Journal of Robotics

WILEY

RESEARCH ARTICLE OPEN ACCESS

Adaptive Path Planning for Mobile Robots Using a Hybrid PRM–GA Optimization Approach

Thanushika Jathunga[1] | Samantha Rajapaksha[2] | Shehani Jayasinghe[3] | Nuwanthi Abeygunawardena[1]

[1]Department of Electrical and Electronic Engineering, Sri Lanka Institute of Information Technology, Malabe, Sri Lanka | [2]Department of Information Technology, Sri Lanka Institute of Information Technology, Malabe, Sri Lanka | [3]CIET/Department of Electrical and Electronic Engineering, Sri Lanka Institute of Information Technology, Malabe, Sri Lanka

Correspondence: Thanushika Jathunga (tjathunga@gmail.com)

Received: 22 December 2025 | **Revised:** 28 March 2026 | **Accepted:** 10 April 2026

Academic Editor: Yangmin Li

Keywords: genetic algorithm | hybrid algorithm | path planning | probabilistic road map

ABSTRACT

This study addresses the challenge of path planning in mobile robots, that requires efficient navigation in complex environments. Traditional approaches often struggle to meet the increasing demands of modern multi-robot systems operating in dynamic environments. To address these limitations, this study proposes an improved path planning technique by combining the probabilistic roadmap (PRM) with the genetic algorithm (GA), forming a hybrid PRM–GA approach designed to optimize the routes of mobile robots. Experiments were carried out for scenarios involving 2, 3, and 9 robots to analyze the performance of the proposed method under increasing complexity. The proposed PRM–GA method was compared with widely used path planning algorithms including A*, Rapidly exploring random tree (RRT), and conventional PRM. Performance of each method was evaluated focusing on path efficiency and energy consumption. The enhanced fitness function within the GA evaluates robot paths based not only on distance but also on smoothness and turn count, promoting routes with fewer directional changes. The proposed PRM–GA method reduces robot energy consumption while improving navigation efficiency. Experimental results demonstrate that the PRM–GA hybrid method outperforms A*, RRT, and PRM by encouraging smoother paths with fewer turns, thereby enhancing the operational efficiency of multi-robot systems. The effectiveness of the proposed approach highlights its potential for practical applications in sectors where efficient mobile robot navigation is essential.

1 | Introduction

The increasing deployment of mobile robots in various sectors requires the development of robust and efficient path planning algorithms. Path planning is a critical process that involves determining the optimal sequence of movements for a robot to navigate from a starting point to a designated destination. This process must ensure not only efficiency in terms of distance traveled but also the avoidance of collisions with obstacles and other robots in the environment. Traditional path planning methods [1, 2] often face significant challenges, particularly in complex multi-robot scenarios where the dynamics of navigation become increasingly intricate.

Traditional path planning algorithms often struggle to address the complexities inherent in multi-robot scenarios, particularly when it comes to identifying efficient and energy-saving solutions. Although genetic algorithms (GAs) present a promising approach to path planning, existing implementations frequently emphasize finding the shortest possible path without adequately considering the number of turns a robot must execute along that route. This singular focus on minimizing distance can result in several significant issues that compromise the overall effectiveness of the robotic system. One major consequence is the selection of suboptimal paths. Paths that are shorter in distance but require many turns can be less efficient than slightly longer paths that feature fewer turns. Frequent changes in direction require an increase in energy expenditure as robots must continuously accelerate and decelerate to navigate these turns. This maneuvering can make paths that initially appear shorter suboptimal in terms of overall energy usage and time, ultimately affecting the robot's performance and effectiveness [3]. In addition, the increased energy consumption associated with the navigation of paths laden with numerous turns poses another critical challenge. Robots that follow such routes expend more energy due to the constant need for acceleration and deceleration during directional changes. This increased energy consumption can significantly reduce the operational efficiency of the multi-robot system, leading to shorter operational lifespans and increased maintenance needs [4]. As a result, the inability of traditional path planning algorithms to balance distance and maneuverability not only impacts individual robots but also diminishes the overall effectiveness of multi-robot coordination and collaboration [3, 4]. Unlike traditional approaches, our method aims to address this issue by optimizing not just for distance, but for smoothness and fewer directional changes, thus improving energy efficiency. In this study, optimization algorithms are employed to overcome the limitations of traditional path planning methods, which often struggle to efficiently handle dynamic, high-dimensional, and multi-objective environments. Such algorithms enable the balancing of several critical objectives such as minimizing path length, energy consumption, and turn count while maintaining smooth and collision-free navigation. Among the various optimization techniques, the GA was selected due to its proven robustness, flexibility, and adaptability in diverse optimization tasks. GA's population-based evolutionary mechanism allows for efficient exploration and exploitation of the solution space, reducing the likelihood of getting trapped in local optima. Furthermore, GA can be easily hybridized with sampling-based planners like the probabilistic roadmap (PRM), combining PRM's efficient global exploration with GA's powerful optimization capability to enhance convergence and solution quality.

Although newer metaheuristic algorithms such as Coronavirus Herd Immunity Optimizer (CHIO) [5], JAYA [6], and other hybrid swarm-based methods [7, 8] have recently emerged, they often require extensive parameter tuning, have limited interpretability, or are designed for domain-specific problems such as vehicle routing or network optimization rather than robot navigation. In contrast, GA offers a well-established theoretical foundation, reliable performance, and ease of customization for multi-robot and real-time path planning. Therefore, the proposed PRM–GA hybrid method was developed to leverage GA's strengths while addressing energy efficiency, path smoothness, and adaptability in dynamic multi-robot environments.

1.1 | Preliminaries

Table 1 lists the acronyms used in the paper.

One of the primary challenges in path planning is computational complexity, as algorithms must evaluate numerous potential paths to identify the most effective one. As the number of robots in a given environment increase, the potential paths and collision scenarios grow exponentially. This exponential increase leads to a substantial increase in computation demands, making traditional algorithms less efficient and more costly to operate. Consequently, coordinating multiple robots to navigate effectively while simultaneously avoiding collisions requires advanced path planning strategies that can manage the intricacies of multiple agents operating in the same space [9]. Furthermore, conventional algorithms may struggle with the issue of local optima, which occurs when they identify a path that appears to be optimal based on local information, but does not represent the best overall solution when considering the movements of all robots in the system. This limitation can hinder the overall performance of the robotic system and prevent the achievement of truly optimal paths for all agents involved [10]. The challenges posed by computational complexity and local optima highlight the urgent need for innovative solutions that can efficiently address the complexities of multi-robot path planning while ensuring optimal navigation and effective collision avoidance. As mobile robotics continues to expand into new applications and environments, developing sophisticated algorithms that can adapt to these challenges will be essential for enhancing operational efficiency and effectiveness [10, 11]. While many existing algorithms struggle with this issue, our PRM–GA hybrid addresses it by ensuring that the search for optimal paths is adaptable to dynamic environments.

Addressing these challenges is essential to enhance the performance and sustainability of robotic systems in diverse applications. Hence, this study introduces a hybrid PRM–GA method enhanced with collision avoidance techniques, specifically tailored for mobile robot path planning. By addressing the limitations of traditional path planning algorithms, this approach aims to optimize path efficiency, reduce energy consumption, and safely navigate complex environments.

1.2 | Paper Organization

Section 2 of the paper discusses related work that has been conducted on existing path planning techniques followed by Section 3

TABLE 1 | List of acronyms.

