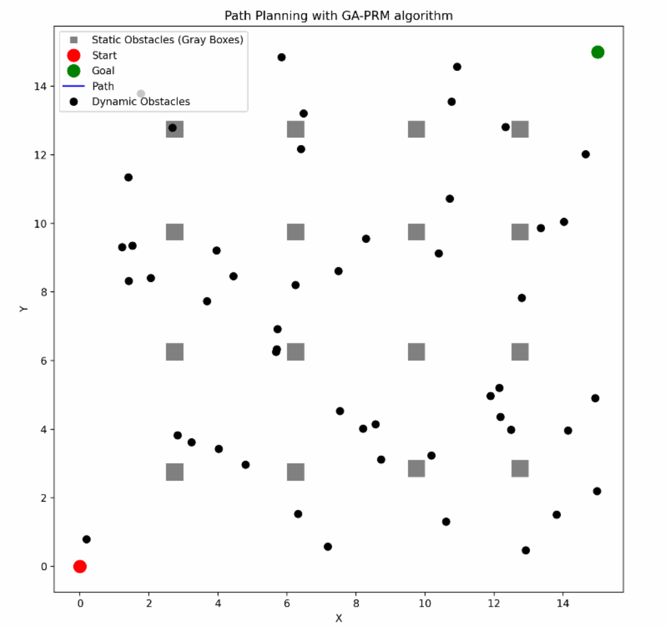


Fig. 1 Simulation environment.

To generate simulations, an interactive approach using matplotlib (a python library) was employed to illustrate the GA-PRM algorithm's path planning. This includes making a series of visuals that demonstrate each step in the path-planning process and generating visualizations viewable in web browsers. The algorithm adapts the path as it encounters dynamic obstacles. The changing path is displayed on a grid representing the workspace. Dynamic obstacles are portrayed as small spheres, and a diamond symbol marks the leading edge of the path. This dynamic presentation helps in understanding how the GA-PRM algorithm adjusts to obstacles while progressing toward its goal.

2.3 Real-World Robotic Platform

The hardware architecture of the robot utilized in this study comprises a carefully designed combination of components, each contributing significantly to its mobility and adaptability. The DC Motor-Driver H-Bridge circuit acts as the main source of power of the robot, enabling precise control over the direction and speed of DC motors. The Arduino Uno microcontroller acts as the brain, coordinating all actions and ensuring stable signals to the motors. Two DC motors with gearboxes propel the robot, while wheels provide the foundation for mobility. A micro servomotor enhances functionality, a swivel wheel enhances maneuverability, and bolts and nuts maintain structural stability. StandOffs maintain component separation and alignment, crucial for structural integrity. Lithium batteries serve as the power source, plates form the framework, and various sensors

(Infra-Red IR, ultrasonic, DHT11) enable obstacle detection and environmental sensing. The main switch facilitates convenient on/off operations, complex connections enable data exchange, and an LCD display serves as a critical interface for real-time feedback. Finally, an Arduino nRF24L01 module enables wireless communication between devices.

In the operational environment (see Fig. 2), the robot navigates a 150 cm by 150cm workspace, tasked with moving from one corner to the diagonally opposite corner while avoiding static and dynamic obstacles. Static obstacles, positioned 22cm from the workspace edges, vary in dimensions and are strategically placed, while dynamic obstacles add unpredictability to the environment.

Precise path planning is imperative for the robot to reach its goal safely without encountering collisions, making this hardware architecture and environment crucial for the study's objectives.

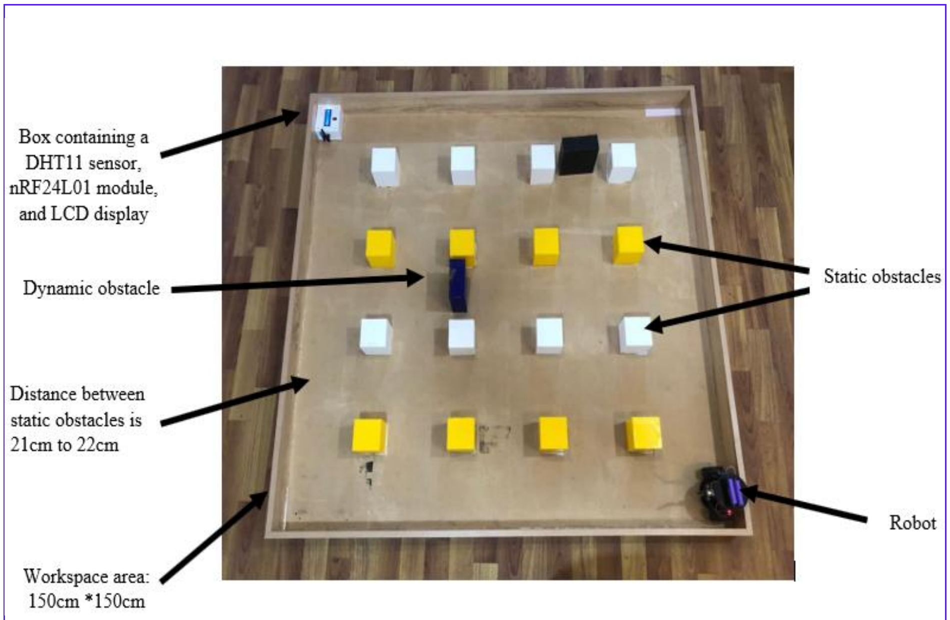


Fig. 2 The robot's operation environment.

2.4 Interaction between the Robot's Hardware and Software Components

The robot uses the GA-PRM algorithm to decide what to do. It looks at data from sensors to figure out the best way to move, considering objects that are moving and things that are not. Two sensors give the robot real-time information about its surroundings: one sees objects, and the other measures how far things are in front of the robot. The GA-PRM algorithm uses this sensor data to make choices about where to go. To make all this work, the "Arduino Uno" microcontroller is important. It takes instructions from the algorithm and turns them into signals for the motors. These signals go to a "DC Motor Driver H Bridge Circuit" that controls how fast the motors go and which way they turn, so the robot can follow paths. Lithium batteries power the Arduino Uno and H Bridge circuit, making the motors work well for moving and responding quickly. There's also a microservo motor for tasks, even though it doesn't help the robot move. The robot's structure, made of nuts, bolts, StandOffs, and body parts, keeps it steady when it moves. This entire works like a team: the algorithm gets information from the sensors to plan the best path. The Arduino Uno and H Bridge turn those plans into motor commands, which make the wheels go and guide the robot along its path. This combination of software and hardware helps the robot avoid obstacles and do what it is supposed to do.

3. Experimental Design

3.1 Selecting Simulation Scenarios and Real-World Test Cases

1) Impact of Different Algorithm Parameters: The criteria for selecting simulation scenarios and real-world test cases in our study were primarily based on a thorough examination of the impact of different algorithm parameters on the GA-PRM algorithm's performance. We aimed to comprehensively understand how specific parameters, namely population size, mutation rate, and the number of generations, influenced the algorithm's behavior. To achieve this, we designed experiments covering a wide range of scenarios to closely investigate the algorithm's response under various conditions.

Firstly, we varied the population size over a range of values, including 50, 100, 200, 300, 400, and 500. Simultaneously, we explored different mutation rates within the range of 0.1 to 0.5 and the number of generations within the range of 100 to 500. These parameter ranges were intentionally selected to encompass a broad portion of the parameter space, ensuring that our study examined a diverse set of scenarios. Additionally, we introduced both static and dynamic obstacles in our experiments to assess how the algorithm performed in varying environmental conditions. The inclusion of dynamic obstacles allowed us to evaluate the algorithm's adaptability to changing challenges over time.