Acronym	Description
AGV	Automatic guided vehicles
CHIO	Coronavirus herd immunity optimizer
GA	Genetic algorithms
IoV	Internet of vehicles
JAYACIoV	JAYA clustering algorithm for IoV
PRM–GA	Probabilistic Roadmap–Genetic algorithm
PSO	Particle swarm optimization
PSO–SA	Swarm optimization–simulated annealing
RRT	Rapidly-exploring random trees
SA	Simulated annealing

explains the Problem formulation and objective function. Methodology is described in Section 4 where it explains the theories behind each step followed. Section 5 provides the simulations results for the proposed PRM–GA technique, and nine test cases are used. Further the statistical performance analysis results of the proposed PRM–GA approach is also mentioned under Section 5. Discussion and future work is carried out under Section 5. Finally, a conclusion of the paper is in Section 7, by summarizing the key findings and contributions of this study.

2 | Related Work

Several recent articles provide a comprehensive overview of existing path planning techniques, highlighting emerging trends and key performance metrics such as path smoothness, real-time applicability, and energy efficiency.

Although they are frequently used for global path planning, graph-based techniques like Dijkstra [12] and A* [13, 14] encounter significant challenges. Due to extensive grid searches, Dijkstra's algorithm has a high computational complexity even though it ensures the shortest path [1, 15]. A* is unsuitable for real-world robot kinematics since it increases efficiency with heuristic guidance but still generates nonsmooth routes with sharp turns [16–18]. D* struggles with frequent recalculations in large-scale maps [19], although it extends A* for dynamic situations. The main drawback of these approaches is their dependance on discrete grid representations. This results in less flexibility in dynamic environments.

As a solution, heuristic and meta-heuristic algorithms have gained popularity. Unlike problem specific classical approaches, meta-heuristics are adaptable optimization tools that are often modeled using natural foraging behaviors or evolutionary processes. Numerous domains have demonstrated their advantages [20–25]. Because they offer greater flexibility and a higher ability to escape local optima when the objective function is properly defined, these algorithms are useful tools for robot path planning problems.

Particle swarm optimization (PSO) [7, 8], GAs [26, 27], simulated annealing (SA) [28], ant colony optimization (ACO) [29, 30], artificial bee colony (ABC) [31–33], and immune plasma algorithm (IPA) [34] are among the most popular meta-heuristic methods. Each has been proven to be successful in certain areas of path planning. For instance, ABC algorithm [35] has been extensively researched due to its low number of control parameters and great exploration capabilities, which makes it useful for path planning for both single and multiple robots [36–38]. Variants and hybrids of ABC have been proposed to enhance performance, such as integrating coevolution frameworks [33], combining ABC with evolutionary programing [32], or improving search equations with physical models [39]. These studies demonstrate how meta-heuristics can be applied to variety of path planning issues. Candidate pathways are evolved by GAs via crossover, mutation, and selection. However, the fixed-length chromosomes used in traditional GA frequently causes sluggish convergence and inefficient encoding. Variable-length chromosomes [40–42]. Bezier curves for smoothness, and clearance-based operators to avoid obstacles [43] are incorporated into enhanced GA variants. In the absence of specified constraints, co-evolutionary GA [44] is unable to prevent collisions while planning pathways for numerous robots. The main drawbacks of GA are its reliance on the starting population's quality and its propensity to become stuck in local optima. In dynamic or multi-robot situations, algorithms like GA and PSO may show inefficiency and slow convergence. In congested environments, PSO struggles with local optima but converges more quickly than GA [45]. Global exploration (PSO) and local refinement (SA) are combined in Hybrid PSO–SA, which shows promise for automatic guided vehicles (AGV) navigation but is not scalable for multi-robot systems [46–49]. Similar to this, even though ABC and its variations are strong, they frequently put lowering path length ahead of other important goals like path smoothness, energy economy, or real-time flexibility.

Sampling-based methods, such as PRM and rapidly exploring random trees (RRT), excel in high-dimensional spaces [50]. PRM constructs a network of collision-free paths via random sampling, making it suitable for complex environments [51]. However, it lacks optimization for smoothness, energy efficiency, or multi-robot coordination. RRT* optimizes paths incrementally but is computationally expensive for real-time applications [52, 53]. The main gap in these methods is that while PRM efficiently generates feasible paths, it does not naturally optimize for multi-objective criteria such as energy consumption, smoothness, or dynamic multi-robot scenarios.

Hybrid approaches have been introduced to overcome individual limitations of these methods. PRM–GA has been applied in medical robotics, combining PRM's exploration with GA's optimization, but it focuses only on single-robot scenarios [54, 55]. GA-Bezier Fusion optimizes control points for smoother paths but ignores dynamic obstacles [56]. The proposed hybrid approach of GA–PRM addresses the shortcomings of previous approaches by integrating PRM-guided GA optimization for collision avoidance, energy efficiency, and real-time adaptability.

While prior work demonstrates the strengths of PRM in scalability and GA in optimization, limited research exists for multi-robot coordination with collision-avoidance and dynamic environment adaptability. In [57], PRM–GA approach is proposed by incorporating adaptive mutation for obstacle clearance and multi-robot collision avoidance and the study focuses on real-time path optimization for multi-robot systems, handling dynamic obstacles while reducing path lengths and turn counts. However, in highly dynamic environments, this method sometimes produces suboptimal paths due to the need for frequent re-optimization.

Hybrid Data-Driven Active Disturbance Rejection Sliding Mode Control (ADRC–SMC) has been successfully applied to tower crane systems and shows the potential for improving system performance through optimization [58]. Another meta-heuristic approach, Enhanced P-type Control, utilizes indirect adaptive learning from set-point updates, demonstrating the flexibility of optimization methods in dynamic environments [59]. While these approaches are typically focused on control systems, their methodologies share common principles with path planning, particularly in dynamic environments where real-time optimization is crucial.

In a similar manner, the CHIO has been applied to the capacitated vehicle routing problem (CVRP), demonstrating how a

TABLE 2 | Comparative summary of path planning methods.

Research/reference	Technique/algorithm	Strengths	Weaknesses
[1, 15]	Dijkstra's algorithm	Guarantees shortest path	High computational complexity, inefficient in dynamic environments
[16–18]	A^*	Efficient with heuristic guidance	Non-smooth paths, impractical for real-world robot kinematics
[19]	D^*	Handles dynamic environments	Frequent recalculations, large-scale maps
[7, 8]	PSO	Faster convergence than GA	Local optima in cluttered environments
[26, 27]	GA	Strong optimization ability	Slow convergence, local optima, fixed-length chromosomes
[40, 41, 43, 44]	Enhanced GA variants	Variable-length encoding, Bezier curves, multi-robot planning	Collisions not always avoided, initial population dependent
[28]	SA	Local refinement, powerful exploration	Limited performance in dynamic environments
[29, 30]	ACO	Effective global optimization	Slow convergence, sensitive to tuning
[31–33]	ABC	Strong exploration, simple parameters	May ignore smoothness/energy factors
[46–49]	PSO–SA Hybrid	Combines exploration + refinement	Poor scalability for multi-robot systems
[51]	PRM	Scalable, handles high-dimensional maps	No energy or smoothness optimization
[52, 53]	RRT	Efficient sampling in cluttered spaces	Expensive for real-time use
[54, 55]	PRM–GA (Medical)	Combines PRM + GA optimization	Only tested for single-robot cases
[56]	GA–Be´zier Fusion	Smooth curved paths	No dynamic obstacle handling
[57]	Adaptive PRM–GA	Real-time multi-robot optimization	Suboptimal in highly dynamic scenes
[58]	ADRC–SMC Hybrid	Improved control in cranes	Not intended for navigation
[59]	Enhanced P-Control	Good for dynamic control	Not suitable for robot path planning
[5]	CHIO	Metaheuristic for routing	Vehicle-focused, not robot kinematics
[6]	JAYA-IoV clustering	Optimized routing in IoV	No multi-objective navigation ability

metaheuristic optimization algorithm can solve complex routing problems with various constraints [5]. CHIO's modification for maintaining solution feasibility and optimizing routes in dynamic conditions offers valuable insights that could be translated to mobile robot path planning, especially in complex environments. However, while CHIO addresses optimization in dynamic settings, its primary focus on vehicle routing may not directly apply to the multi-dimensional challenges faced in robot path planning, especially with the inclusion of energy consumption and maneuverability considerations in mobile robots.

Additionally, advancements in the Internet of Vehicles (IoV) have led to the development of the JAYA Clustering Algorithm for IoV (JAYACIoV), a hybrid metaheuristic approach combining JAYA with K-means and Automatic Clustering to optimize communication pathways in dynamic networks [6]. While the algorithm demonstrates significant improvements in routing performance under dynamic traffic conditions, its application in vehicular networks does not directly translate to the path planning needs of mobile robots, where factors such as energy consumption, smoothness, and real-time adaptability play crucial roles. Moreover, while JAY-ACIoV optimizes communication routes, it lacks the multi-objective optimization required for efficient robot navigation, where both energy efficiency and minimal path length must be considered in tandem.

Table 2 presents the summary of the literature. To mitigate the shortcomings of past research, our research introduces an improved path-planning method by integrating the PRM with the GA, forming the PRM–GA hybrid. The novelty of the proposed PRM–GA hybrid method lies in its integration of PRM for efficient path exploration with the optimization capabilities of GAs. While traditional path planning methods, including Dijkstra's and A^*, are known for their ability to guarantee the shortest paths, they often struggle with computational complexity and smoothness. Sampling-based methods like PRM and RRT excel in high-dimensional spaces but lack optimization for energy efficiency and smoothness. Similarly, bio-inspired methods such as GA and PSO face challenges like slow convergence and inefficiency, especially in dynamic, multi-robot environments. Our PRM–GA hybrid addresses these gaps by incorporating a multi-objective fitness function that not only minimizes distance but also optimizes for path smoothness and turn count, ensuring improved energy efficiency and operational performance. This approach fills a critical gap that many existing algorithms fail to address, offering significant improvements in path planning by focusing on real-time dynamic adaptability and multi-robot coordination. By providing a more robust solution, the PRM–GA hybrid is positioned to advance applications in logistics, industrial automation, and autonomous robotics.

The key contributions of this paper are as follows:

- A hybrid approach combining PRM and GA for optimizing mobile robot path planning. The integration of PRM's mapping capabilities with GA's optimization strengths for improved path efficiency, smoothness, and energy consumption.
- An enhanced fitness function incorporating path distance, turn count, and smoothness, ensuring energy efficiency and minimizing unnecessary directional changes.
- Collision avoidance techniques integrated into the hybrid PRM–GA approach for safe and efficient navigation in dynamic multi-robot scenarios.
- A simulation framework for testing and validating the PRM–GA method, ensuring its suitability for real-time, dynamic path planning scenarios.

3 | Problem Formulation and Objective Function

The primary objective of this study is to achieve efficient and smooth path planning for multiple robots navigating within a shared environment. Each robot must travel from its start point to a goal point while minimizing overall energy consumption and avoiding collisions.

The optimization problem is formulated to minimize the total energy cost, which is mainly composed of distance and turning energy components derived from the fitness function used in the GA. The objective function is expressed as:

$$\min J = E_{\text{distance}} + E_{\text{turn}}, \tag{1}$$

where E_{distance} represents the energy associated with the total travel distance and E_{turn} represents the additional energy consumed due to direction changes along the path.

To ensure safe navigation, a collision avoidance mechanism is integrated as a geometric constraint. Specifically, the Euclidean distance between any two robots i and j at any time t must satisfy:

$$\| p_i(t) - p_j(t) \| \geq d_{\text{safe}}, \ \forall i \neq j, \ \forall t, \tag{2}$$

where $\mathbf{p}_i(t)$ and $\mathbf{p}_j(t)$ are the positions of robots i and j at time t, and d_{safe} is the minimum allowable safety distance. While this constraint does not directly contribute to the objective function, it ensures collision-free paths for all robots.

This problem formulation establishes the foundation for the proposed PRM–GA-based path planning approach discussed in the following Methodology section.

4 | Methodology

The methodology behind the GA used in this multi-robot pathfinding simulation follows a structured, phase-based approach aimed at achieving efficient and collision-free navigation in a grid environment with fixed obstacles.

4.1 | Environmental Modeling

The simulation's environmental model was based on a binary occupancy grid consisting of a 10×10 unit layout, representing a typical industrial indoor environment. Automated warehouses factory floors are one such example where a similar layout as such is present. In this configuration, each cell corresponds to a specific location in the grid, enabling clear differentiation between free and blocked areas. Obstacles were placed at designated coordinates within the grid, where these positions were marked with a binary value of 1 to denote non-navigable regions. All remaining cells were set to 0, indicating available paths that the robots could traverse.

To efficiently assign obstacle positions within the grid, their coordinates were converted into linear indices, precisely identifying the occupied cells. This indexing approach enabled accurate mapping of each obstacle in the structured workspace, improving spatial representation by defining limits and barriers. The chosen modeling method highlighted the spatial layout and concentration of obstacles, offering the robots a well-defined and practical simulation space for navigation and obstacle avoidance. This setup served as a basis for refining robot movements in the grid. Figure 1 depicts the simulated area, showing the fixed obstacles within the environment.

4.2 | Population Initialization

The initial path planning used the PRM technique. The PRM method utilized 1000 randomly sampled nodes, connecting pairs within a 1 unit range if collision free, to generate feasible paths. To validate this fixed node count and avoid overfitting, a sensitivity analysis was conducted comparing sampling

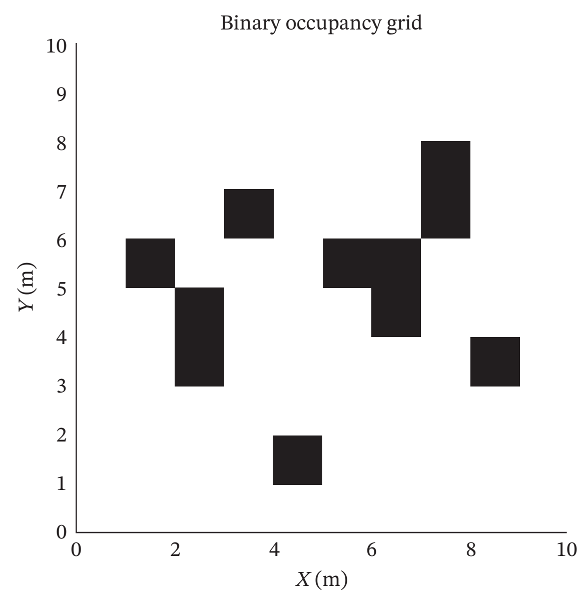

FIGURE 1 | Simulation environment.

densities of 500, 750, 1000, and 1250 nodes. The analysis demonstrated that 1000 nodes offered an optimal trade off between path success rate and computational cost, with negligible improvements observed.

beyond this threshold. This justifies the selection of 1000 nodes as a balanced parameter for robust path planning. While these PRM generated paths avoided obstacles, they did not always reduce length or turns.