The primary goal of these carefully selected scenarios was to identify patterns, trends, and relationships in the algorithm's behavior across different parameter settings. Conducting these experiments allowed gaining a detailed understanding of how population size, mutation rate, and the number of generations influenced the algorithm's performance. This knowledge was essential for selecting optimal parameter configurations for our different practical scenarios, ultimately contributing to improved path planning and navigation in complex and challenging environments. The experiments were rigorously executed, with 150 runs, each representing an average of 50 iterations of the GA-PRM algorithm, ensuring robust and statistically significant results.

2) Comparison with Other Algorithms: We also conducted an extensive comparison was conducted between the Genetic Algorithm-based Probabilistic Roadmap (GA-PRM) and four other well-known path planning methods: A*, Rapidly exploring Random Tree (RRT), Genetic Algorithm (GA), and Probabilistic Roadmap (PRM) across 50 runs. The primary aim of this comparison was to gain insights into the strengths and weaknesses of these methods concerning three critical performance measures: Average Path Length, Average Computational Time, and Average Smoothness.

Each of these methods employs a unique approach to tackle the complex problem of finding paths in dynamic and cluttered environments. A* utilizes a heuristic-based search approach, RRT focuses on rapid exploration, GA relies on genetic evolution, and PRM employs a probabilistic roadmap model. Subjecting these methods to comprehensive examination in different scenarios allowed for a deeper understanding of their behavior and their ability to discover optimal and feasible paths while considering computational efficiency and path smoothness.

The choice to compare the GA-PRM algorithm with A*, RRT, GA, and PRM was guided by their well-established effectiveness and relevance in the field of motion planning and optimization. For instance, A* is renowned for its efficiency and optimality, especially in grid-based environments, making it a benchmark for path-length optimization scenarios. RRT excels in high-dimensional configuration spaces and dynamic obstacle environments, offering probabilistic completeness and rapid convergence. Genetic algorithms are versatile optimization techniques applied in complex and nonlinear spaces, making them suitable for evaluating the GA-PRM algorithm's global optimization capabilities. Finally, PRM, a classical sampling-based motion planning method, is known for its simplicity and efficiency in constructing roadmaps, serving as a valuable comparison in terms of roadmap generation when assessing the GA-PRM algorithm's performance.

In the comparative analysis with other algorithms, the GA-PRM algorithm underwent the evaluation using a specific set of parameters carefully selected based on insights from 150 experiments. These experiments aimed to identify the optimal configuration for the GA-PRM algorithm, resulting in the selection of parameters that proved most effective: a population size of 200, a mutation rate of 0.3, and a maximum number of generations set at 300.

3) Real-World Test Cases: The criteria employed for selecting real-world test cases to evaluate the GA-PRM algorithm were based on several key considerations aimed at assessing its performance and practicality in healthcare environments. This section examines the rationale behind these criteria and their significance in the evaluation process.

First and foremost, the real-world test cases were designed to closely mimic the complex scenarios encountered in healthcare environments. As depicted in Fig. (3), the physical environment in which the robot operates featured a dynamic landscape with both static and dynamic obstacles. Healthcare facilities often involve crowded spaces with patients, medical equipment, and personnel, making navigation a challenging task for robots. Therefore, the inclusion of scenarios that mirror these complexities allowed us to gauge how well the GA-PRM algorithm can handle such real-world challenges.

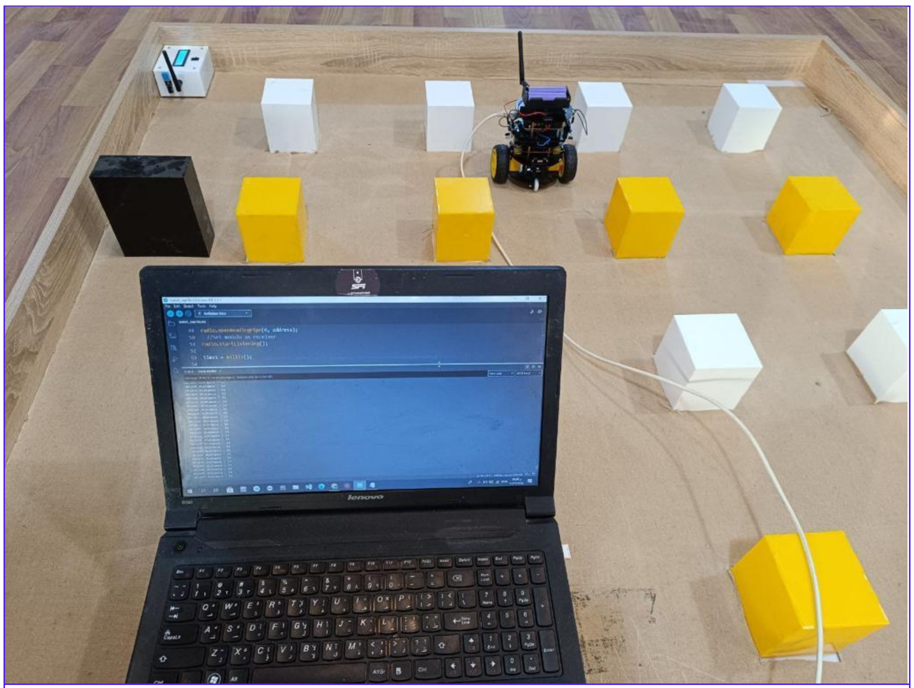


Fig. 3 Robot navigation environment and development workspace.

Another critical factor in the selection of test cases was the evaluation of key performance metrics. The test cases were designed to capture essential metrics such as average path length, computational time, and path smoothness. These metrics provide valuable insights into the algorithm's efficiency and suitability for real-world applications. For instance, the average path length indicates how efficiently the robot can navigate within the environment, a crucial aspect in healthcare settings where time and precision are vital. The computational time measurement helps assess the algorithm's real-time capabilities, which are essential for responsive robot behavior in dynamic environments. Lastly, path smoothness reflects the quality of the robot's movements, an important consideration in healthcare environments to avoid abrupt or jerky motions that could potentially disrupt sensitive operations.

Additionally, the selection of test cases considered environmental variations. Fifty experiments were conducted, involving changes in the positions of dynamic obstacles. This approach allowed us to comprehensively evaluate the GA-PRM algorithm's adaptability to dynamic and evolving conditions, a common occurrence in healthcare facilities. Furthermore, deliberate variations in lighting conditions, including low lighting levels down to 40%, were introduced to assess the algorithm's performance under challenging visibility scenarios. This mimicked real-world situations where lighting conditions may not always be ideal. Importantly, the overarching goal of all these experiments was to ensure that the robot successfully reached its destination and achieved its goals, affirming the algorithm's reliability and effectiveness in practical healthcare applications.

- A carefully designed combination of hardware components takes on roles in ensuring that the robot can move and adapt effectively. Fig. 4 illustrates the robot along with its components. Components in the figure include DC Motor, Arduino Uno microcontroller, DC motors with gearbox, wheels, micro servo motor, swivel wheel, bolts and nuts, standoffs, lithium batteries, plates, sensors, main switch, and connections.