1. Chromosome Representation and Validity: In the proposed GA, each chromosome encodes a complete robot path as a sequence of waypoints:

$$\text{Chromosome} = [\text{P1}, \text{P2}, \ldots, \text{Pn},$$

where P_i represents the ith waypoint in the grid. This structure ensures that the path is continuous from start to goal, and each waypoint corresponds to a valid, unoccupied grid cell.

During crossover and mutation, constraints are applied to maintain path feasibility. For crossover, only internal segments are exchanged while preserving start and goal points. Any offspring path is checked to avoid obstacles and maintain connectivity. For mutation, small adjustments are applied to waypoints, but each mutation is accepted only if it remains within grid boundaries, avoids obstacles, and maintains a continuous path. Invalid mutations are discarded or retried.

This representation and constraint mechanism ensure that all post-operation chromosomes remain navigable and collision free, supporting reliable multi-robot path planning.

4.3 | Fitness Evaluation

The overall fitness for a given path is calculated as the inverse of the total energy consumed, to encourage paths that minimize energy usage. The formula used is shown by Equation (3).

$$\text{Fitness Function} = \frac{1}{\text{Total energy consumption}}. \tag{3}$$

The energy consumed along the path depends on the distance traveled and the sharpness of the turns between consecutive waypoints. The energy consumption for traveling between two consecutive waypoints is proportional to the distance assuming friction or resistance per unit of distance is constant. When calculating the distance, Euclidean distance was used as it provides the shortest distance between the two points. Other methods of calculating distance such as Chebyshev and Manhattan are suited for grid or axis-based motions but not for mobile robot motions [57]. Additionally, energy is added for sharp turns, where sharpness is determined by the angle between consecutive path segments. Equal weighting factors were used for path distance and turn energy in the study. This is to ensure a fair trade-off between minimizing travel distance and reducing sharp directional changes.

1. Travel Energy Calculation: The energy for traveling between consecutive points is calculated using Equation (4) shown below.

$$\text{Travel Energy} = \text{Distance} \times \text{Energy Per Unit Distance}. \tag{4}$$

The total distance traveled is computed by summing the Euclidean distances between consecutive way-points. The Euclidean distance between two points (x_1,y_1) and (x_2,y_2) is given in Equation (5).

$$\begin{aligned}&\text{Distance Between Two Consecutive Points}\\ &= \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}.\end{aligned} \tag{5}$$

The total distance traveled can be computed using Equation (6) where i is the index and n is the last value of index.

$$\text{Total Distance} = \sum_{i=0}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}. \tag{6}$$

2. Turn Energy Calculation: The turning angle θ represents the change in orientation between two consecutive motion vectors along the robot's path. For each turn, the energy consumption is determined by this angle θ between the two segments. Thus, the turn energy is modeled as being directly proportional to the magnitude of the turning angle θ, as shown in Equation (7).

$$\text{Turn Energy} = |\theta| \times \text{Turn Energy Per Degree}. \tag{7}$$

The turning angle θ is calculated using the geometric relationship between consecutive path vectors. For three consecutive points (P_{i-2},P_{i-1},P_i), the angle is formed between the vectors $P_{i-2} \to P_{i-1}$ and $P_{i-1} \to P_{i.}$ The dot product of these vectors gives the cosine of the included angle, which can be expressed as in Equations (8) and (9).

$$D = \sqrt{(x_{i-1} - x_{i-2})^2 + (y_{i-1} - y_{i-2})^2} . \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}, \tag{8}$$

$$\theta = \cos^{-1} \frac{(x_{i-1} - x_{i-2})(x_i - x_{i-1}) + (y_{i-1} - y_{i-2})(y_i - y_{i-1})}{D}. \tag{9}$$

This formulation ensures that both direction and magnitude of angular change are considered, allowing accurate estimation of the rotational energy component.

3. Total Energy Consumption: The total energy required for the path is the sum of the travel energy and the turn energy and can be obtained by using Equation (10).

$$\text{Total Energy Consumption} = \text{Travel Energy} + \text{Turn Energy}. \tag{10}$$

In this study, the total energy is quantified using a simulation-based model that evaluates both translational and rotational energy components of robot motion. The travel energy is proportional to the Euclidean distance traveled between consecutive waypoints, representing the linear movement cost. The turn energy is derived from the change in heading angle between two consecutive path segments, which reflects the steering effort required to execute turns.

Both components are computed for every path generated by the PRM and PRM–GA algorithms in the MATLAB R2023a simulation environment using a 10×10 binary occupancy grid. The total energy of each path is then used as the fitness criterion in the optimization process, where paths with lower total energy are considered more efficient.

The proposed PRM–GA hybrid method was validated by comparing the simulated energy results with those of the baseline PRM approach. Although the evaluation is based on simulation, the modeling approach reflects energy behaviors observed in mobile robots operating in structured indoor environments, thereby providing a realistic performance benchmark.

4.4 | Selection

The selection process ensures that the best-performing paths 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.

4.5 | Crossover

Crossover is a genetic operator that combines two selected paths (parents) to create new paths (offspring). The idea is to blend the traits of both parents to produce new solutions that may perform better. The process involves the following steps.

1. Pairing Parents: The top-selected paths are paired in pairs of two. These pairs are randomly selected from the top ranked individuals.
2. Crossover Point: A random crossover point is chosen within the path, typically not at the start or end of the path to ensure a valid exchange. For instance, a crossover point might be at a random position along the route, ensuring that both parents contribute a portion of their path to the offspring.
3. Offspring Creation: The first offspring takes the first portion of its path from the first parent up to the crossover point and the rest from the second parent. Similarly, the second offspring takes the first part of the path from the second parent and the latter portion from the first parent. This produces two new offspring paths, combining the traits of both parents in different ways.

4.6 | Mutation

Following crossover, mutation introduces small adjustments to selected paths, such as minor shifts in waypoint positions or changes in the number of intermediate waypoints. This allows the algorithm to escape local optima and explore alternative routes that may yield better solutions. Each mutation is constrained to ensure feasible and collision-free paths: waypoints must remain within the grid boundaries, avoid obstacles, and maintain path connectivity. For example, a waypoint Pi may be shifted by one cell in any direction; the new position is accepted only if it satisfies all constraints, otherwise the mutation is discarded or retried. This ensures mutation enhances diversity while preserving the reliability and safety of multirobot navigation.

4.7 | New Population Generation

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 GA continues to iterate for a predefined number of generations (in this case, 400), gradually improving the quality of the population with each cycle.

4.8 | Collision Avoidance

To prevent collisions between robots, a dynamic, proximity based collision avoidance system was implemented. The system relied on a predefined collision threshold value. If the distance between two robots fell below this threshold, the system would trigger avoidance measures to prevent a collision.

The collision avoidance strategy operates by first computing the distance between the current position of a robot and the positions of all other robots in the environment. If any robot was found to be within the defined threshold, an avoidance direction was calculated. This direction was determined as the unit vector pointing away from the other robot, derived by normalizing the difference between the robot's current position P_{Robot} and the position of the nearby robot $P_{\text{Other Robot}}$ (Equation (11)).

$$\text{Direction Avoidance} = \frac{P_{\text{Robot}} - P_{\text{Other Robot}}}{|P_{\text{Robot}} - P_{\text{Other Robot}}|}. \tag{11}$$

The Direction Avoidance vector is calculated by taking the difference between the position of the other robot $P_{\text{Other Robot}}$ and the position of the robot executing the avoidance P_{Robot}. The resulting vector is then normalized by dividing by the Euclidean distance $|P_{\text{Robot}} - P_{\text{Other Robot}}|$ between the two robots. This ensures that the avoidance direction vector has a magnitude of 1, making it a unit vector pointing away from the other robot. This direction is then used to adjust the robot's position to maintain a safe distance.

Once the avoidance direction is established, the robot's new position is adjusted by moving a fixed moving distance value:

$$\text{New Position}_{\text{Robot}} = P_{\text{Robot}} + (\text{Moving Distance} \times \text{Direction Avoidance}). \tag{12}$$

This distance ensures that the robot maintains a safe distance while avoiding collisions. The process iteratively checks all robot pairs and applies the avoidance strategy to any pair within the threshold distance. Robot positions and avoidance calculations are updated at a fixed sampling time of 0.1 s, ensuring consistent evaluation of inter-robot distances and smooth motion along paths. On the other hand, the number of turns increases compared to the original path, which unintentionally raises the total turn energy component when calculating the overall energy consumption.

The robot size plays an important role in determining parameters such as the collision avoidance threshold and moving distance. The robots in this system are equal in size and have a radius r of 0.25 m ($r = 0.25$ m). Therefore, to avoid collision, a minimum clearance of $2r = 0.5$ m must be maintained between the robots. In this system, the collision avoidance threshold was set at 0.6 m, and the moving distance was set at 0.1 m. These values were determined through extensive trials to balance collision prevention, detour minimization, and efficiency.

Tables 3–13 present the performance of the collision avoidance system under different combinations of collision thresholds and moving distances. The scenarios are evaluated based on robot count, task completion time, number of collisions, and general observations

TABLE 3 | Results—collision threshold 0.6 m and moving distance 0.01 m.

Robot count	Task completion time (s)	Number of collisions	Observations
2	15	0	Very slow, precise avoidance, no collisions.
3	20	0	Slow but safe, frequent minor adjustments.
4	25	0	Slow with precise movements, highly cautious.
5	30	0	Slow but safe, highly cautious movements.
6	35	0	Very slow, more frequent adjustments.
7	40	0	Slow, steady performance with high precision.
8	45	0	Very slow due to small movements and frequent adjustments.
9	50	0	Extremely slow with very high precision, frequent corrections.

TABLE 4 | Results—collision threshold 0.6 m and moving distance 0.05 m.

Robot count	Task completion time (s)	Number of collisions	Observations
2	12	0	Good prevention, moderate delays, no collisions.
3	16	0	Good collision prevention with moderate delays.
4	20	0	Good avoidance with moderate delays.
5	25	0	Slightly faster than 0.01 move distance, but still slow.
6	30	0	Adequate collision avoidance with moderate task times.
7	35	0	Moderately slower than lower robot counts, still effective.
8	40	0	Effective, but slightly slower with higher robot count.
9	45	0	Delays increase as robot count grows but still efficient.

TABLE 5 | Results—collision threshold 0.6 m and moving distance 0.1 m.

Robot count	Task completion time (s)	Number of collisions	Observations
2	7	0	Best balance of collision prevention and speed.
3	10	0	Balanced collision prevention with reasonable speed.
4	13	0	Optimized collision avoidance with minimal delays.
5	16	0	Efficient avoidance with reasonable task completion times.
6	18	0	Slight delays, efficient performance for 6 robots.
7	22	0	Good collision avoidance but task completion time starts to increase.
8	25	0	Good collision avoidance but noticeable delays with more robots.
9	30	0	Efficient but noticeable delays as robot count increases.

TABLE 6 | Results—collision threshold 0.6 m and moving distance 0.2 m.

Robot count	Task completion time (s)	Number of collisions	Observations
2	5	0	Minor overcorrection, larger detours.
3	8	1	Larger detours, occasional inefficiency.
4	11	1	Larger detours cause slight inefficiency.
5	13	1	Larger movements cause more noticeable detours, but still functional.
6	15	1	More noticeable detours, leading to slightly slower task times.
7	18	1	Larger movements lead to more detours and reduced efficiency.
8	22	1	Detours become more significant, resulting in lower efficiency.
9	25	1	Larger move distance causes significant detours and delays.

TABLE 7 | Results—collision threshold 0.6 m and moving distance 0.5 m.

Robot count	Task completion time (s)	Number of collisions	Observations
2	4	1	Excessive detours and inefficiencies.
3	6	2	Larger detours cause more collisions and inefficiencies.
4	9	2	Excessive detours cause slower task completion times.
5	12	2	Overcorrection and wasteful detours.
6	14	2	Larger move step causes large detours and inefficiency.
7	16	3	Excessive detours, many collisions, and slow task completion.
8	20	3	Significant inefficiencies and collisions.
9	22	4	Excessive detours and a high number of collisions.

TABLE 8 | Collision threshold 0.7 m and moving distance 0.01 m.

Robot count	Task completion time (s)	Number of collisions	Observations
2	16	0	Very slow, precise avoidance, no collisions.
3	22	0	Slow but safe, frequent minor adjustments.
4	28	0	Slow with precise movements, highly cautious.
5	34	0	Slow but safe, highly cautious movements.
6	40	0	Very slow, more frequent adjustments.
7	46	0	Slow, steady performance with high precision.
8	52	0	Very slow due to small movements and frequent adjustments.
9	58	0	Extremely slow with very high precision, frequent corrections.

TABLE 9 | Results—collision threshold 0.7 m and moving distance 0.05 m.

Robot count	Task completion time (s)	Number of collisions	Observations
2	13	0	Good prevention, moderate delays, no collisions.
3	18	0	Good collision prevention with moderate delays.
4	23	0	Good avoidance with moderate delays.
5	28	0	Slightly faster than 0.01 move distance, but still slow.

(Continues)

TABLE 9 | (Continued)

Robot count	Task completion time (s)	Number of collisions	Observations
6	33	0	Adequate collision avoidance with moderate task times.
7	38	0	Moderately slower than lower robot counts, still effective.
8	43	0	Effective, but slightly slower with higher robot count.
9	48	0	Delays increase as robot count grows but still efficient.

TABLE 10 | Results—collision threshold 0.7 m and moving distance 0.1 m.

Robot count	Task completion time (s)	Number of collisions	Observations
2	8	0	Balanced collision prevention and speed.
3	12	0	Balanced collision prevention with reasonable speed.
4	16	0	Optimized collision avoidance with minimal delays.
5	20	1	Efficient avoidance with reasonable task completion times.
6	24	1	Slight delays, efficient performance for 6 robots.
7	28	1	Slight delays.
8	32	1	Good collision avoidance but noticeable delays with more robots.
9	36	1	Efficient but noticeable delays as robot count increases.

TABLE 11 | Results—collision threshold 0.7 m and moving distance 0.2 m.

Robot count	Task completion time (s)	Number of collisions	Observations
2	6	0	Minor overcorrection, larger detours.
3	10	1	Larger detours, occasional inefficiency.
4	14	1	Larger detours cause slight inefficiency.
5	18	2	More noticeable detours.
6	22	2	More noticeable detours, leading to slightly slower task times.
7	26	2	Larger movements lead to more detours and reduced efficiency.
8	30	2	Detours become more significant, resulting in lower efficiency.
9	34	2	Larger move distance causes significant detours and delays.

TABLE 12 | Results—collision threshold 0.7 m and moving distance 0.5 m.