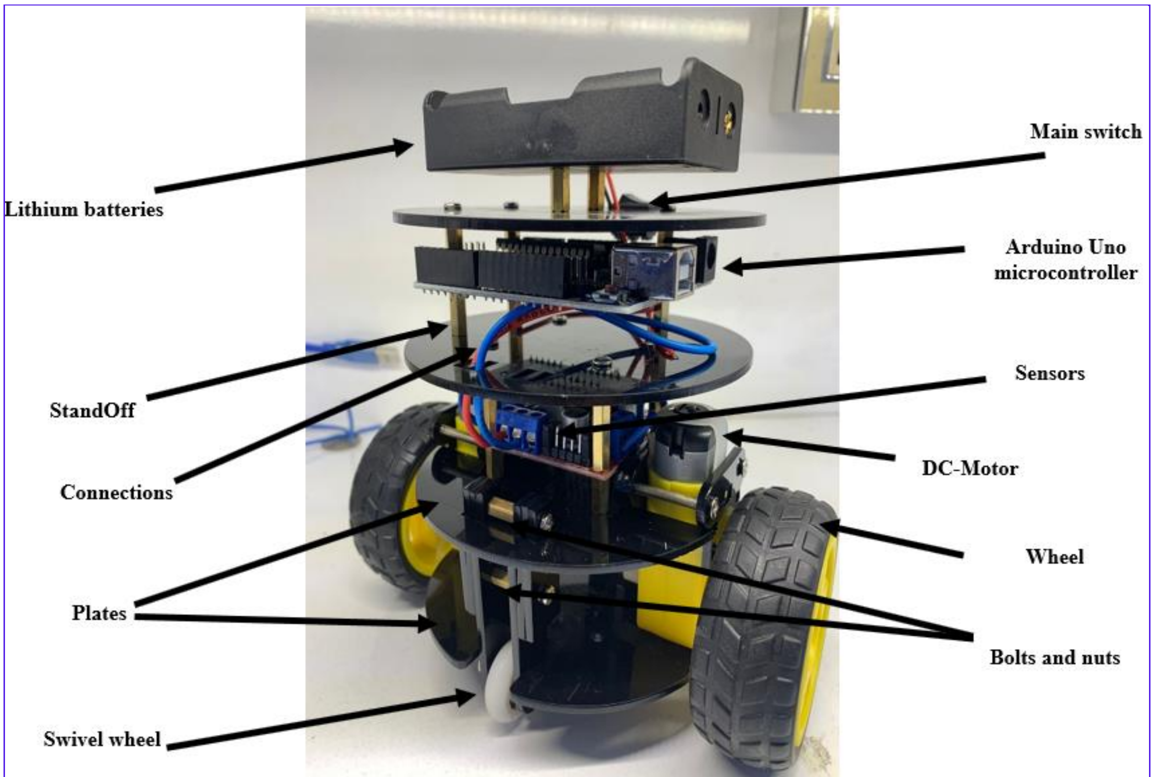


Fig. 4 The robot used in the experiments.

3.2 Evaluation Methods

To comprehensively assess the performance of the GA-PRM algorithm in both simulation and real-world scenarios, several key evaluation metrics were employed. These metrics offer insights into the algorithm's effectiveness, efficiency, and path quality.

1) **Average Path Length (APL)**: Average path length (APL) quantifies the efficiency of the generated paths by measuring the average distance traversed by the robot from the starting point to the goal. Mathematically, APL is defined in equation (1):

$$\text{APL} = \frac{1}{N}\sum_{i=1}^{N} L_i \quad (1)$$

Where:

- N represents the total number of tested scenarios.
- L_i is the length of the path for scenario i.

A lower APL indicates shorter and more efficient paths, reflecting the algorithm's ability to navigate the environment optimally.

2) **Average Computational Time (ACT)**: Average computational time (ACT) measures the algorithm's efficiency in terms of processing speed. It quantifies the time required to plan a path in both simulated and real-world settings. ACT is calculated in equation (2)

$$\text{ACT} = \frac{1}{N}\sum_{i=1}^{N} T_i \quad (2)$$

Where:

- N denotes the total number of tested scenarios.
- T_i represents the time taken to plan a path for scenario *i*.

A lower ACT indicates faster path planning, which is essential for real-time applications.

3) **Average Smoothness (AS)**: Average smoothness (AS) evaluates the quality of generated paths by assessing their continuity and lack of abrupt changes in direction. This metric quantifies the deviation of the path from a straight line. Mathematically, AS is defined as shown in equation (3):

$$\text{AS} = \frac{1}{N}\sum_{i=1}^{N} S_i \quad (3)$$

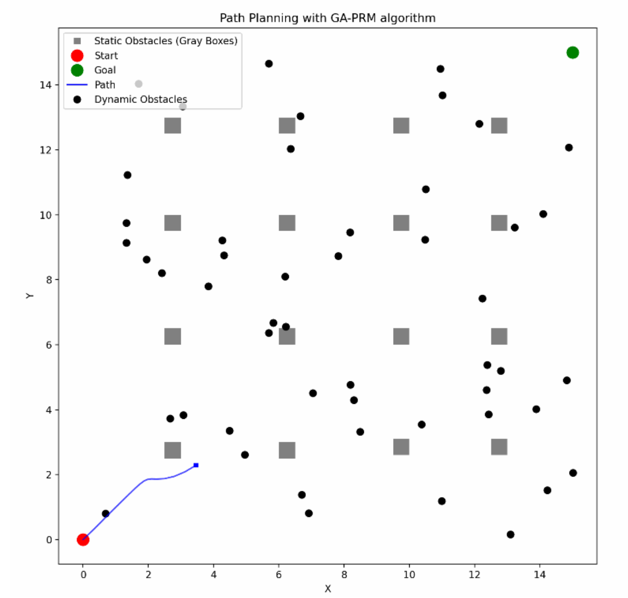
Where:

- N represents the total number of tested scenarios.
- S_i measures the smoothness of the path for scenario ii.

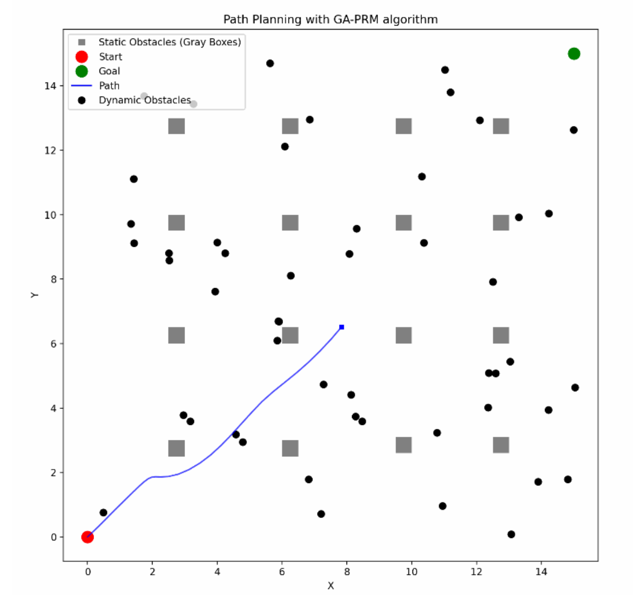
A higher AS value indicates smoother paths, which are preferable for improving robot stability and safety.

4. Simulation Results

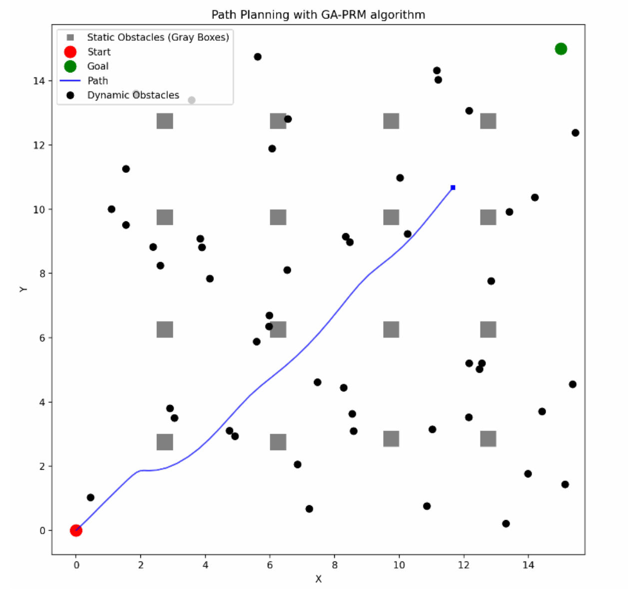
A visual depiction of the robot's motion within the simulated workspace is shown in Fig. 5. The workspace is divided into four sections, each associated with a distinct quadrant. In Subfigure 2-A, the robot's location is presented within quadrant 1, Subfigure 2-B depicts its position in quadrant 2, Subfigure 2-C displays its location in quadrant 3, and Subfigure 2-D demonstrates its placement in quadrant 4. Each segment of the diagram illustrates the path taken by the robot as it navigates through its designated workspace portion. The presence of stationary obstacles is denoted by gray boxes, while small black dots represent moving obstacles.



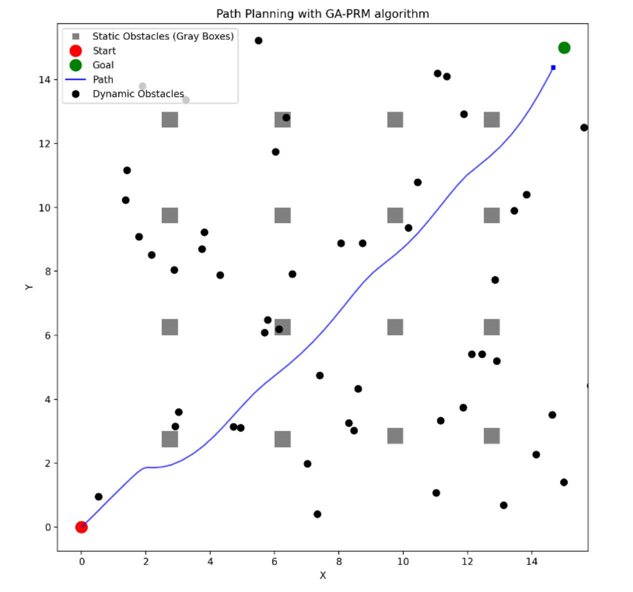
(A) Robot in quadrant 1



(B) Robot in quadrant 2



(C) Robot in quadrant 3



(D) Robot in quadrant 4

Fig. 5 Path planning with GA-PRM algorithm.

4.1 Optimal Parameters Selection

In terms of the optimal parameters to use with the GA-PRM algorithm, Table (1) presents recommendations for achieving an overall balance in the configuration of parameters for the GA-PRM algorithm. To strike a balance between path length and computational time, it is recommended to opt for a moderate population size, typically ranging from 200 to 300, along with an intermediate number of generations, such as 300 to 400. This combination ensures that generated paths are reasonably short without imposing excessive computational demands. Additionally, maintaining a moderate mutation rate, approximately around 0.3, is effective in producing shorter paths while maintaining satisfactory smoothness. For cases where computational efficiency is crucial, smaller population sizes of 100 to 200, lower mutation rates of 0.1 to 0.2, and a limited number of generations between 100 to 200 strikes a balance between rapid solutions and slightly longer paths. When the focus is on path smoothness while maintaining computational efficiency, a moderate population size, combined with advanced post-processing techniques, can enhance smoothness without significantly extending computation time

Table 1: Parameter configuration recommendations for GA-PRM algorithm

Aspect	Recommendation
Population Size	200-300 for balanced path length and computation
Number of Generations	300-400 for thorough exploration without excess time
Mutation Rate	Approximately 0.3 for shorter paths with adequate smoothness
Computational Efficiency	100-200 population size and 0.1-0.2 mutation rate with 100-200 generations for quick solutions
Emphasizing Path Smoothness	Moderate population size with advanced post-processing techniques for enhanced smoothness without excessive computation time

4.2 GA-PRM algorithm Performance Trends

To gain a better understanding of how the GA-PRM algorithm responds when we adjust its parameters, Fig. 6 displays the changes in "Average Path Length," "Average Computational Time," and "Average Smoothness" during 150 iterations. This visual representation provides valuable insights into how the algorithm performs in different situations.

The top section, labeled 'Average Path Length vs. Iteration,' shows how the average path length changes as the experiment progresses through various iterations. This helps us identify trends, such as whether the algorithm is consistently producing shorter paths or if there are fluctuations in path length.

In the middle part, 'Average Computational Time vs. Iteration,' we can observe how the average computational time evolves with each iteration of the experiment. This information gives us a sense of how different parameter settings impact computational efficiency and helps identify any moments where the algorithm's execution time experiences significant changes.

The bottom section, 'Average Smoothness vs. Iteration,' illustrates how the average smoothness of paths changes across different iterations. This graph helps assess the balance between path smoothness and other factors, shedding light on how parameter choices influence the algorithm's ability to generate smooth paths.

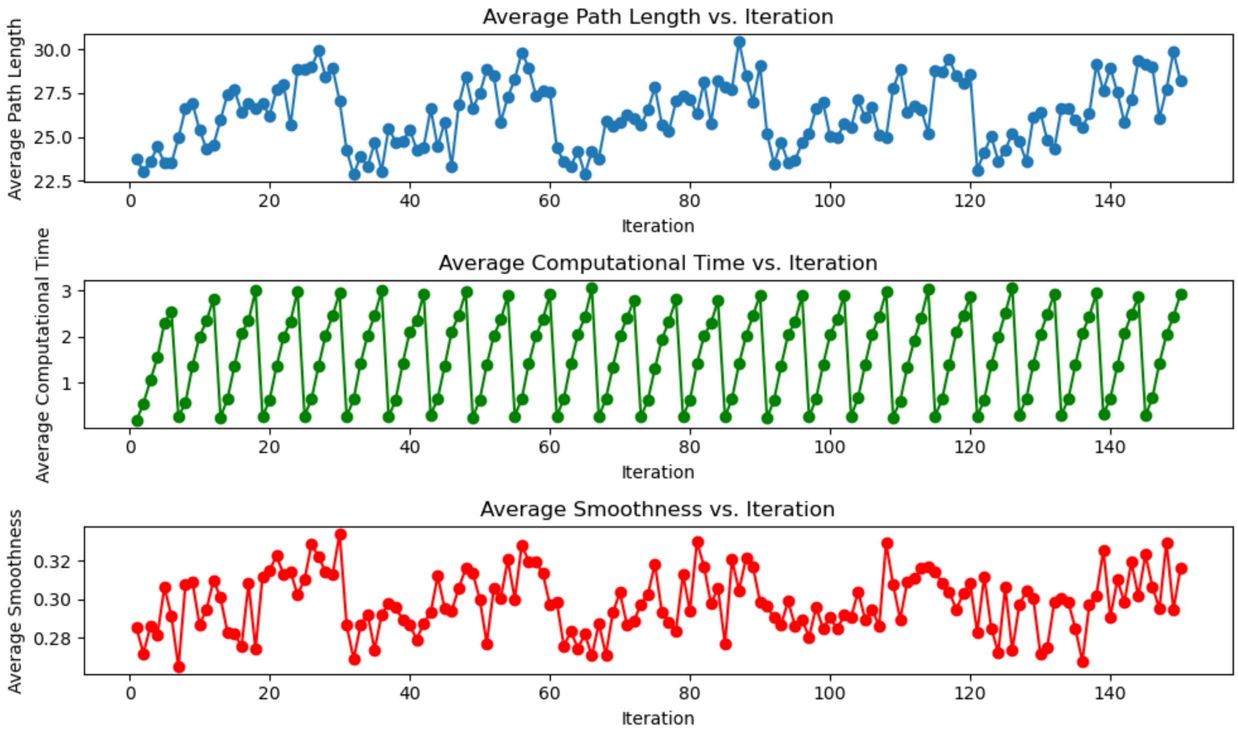


Fig. 6 Performance trends of GA-PRM algorithm over iterations.