Robot count	Task completion time (s)	Number of collisions	Observations
2	5	1	Excessive detours and inefficiencies.
3	7	2	Larger detours cause more collisions and inefficiencies.
4	11	2	Excessive detours cause slower task completion times.
5	15	3	Overcorrection and wasteful detours.
6	19	3	Larger move step causes large detours and inefficiency.
7	23	4	Excessive detours, many collisions, and slow task completion.
8	27	4	Significant inefficiencies and collisions.
9	31	5	Excessive detours and a high number of collisions.

TABLE 13 | Initial simulation parameters and environment setup.

Parameter	Value/description
Simulation platform	MATLAB R2023a
Environment size	10×10 binary occupancy grid
Number of nodes (PRM)	1000
Connection radius (PRM)	1 unit
Start and goal positions	Predefined for each robot
Number of robots	2–9
Robot radius	0.25 m
Collision threshold	0.6 m
Movement step size	0.1 m
Population size (GA)	100
Number of generations	400
Crossover/mutation rates	0.8/0.05
Elitism rate	Top 5% retained
Selection method	Elitist ranking
Fitness function	Energy-based (Equations (2)–(10))
Collision avoidance	Normalized avoidance vector (Equations (11) and (12))
Travel time model	Distance and turn-speed model (Equations (13) and (14))

regarding system behavior. These tables help in comparing the efficiency of various configurations, highlighting the trade-offs between avoiding collisions, minimizing detours, and maintaining overall system performance. Each table corresponds to a specific combination of threshold and moving distance, allowing for a comprehensive analysis of how different parameters affect the system's ability to handle multiple robots in dynamic environments.

4.9 | Time Calculation

The methodology for calculating the travel time of a robotic path is designed to incorporate both the distance and the turns along the path, which affect the robot's speed due to the need to decelerate for sharper turns. This approach balances speed optimization with path efficiency.

1. Base Speed: A base speed is established to define the robot's speed when moving in a straight line or along gentle curves. This speed is adjusted based on the angle between consecutive path segments to account for turns.
2. Total Time Calculation: For each segment, the time required is computed as the ratio of the segment distance to the speed:

$$\text{Time}_{\text{segment}} = \frac{\text{Distance}_{\text{segment}}}{\text{Speed}_{\text{segment}}}. \tag{13}$$

The total time to traverse the entire path is the sum of the times for each segment ($\text{Time}_{\text{Segment}}$), where n is the total number of waypoints in the path and i is the index as shown by Equation (14)

$$\text{Total Time} = \textstyle\sum_{i=2}^{n} \text{Time}_{\text{segment}}. \tag{14}$$

4.10 | Path Animation

In this simulation, animated visualization was implemented to illustrate each robot's movement along its optimized path, providing a thorough understanding of the pathfinding process and the progression of each robot over time. Unlike static representations that show only start and end points, the animation decomposed each robot's trajectory into sequential frames, updating positions at regular intervals to capture incremental movements from start to finish.

The animated approach proved critical, as the optimized paths alone did not ensure that robots would meet timely placements or maintain sufficient distances to avoid collisions. By animating the paths, the simulation addressed these limitations, presenting a real-time visualization of robot positions that allowed for assessments of both timing adherence and spatial relationships between robots.

Each frame of the animation provided a precise indication of the robots' positions relative to one another and any obstacles. This visual detail was essential in verifying collision avoidance, particularly in situations where robots operated in close proximity or confined spaces. Furthermore, the animation allowed for real-time observation of how each robot dynamically adjusted its position or speed to prevent overlaps or collisions. By illustrating these adjustments, the animation validated the adaptability of the collision-avoidance algorithm, confirming that robots could navigate safely within shared environments. Throughout the animation, visual markers are strategically placed at consistent time intervals along each robot's path. These markers serve as checkpoints, visually capturing and marking each robot's location at various stages, which enhances the viewer's ability to track their progression over time. By including these timely markers, the animation not only reveals the robots' positions at specific moments but also highlights the broader patterns in their movement paths, such as changes in speed or direction. These markers act as supplementary reference points that make it easier to observe each robot's trajectory as well as any adjustments it makes during navigation. Viewers can gain insight into how efficiently each robot progresses toward its target, observing acceleration or deceleration phases and understanding any path deviations made for collision avoidance or spatial adjustments. This temporal layering of markers also adds depth to the animation, offering a frame-by frame overview that makes it possible to evaluate movement strategies in detail and to verify that all robots maintain the intended sequence of positions without overlap. By clearly marking these intervals, the animation enables a transparent view of each robot's movement behavior, providing essential insights into the system's operational efficiency and ensuring that all robots are moving safely and predictably within the multi-agent environment. This allows for an in depth understanding of path planning, collision management, and overall coordination. Overall, this visual approach delivered an intuitive understanding of the algorithms that governed robot movements, including spatial awareness and synchronized timing, which are vital in complex, multi-robot scenarios. The step-by-step visualization underscored the capacity of the robots to navigate both optimally and safely, emphasizing the robustness and practical viability of the proposed path-planning and collision-avoidance strategies.

5 | Results

This section presents the experimental evaluation of the proposed hybrid PRM and GA framework detailed in Section 4. The implemented methodology comprising environmental modeling, an energy-based fitness function, GA operators, collision avoidance, and time calculations was employed to generate optimized, collision-free paths for multiple robots.

Simulations were conducted in MATLAB R2023a on a 10×10 binary occupancy grid, where obstacles were represented as occupied cells using linear indices. This grid serves as a conceptual validation of the proposed PRM–GA framework. Robots were assigned predefined start and goal positions within this environment and tasked with energy-efficient navigation while avoiding collisions.

To ensure reproducibility and transparency, the initial simulation parameters and environmental configuration are summarized in Table 13. These parameters define the operating environment, robot dynamics, and optimization settings for both the PRM and GA components of the hybrid framework.

The PRM was used to generate an initial roadmap by randomly sampling 1000 nodes and connecting nodes within a 1-unit radius where obstacles were absent. The resulting shortest paths served as the initial population for GA optimization. Using the defined fitness function (Equations (3)–(10)), the GA refined paths to minimize total energy while maintaining collision-free motion.

Collision avoidance was dynamically enforced using a proximity threshold of 0.6 m, with each robot modeled as a circle of radius 0.25 m. When inter-robot distances fell below this threshold, position adjustments were applied using the normalized avoidance vector (Equation (11)) and a step size of 0.1 m (Equation (12)) to maintain safe separation.

Travel time for each robot was calculated based on segment distances and turn-related speed reductions (Equation (14)), representing realistic kinematic behavior. Robot motion was visualized through time-based interpolation along optimized paths, supplemented by periodic position markers for real-time tracking of coordination and avoidance efficiency.

To account for the stochastic nature of PRM node sampling and GA operations, all experiments were conducted 30 independent times for each robot count, with the MATLAB random number generator initialized using different seeds for each run. Results were aggregated by reporting the mean and standard deviation of total energy consumption and travel time. The performance of the proposed hybrid PRM–GA framework was benchmarked against a PRM-only approach, as well as standard path-planning algorithms including A^* and RRT, using identical environments, start and goal positions, and robot counts. These evaluation metrics quantify the improvements in energy efficiency and travel time achieved by the hybrid approach.

These three algorithms were selected for comparison because PRM-only represents the baseline PRM method, A^* is a widely used classical grid-based planning algorithm, and RRT is a standard sampling-based approach. This comparison can be used to measure the effectiveness of the proposed PRM–GA framework across both classical and sampling-based planning strategies.