4.3 Best and Worst Parameter Configurations

Fig. 7 aims to visually represent how different parameter configurations affect the performance of the GA-PRM algorithm in various evaluation metrics. This helps understand how parameter choices influences the algorithm's performance, with purple and green indicating the best values and blue indicating the worst values.

The upper part of the figure shows the difference between parameter values that lead to the best and worst average path lengths. The best results are achieved when using a combination of a medium-sized population, a low mutation rate, and a high number of generations. In contrast, less favorable outcomes occur when using a larger population, a relatively higher mutation rate, and fewer generations. This highlights the importance of finding a balance between these parameters for the algorithm to generate better paths.

The middle part of the figure focuses on striking a balance between computational speed and performance. The best conditions are observed when using a smaller population, a balanced mutation rate, and a moderate number of generations. Conversely, less ideal scenarios involve using larger populations, higher mutation rates, and fewer generations. This reflects a trade-off where larger populations and more generations improve solution exploration but require more computation time.

In the lower part of the figure, optimal parameter settings involve a well-balanced mix of a medium-sized population, a low mutation rate, and a higher number of generations. Less desirable settings involve larger populations, relatively higher mutation rates, and fewer generations. This emphasizes the importance of achieving smoother paths through a balanced parameter approach, ensuring a harmony between exploration and exploitation during the optimization process.

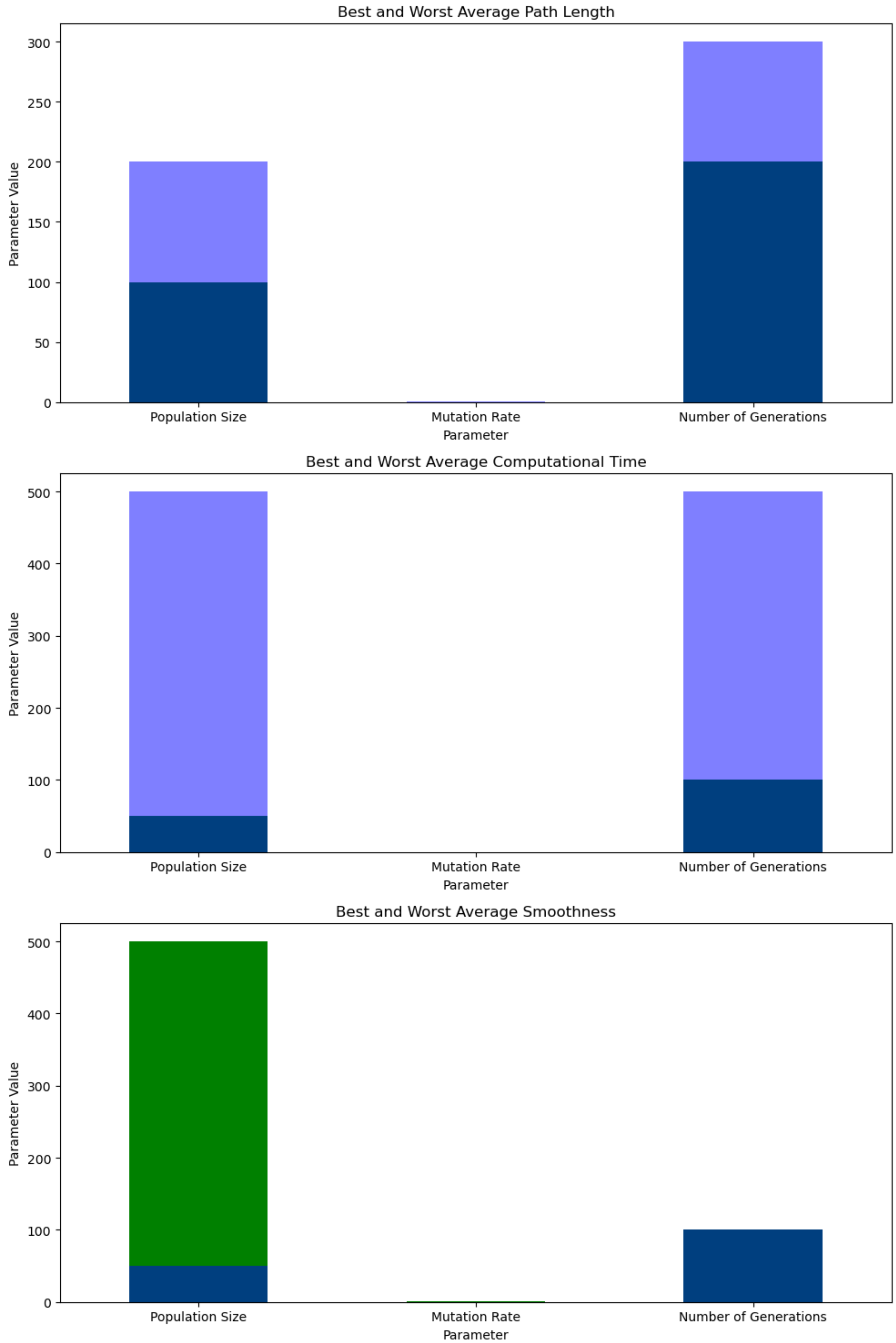


Fig. 7 Comparison of parameter impact on performance metrics.

4.4 Result of Comparison with Other Algorithms

In addition, Table (2) presents a comprehensive comparison of path planning algorithms, including the GA-PRM algorithm, A* algorithm, RRT algorithm, Genetic algorithm, and PRM Algorithm. The evaluation metrics considered are average path length (APL), average computational time (ACT), and average smoothness (AS). These metrics offer insights into the trade-offs between path length, computational efficiency, and path smoothness, providing valuable information for selecting the most suitable algorithm for various applications.

Table 2: Performance comparison between GA-PRM Algorithm and A*, RRT, Genetic, and PRM Algorithms

		Average path length	Average computational time	Average smoothness
1	GA-PRM algorithm	25.6235	0.6881	0.3133
2	A* algorithm	29.1758	0.7452	0.0803
3	RRT algorithm	36.2037	0.6209	0.2911
4	Genetic algorithm	37.43	0.7147	1.5308
5	PRM Algorithm	26.8700	0.9962	1.8543

As illustrated in Fig. 8, GA-PRM algorithm achieves the shortest average path length (APL) among all the algorithms, with an APL of 25.6235 units. This indicates that the current algorithm is successful in finding paths that are, on average, shorter than those generated by the other algorithms.

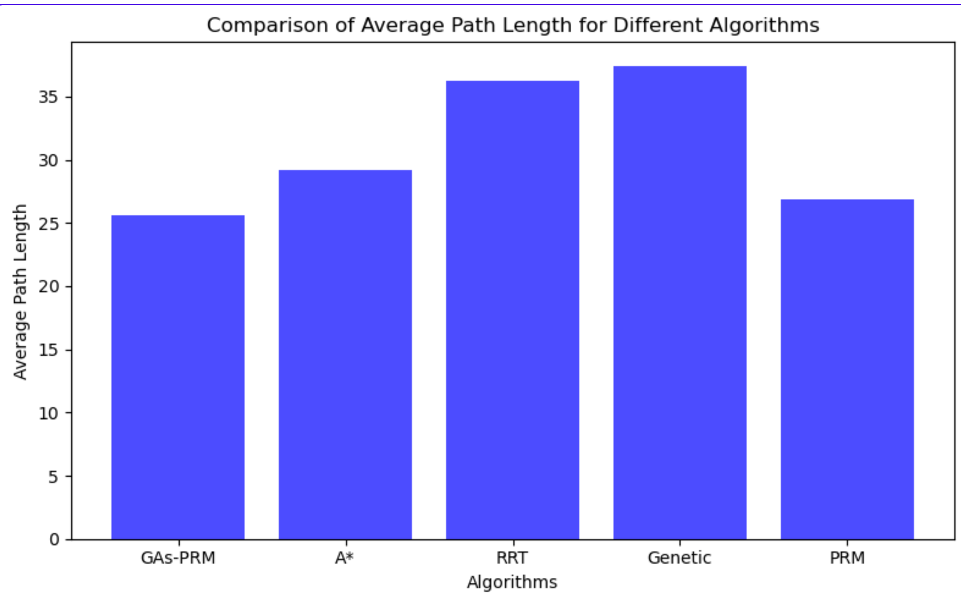


Fig. 8 Comparison of average path length for different algorithm.

While GA-PRM excels in APL, it requires a moderate amount of computational time, with an ACT of 0.6881 seconds. This indicates that it strikes a balance between path length and computational efficiency (see Fig. 9).

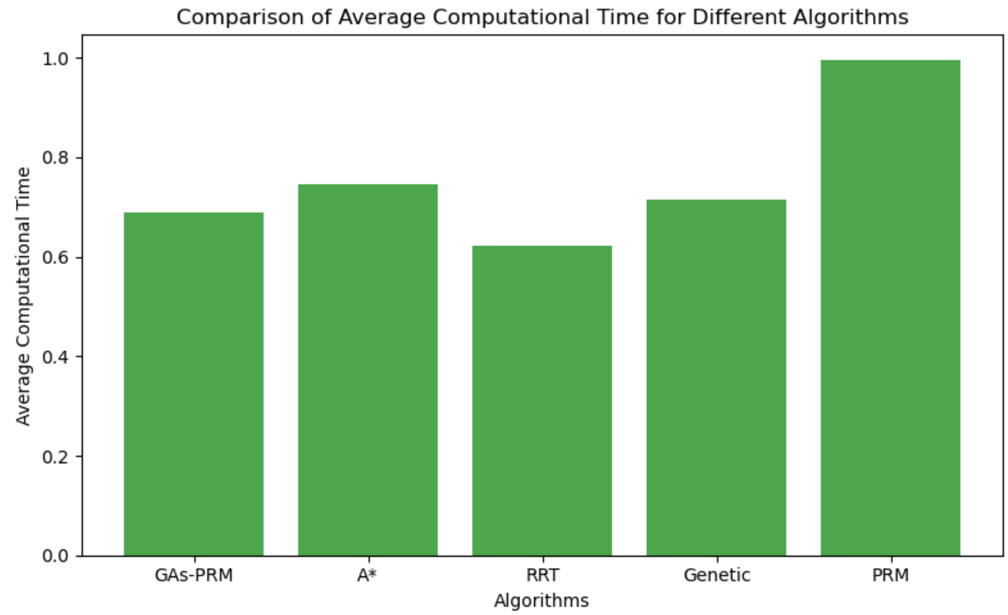


Fig. 9 Comparison of average computational time for different algorithms.

In addition, GA-PRM algorithm produces paths with a relatively high average smoothness (AS of 0.3133). This suggests that it achieves a good balance between path length and smoothness, resulting in paths with less abrupt changes in direction (see Fig. 10).

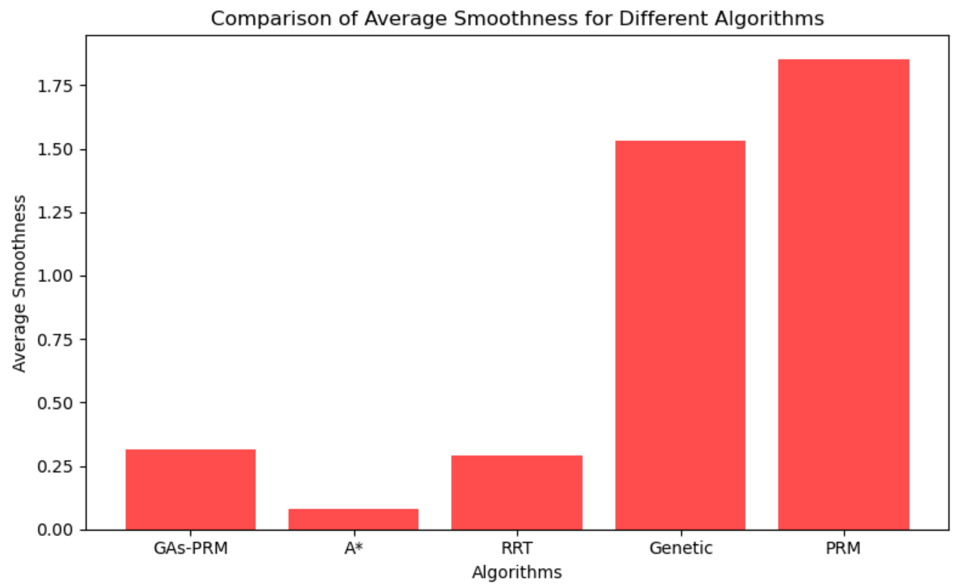


Fig. 10: Comparison of average smoothness for different algorithms.

5. Real-World Evaluation

The implementation of the GA-PRM algorithm on a real robot yielded noteworthy results (see Table 3). The robot achieved an average path length of 21.428 cm, an average computational time of 0.375 seconds, and an average smoothness rating of 0.432. These outcomes carry significant implications, particularly in the context of deploying the robot in healthcare environments. Firstly, the relatively short average path length of 21.428 cm indicates that the robot can navigate hospital corridors and rooms efficiently and accurately. This is advantageous in healthcare settings where space is often limited. A shorter path minimizes the robot's physical presence, reducing the risk of obstructions and making it easier for the robot to move through crowded areas without causing disruptions. Secondly, the average computational time of 0.375 seconds is promising for healthcare tasks that demand timely responses. The robot can rapidly plan its paths and make decisions, a crucial aspect for tasks such as patient transportation, medication delivery, or assisting hospital staff.

Lastly, the average smoothness rating of 0.432 signifies that the robot's movements are fluid and not abrupt. In healthcare environments where patient comfort and safety are paramount, smooth movements are essential. The robot's ability to maintain a high level of path smoothness ensures it can navigate around patients, equipment, and medical staff without causing jarring movements or disturbances.

These results emphasize the effectiveness of the GA-PRM algorithm in optimizing path quality, computational efficiency, and movement smoothness. These attributes are highly desirable when deploying a robot in healthcare settings, where precision, patient care, and operational efficiency are of utmost importance.

Table 3: Implementation results of the GA-PRM algorithm on a real robot

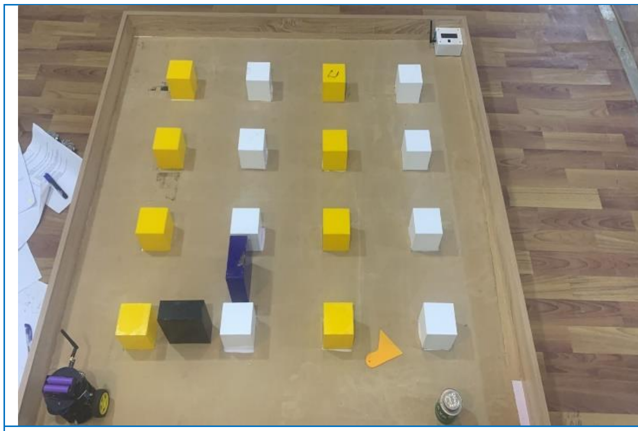
Metric	Results
Average Path Length	21.428 cm
Average computational time	0.375 seconds
Average smoothness	0.432

During the real robot testing of the GA-PRM algorithm, several successful navigation scenarios were observed, showcasing the algorithm's effectiveness in various environments. One notable achievement was the successful negotiation of complex and dynamic obstacle-rich environments. The GA-PRM algorithm demonstrated an impressive capability to adapt and find feasible paths even in scenarios with multiple moving obstacles. This adaptability is crucial for real-world applications, especially in dynamic settings like health care facilities, where robots must navigate safely while avoiding collisions with both static and dynamic obstacles. Additionally, the algorithm displayed robustness in handling changes in the environment's layout. It efficiently adapted to alterations, such as the sudden appearance of new obstacles or modifications to the existing ones, ensuring continuous, obstacle-free motion. These successful navigation scenarios highlight the algorithm's potential for real-world applications where robots must operate autonomously and navigate through ever-changing environments.