The configuration shown in Table 14 represents the reference setup used for all experiments. For scenarios with fewer robots, the same arrangement is followed by selecting the first *n* robots from the table. For example, the 2-robot case uses Robots 1 and 2, the 3-robot case uses Robots 1–3, and so on.

1. Path Planning for Two Robots: The improved PRM–GA algorithm was initially applied to two robots using the reference configuration shown in Table 14. For this case, only Robots 1 and 2 were considered, each assigned a distinct color to simplify visual tracking.

Figure 2 displays the paths generated for each robot using the optimized PRM–GA method.

a. GA–PRM Vs PRM Vs A^* Vs RRT Energy Comparison for Two Robots: Table 15 and Figure 3 indicate the energy consumption of two robots evaluated across four path

TABLE 14 | Initial setup for nine robots.

Robot number	Color of the robot	Starting position	Ending position
1	Red	(9, 1)	(1, 9)
2	Blue	(1, 9)	(9, 1)
3	Green	(1, 7)	(7, 1)
4	Magenta	(9, 9)	(1, 1)
5	Cyan	(7, 2)	(1, 4)
6	Yellow	(7, 1)	(1, 7)
7	Black	(1, 4)	(7, 2)
8	Gray	(3, 1)	(6, 9)
9	Orange	(2, 8)	(8, 9)

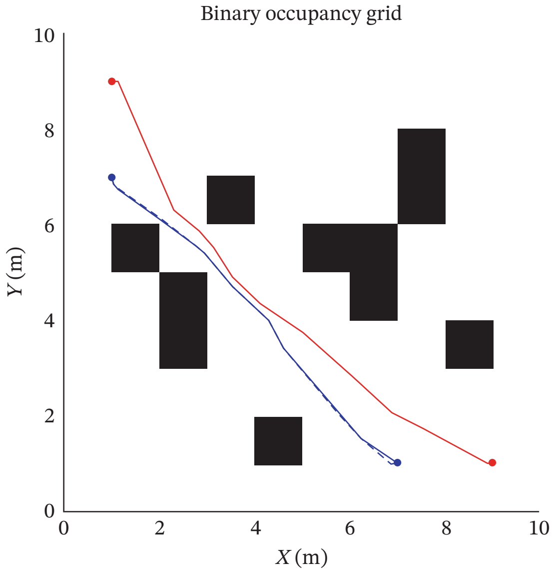

FIGURE 2 | Improved PRM–GA path for two robots.

TABLE 15 | Energy comparison for two robots in Joule (J).

Robot	PRM–GA	PRM	A*	RRT
1	42.94	72.55	102.01	581.19
2	77.69	134.78	102.01	610.07

planning algorithms: PRM–GA, PRM, A*, and RRT for the two robots scenario. The results show a significant variation in energy efficiency among the methods.

For Robot 1, the PRM–GA algorithm recorded the lowest energy consumption at 42.94 J, followed by PRM at 72.55 J and A* at 102.01 J. In contrast, RRT showed a significantly higher energy usage of 581.19 J. A similar trend is observed for Robot 2, where PRM–GA again indicated the lowest energy consumption at 77.69 J, compared to PRM (134.78 J) and A* (102.01 J), while RRT exhibited the highest energy usage at 610.07 J.

Figure 3 illustrates the energy consumption of each robot across the four techniques employed in this study.

b. GA–PRM vs PRM Vs A* vs RRT Time Comparison for Two Robots: As given in Table 16 and Figure 4, PRM–GA algorithm consistently outperforms the other techniques, achieving the shortest completion times of 11.43 and 11.63 s for Robot 1 and Robot 2, respectively. While PRM and A* show similar performance, with marginally higher times, the RRT algorithm exhibits the longest execution durations, reaching ~14.8 s for both robots.

2. Path Planning for Three Robots: The improved PRM–GA algorithm was initially applied to three robots using the reference configuration shown in Table 14. For this case, only Robots 1, 2, and 3 were considered, each assigned a distinct color to simplify visual tracking.

Figure 5 displays the paths generated for each robot using the optimized PRM–GA method in three robots scenario.

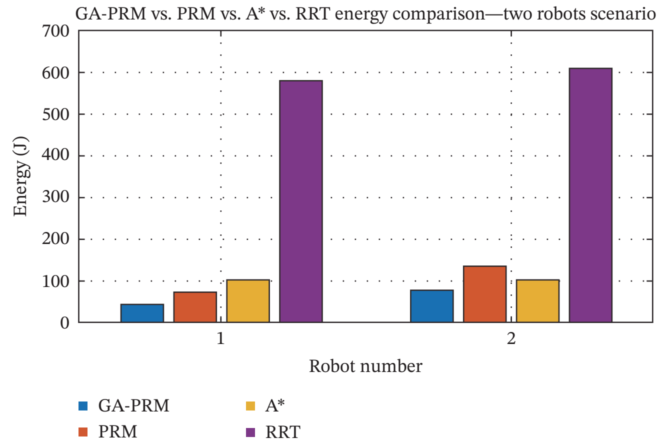

FIGURE 3 | GA–PRM vs PRM vs A* vs RRT energy comparison for two robots.

TABLE 16 | Time comparison for two robots in seconds (s).

Robot	PRM–GA	PRM	A*	RRT
1	11.43	11.60	11.90	14.76
2	11.63	11.93	11.90	14.80

a. GA–PRM vs PRM vs A* vs RRT Energy Comparison for Three Robots: As per Table 17 and Figure 6, for three robot scenario, PRM–GA algorithm is the most efficient among the four methods, with Robot 1 consuming 73.21 J, Robot 2 32.52 J, and Robot 3 37.04 J. In comparison, PRM and A* require substantially more energy, particularly PRM, which reaches up to 120.29 J for Robot 1 and 116.96 J for Robot 3. The RRT algorithm exhibits the highest energy consumption, with values ranging from 460.65 to 708.25 J, reflecting its random path exploration and longer traversal distances.

b. GA- PRM vs PRM vs A* vs RRT Time Comparison for Three Robots: Table 18 and Figure 7 indicates that the PRM–GA algorithm consistently achieves the lowest completion times in three robot scenario, with Robot 1 completing its path in 11.60 s, Robot 2 in 11.43 s, and Robot 3 in 8.52 s. PRM and A* show similar performance across the

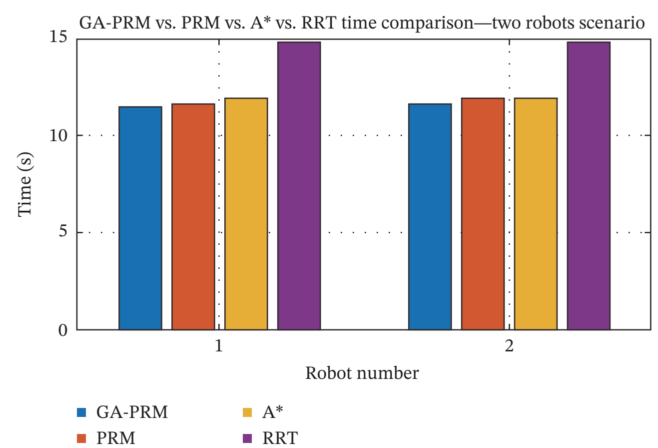

FIGURE 4 | GA–PRM vs PRM vs A* vs RRT time comparison for two robots.