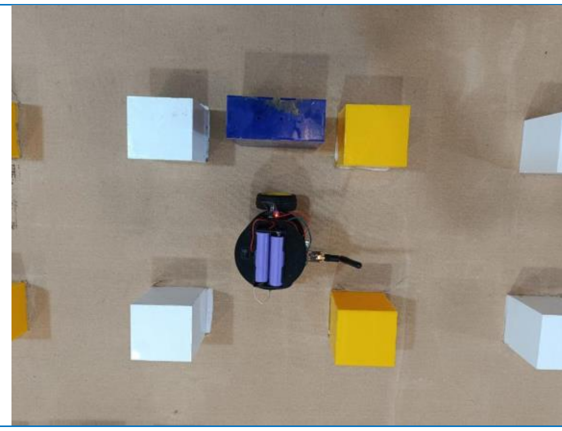
However, during the real robot testing, certain issues were encountered that warrant consideration. One notable challenge was the need for fine-tuning algorithm parameters to achieve optimal performance in specific environments. While the GA-PRM algorithm demonstrated adaptability, achieving the best results often required adjustments to parameters like population size, mutation rate, and the number of generations. This parameter tuning process can be time-consuming and might require domain-specific knowledge, which could limit the algorithm's ease of use for non-experts. Another issue was related to sensor integration and data fusion. Real-world sensors can introduce noise and uncertainties, which, if not properly managed, may lead to suboptimal navigation decisions. Robust sensor calibration and data fusion techniques are essential to enhance the algorithm's performance in real-world scenarios.

Despite these challenges, the successful navigation scenarios and the algorithm's adaptability highlight the GA-PRM algorithm's potential for addressing complex path planning problems in dynamic environments, emphasizing the importance of continued research and development in this area.

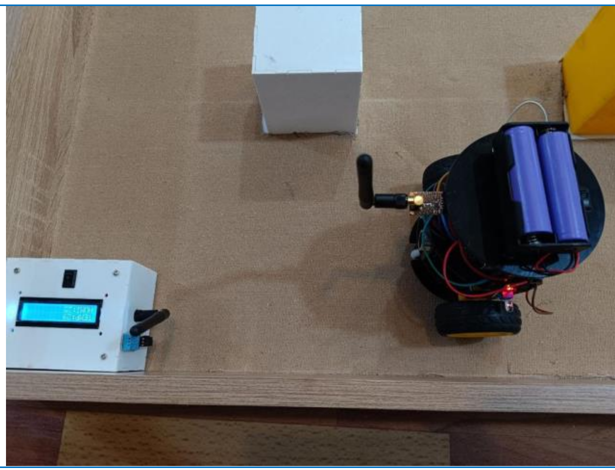
Fig. 11 depicts how the robot moves in various positions within the working space. A humidity and temperature sensor is enclosed inside a white box at the opposite corner of the robot’s location. This sensor records temperature and humidity data and transmits it wirelessly to the robot. When the temperature rises, the robot is prompted to move towards the sensor, demonstrating the objective of the current study of creating a sensor network (WSN).



(A) Robot in start point



(B) Robot moving towards the goal



(C) Robot reaching the goal

(D) Robot moving towards the goal

Fig. 11: Path planning with GA-PRM algorithm

6. Comparative Analysis

6.1 GA-PRM algorithm performance in simulation and real world

In this section, a comprehensive comparison is conducted between the performance of the GA-PRM algorithm in simulation and real-world settings (see Table 4), offering valuable insights into its robustness and effectiveness for practical healthcare applications. This comparative analysis plays a pivotal role in validating the algorithm's suitability in real healthcare scenarios. Firstly, the assessment focuses on the average path length generated by the GA-PRM algorithm. In the simulation environment, the algorithm yields an average path length of 25.6235 units. In stark contrast, when deployed on a physical robot within a real-world setting, the algorithm produces significantly shorter paths, with an average length of 21.428 cm. This indicates the algorithm's adaptability to real-world conditions, emphasizing its capability to generate more concise paths, a crucial factor for efficient navigation in healthcare environments.

Next, consideration is given to the average computational time required for path planning. In the simulation, the GA-PRM algorithm exhibits an average computational time of 0.6881 seconds. However, when executed on an actual robot, this time is notably reduced to 0.375 seconds. This enhanced computational efficiency in real-world execution emphasizes the algorithm's ability to operate swiftly in practical healthcare scenarios, where timely responses are of paramount importance.

Smoothness is another critical aspect, particularly in healthcare robotics, ensuring the safety and comfort of both patients and staff. In the simulation, the GA-PRM algorithm achieves an average smoothness score of 0.3133, indicating relatively smooth paths. Impressively, when tested on a real robot, the algorithm exhibits even higher levels of smoothness, boasting an average score of 0.432. This notable improvement in path smoothness underscores the algorithm's adaptability to real-world constraints, where navigating through tight spaces and avoiding obstacles with precision is imperative for successful healthcare applications. Table 3 provides the comparison between the simulation and robot results

Table 4: Performance comparison between simulation and real-world scenarios

Metric	Simulation	Robot
Average Path Length	25.6235 units	21.428 cm
Average computational time	0.6881 seconds	0.375 seconds
Average smoothness	0.3133	0.432

6.2 Differences between Simulation and Actual Robot Behavior

Comparing the performance of the GA-PRM algorithm in simulations and real-world scenarios reveals significant differences. First, the most noticeable distinction lies in the length of paths. In simulations, the algorithm generates paths with an average length of about 25.62 units, whereas real-world paths tend to be shorter, averaging around 21.43 cm. This difference stems from simulations simplifying the real world, neglecting complexities like physical obstacles, sensor limitations, and the robot's size. These real-world factors affect path planning differently, resulting in shorter routes. Another difference concerns computational time. In simulations, the GA-PRM algorithm requires approximately 0.6881 seconds on average to compute a path, while in the real world, it is notably faster, taking around 0.375 seconds. This variation arises from differences in hardware capabilities. Simulations run on powerful computers with optimized algorithms, enabling them to explore complex solutions, albeit at the cost of more time. In contrast, real-world robots often possess limited computational resources, leading to faster path planning but potentially less optimal paths.

Lastly, the smoothness of paths presents another point of contrast. Real-world robots tend to follow smoother paths, with an average smoothness score of 0.432, in comparison to simulations, which yield an average smoothness score of 0.3133. This unexpected outcome may result from factors such as sensor accuracy and control precision, which influence a robot's ability to execute smooth paths in the real world. Simulations assume perfect information and may prioritize shorter paths even if they involve abrupt changes in direction, while real-world robots adapt to smoother paths.

These differences underscore the importance of considering real-world constraints and hardware limitations when transitioning from simulation to practical robot deployment.

6.3 Advantages and Limitations of Each Testing Approach

Analyzing the performance of the GA-PRM algorithm in both simulation and real-world scenarios provides valuable insights into its strengths and limitations. Simulations offer several advantages. They provide a controlled and reproducible testing environment, allowing for thorough experimentation with different parameter settings. This controlled setting is particularly useful for fine-tuning the algorithm, enabling rapid iterations. Moreover, simulations are cost-effective and safe since they do not involve physical robots or potential risks to equipment and personnel.

However, simulations have their drawbacks. They may not perfectly simulate the complexities of the real world, potentially creating a gap between simulation results and actual robot behavior. Assumptions made during simulation modeling can introduce biases or inaccuracies, such as accurately modeling a physical robot's dynamics or accounting for environmental variations and sensor noise. Thus, while simulations provide valuable initial insights, they may not fully represent real-world performance.

On the other hand, real-world testing offers the advantage of validating the algorithm's performance in its intended environment. This approach considers the inherent complexities and uncertainties of real-world situations, making it the ultimate test of the algorithm's suitability. Real-world testing accounts for factors like sensor limitations, unexpected obstacles, and variations in terrain or lighting, providing a more accurate assessment of the algorithm's robustness.

However, real-world testing comes with its own set of challenges. It requires significant resources, involving the use of physical robots, which can be expensive and time-consuming. Real world testing also presents risks to equipment and personnel, especially in sensitive settings like healthcare. Additionally, conducting systematic experiments with real robots may be limited by factors such as restricted access to test environments or ethical concerns. Table (5) summarizes the advantages and limitations of each testing approach:

Table 5: Advantages and limitations of simulation and real-world scenarios

Testing Approach	Advantages	Limitations
Simulation Testing	- Provides a controlled and safe testing environment. - Allows for rapid and cost-effective testing of algorithms. - Enables testing under various scenarios and conditions.	- May not fully capture real-world complexities and constraints. - Relies on assumptions and simplifications that might not hold in reality. - Limited to the accuracy of the simulation model.
Real-World Testing	- Offers validation in the actual operational environment. - Provides insights into real-world challenges and constraints. - Offers more accurate validation of algorithm behavior.	- Can be time-consuming and expensive. - Poses potential risks to hardware and equipment. - Difficult to create controlled experiments in complex real-world environments.

6.4 Assessment of the Algorithm's Performance in Various Scenarios

In complex and changing environments, the GA-PRM algorithm is adaptable, making it valuable for healthcare robotics. It can adjust its path while dealing with moving obstacles, which is important in places where obstacles change or robots interact with people a lot. One of its strengths is creating smooth paths. This is crucial for safe and comfortable robot movements, especially in healthcare. The algorithm consistently maintains smoothness, both in simulations and when the robot is actually moving, showing it is reliable.

In the real world, the algorithm works well for healthcare. It makes paths that are about 21.43 cm long and plans them in only 0.375 seconds. This means it can efficiently and precisely navigate healthcare environments, finding the shortest and smoothest routes. However, it is important to note that there can be differences between what the simulations predict and what the real robot does. These differences help us make the algorithm even better for the real world. Therefore, it keeps improving and can be even more useful in healthcare to improve patient care.

7. Conclusions

In this study, we have presented the GA-PRM algorithm as an innovative approach to path planning for mobile robots. We have demonstrated its effectiveness through a comprehensive analysis that encompasses both simulation and real-world evaluation. Our findings highlight the algorithm's potential for enhancing path planning efficiency and its adaptability to dynamic environments.

Through simulations, we have shown that the GA-PRM algorithm offers numerous advantages, including a higher search success rate and the generation of shorter, smoother paths when compared to the original A* algorithm. These results underscore the algorithm's ability to optimize path planning in controlled settings.

Real-world testing has further validated the algorithm's capabilities, as it successfully navigated the complexities and uncertainties of actual environments. While real-world testing poses challenges, such as resource requirements and potential risks, it provides crucial insights into the algorithm's robustness and suitability for practical applications.

At the end, the GA-PRM algorithm presents a promising approach for planning paths of mobile robots, offering advantages in both simulated and real-world scenarios. Through bridging the gap between simulation and reality, this research contributes to the advancement of robotics and lays the foundation for practical implementations in various fields, including healthcare, manufacturing, and autonomous transportation. Future work may involve further optimizations and scalability assessments to maximize the algorithm's potential impact on real-world robotic systems.

Acknowledgements

The researcher would like to thank and express the deepest gratitude grateful to all reviewers.

Author Contributions: The authors contributed to all parts of the current study.

Funding: This study received no external funding.

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

References

[1] P. Bajpai and M. Kumar, "Genetic algorithm–an approach to solve global optimization problems," *Indian Journal of computer science and engineering,* vol. 1, no. 3, pp. 199-206, 2010.

[2] A. Keshari and J. Mohanta, "A knowledge based fuzzy-probabilistic roadmap method for mobile robot navigation," *Applied Soft Computing,* vol. 79, pp. 391-409, 2019.

[3] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research,* vol. 30, no. 7, pp. 846-894, 2011.

[4] A. Gasparetto, P. Boscariol, A. Lanzutti, and R. Vidoni, "Path planning and trajectory planning algorithms: A general overview," *Motion and Operation Planning of Robotic Systems: Background and Practical Approaches,* pp. 3-27, 2015.

[5] Y. Nonomura, "Sensor technologies for automobiles and robots," *IEEJ Transactions on Electrical and Electronic Engineering,* vol. 15, no. 7, pp. 984-994, 2020.

[6] J. van Den Berg, J. Snoeyink, M. C. Lin, and D. Manocha, "Centralized path planning for multiple robots: Optimal decoupling into sequential plans," in *Robotics: Science and systems*, 2009, vol. 2, no. 2.5, p. 2.3.

[7] E. W. Dijkstra, "A note on two problems in connexion with graphs," in *Edsger Wybe Dijkstra: His Life, Work, and Legacy*, 2022, pp. 287-290.

[8] B. Li and B. Chen, "An adaptive rapidly-exploring random tree," *IEEE/CAA Journal of Automatica Sinica,* vol. 9, no. 2, pp. 283-294, 2021.

[9] G. Chen, N. Luo, D. Liu, Z. Zhao, and C. Liang, "Path planning for manipulators based on an improved probabilistic roadmap method," *Robotics and Computer-Integrated Manufacturing,* vol. 72, p. 102196, 2021.

[10] S. Karaman and E. Frazzoli, "Incremental sampling-based algorithms for optimal motion planning," *Robotics Science and Systems VI,* vol. 104, no. 2, pp. 267-274, 2010.

[11] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *nature,* vol. 518, no. 7540, pp. 529-533, 2015.

[12] A. Shademan, R. S. Decker, J. D. Opfermann, S. Leonard, A. Krieger, and P. C. Kim, "Supervised autonomous robotic soft tissue surgery," *Science translational medicine*, vol. 8, no. 337, pp. 337ra64-337ra64, 2016.

[13] M. Kyrarini *et al.*, "A survey of robots in healthcare," *Technologies,* vol. 9, no. 1, p. 8, 2021.

[14] B. Acosta, W. Yang, and M. Posa, "Validating robotics simulators on real-world impacts," *IEEE Robotics and Automation Letters,* vol. 7, no. 3, pp. 6471-6478, 2022.

[15] M. S. P. De Melo, J. G. da Silva Neto, P. J. L. Da Silva, J. M. X. N. Teixeira, and V. Teichrieb, "Analysis and comparison of robotics 3d simulators," in *2019 21st Symposium on Virtual and Augmented Reality (SVR)*, 2019: IEEE, pp. 242-251.

[16] A. Ayala, F. Cruz, D. Campos, R. Rubio, B. Fernandes, and R. Dazeley, "A comparison of humanoid robot simulators: A quantitative approach," in *2020 Joint IEEE 10th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, 2020: IEEE, pp. 1-6.

[17] S. C. Yun, S. Parasuraman, and V. Ganapathy, "Dynamic path planning algorithm in mobile robot navigation," in *2011 IEEE Symposium on Industrial Electronics and Applications*, 2011: IEEE, pp. 364-369.

[18] B. Fu *et al.*, "An improved A* algorithm for the industrial robot path planning with high success rate and short length," *Robotics and Autonomous Systems,* vol. 106, pp. 26-37, 2018.