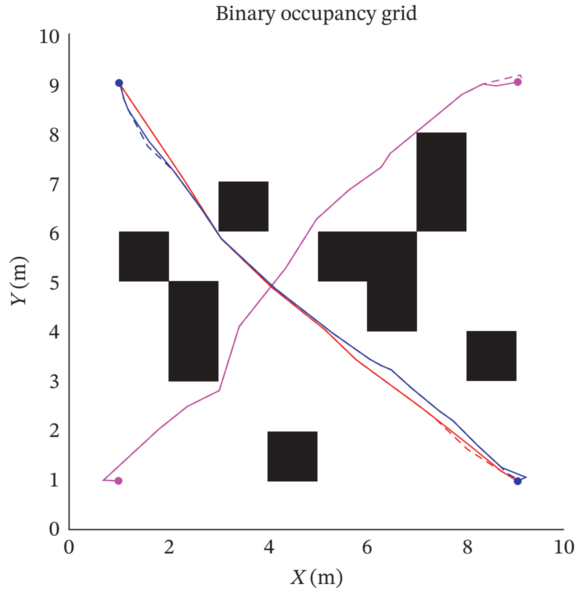

FIGURE 5 | Improved PRM–GA path for three robots.

TABLE 17 | Energy comparison for three robots in Joule (J).

Robot	PRM–GA	PRM	A^*	RRT
1	73.21	120.29	102.01	596.34
2	32.52	70.26	102.01	708.25
3	37.04	116.96	99.13	460.65

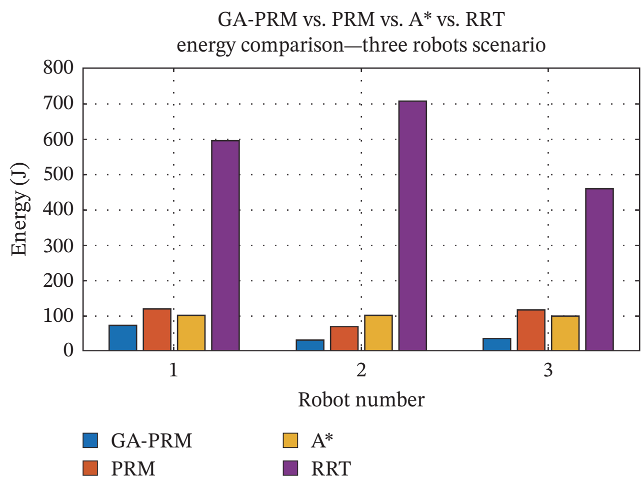

FIGURE 6 | GA–PRM vs PRM vs A^* vs RRT energy comparison for three robots.

TABLE 18 | Time comparison for three robots in seconds (s).

Robot	PRM–GA	PRM	A^*	RRT
1	11.60	11.89	11.90	16.44
2	11.43	11.52	11.90	14.79
3	8.52	8.68	9.07	11.99

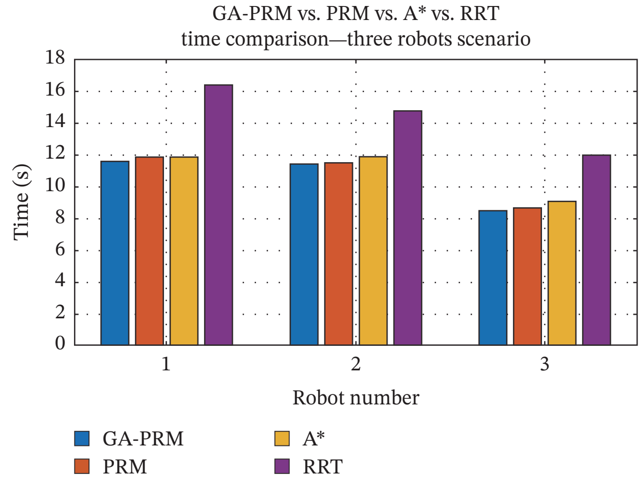

FIGURE 7 | Time comparison of GA–PRM vs PRM vs A^* vs RRT methods for three robots.

robots, with only marginally higher times, indicating that deterministic planning methods are effective in the given environment. In contrast, RRT requires significantly longer durations, ranging from 11.99 to 16.44 s, due to its randomized exploration and less direct paths.

3. Path Planning for Four Robots: The improved PRM–GA algorithm was initially applied to four robots using the reference configuration shown in Table 14. For this case, only Robots 1, 2, 3, and 4 were considered, each assigned a distinct color to simplify visual tracking.

Figure 8 displays the paths generated for each robot using the optimized PRM–GA method in four robots scenario.

a. GA–PRM vs PRM Vs A^* vs RRT Energy Comparison for Four Robots: As shown in Table 19 and Figure 9, energy consumption analysis for four robots shows that the

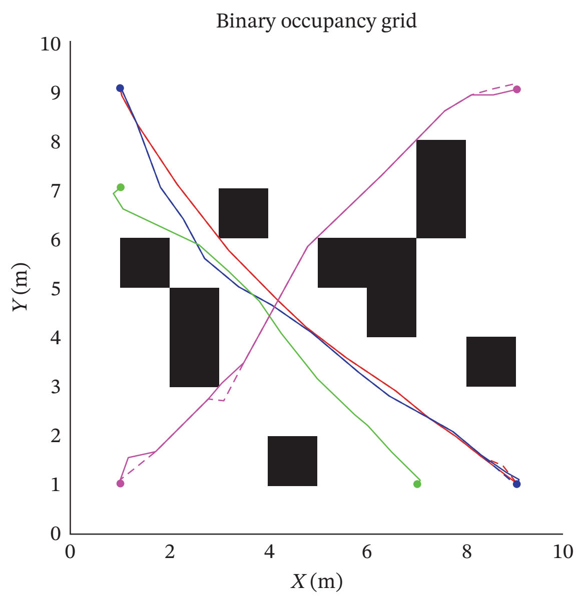

FIGURE 8 | Improved PRM–GA path for four robots.

TABLE 19 | Energy comparison for four robots in Joule (J).

Robot	PRM–GA	PRM	A^*	RRT
1	33.76	105.04	102.01	576.20
2	56.12	112.92	102.01	397.14
3	43.27	131.30	99.13	452.15
4	94.16	154.66	79.54	417.79

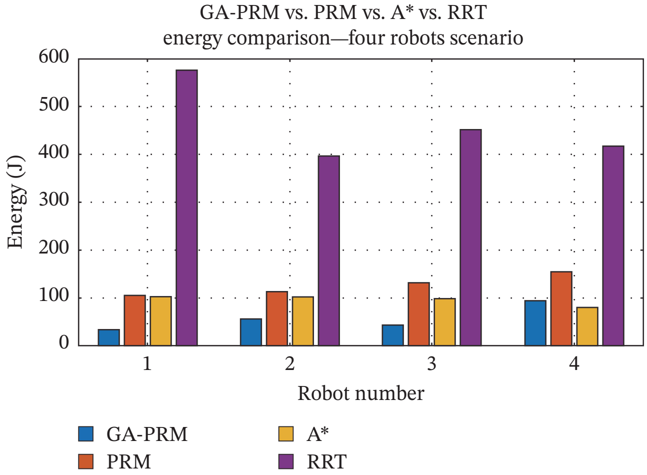

FIGURE 9 | Energy comparison of GA–PRM vs PRM Vs A^* vs RRT methods for four robots.

PRM–GA algorithm is the most energy-efficient method, with energy usage ranging from 33.76 to 94.16 J across the robots. In comparison, PRM and A^* exhibit significantly higher energy demands, particularly PRM, which reaches up to 154.66 J for Robot 4. RRT consistently consumes the most energy, with values between 397.14 and 576.20 J, reflecting its randomized path exploration and longer traversal distances.

b. GA- PRM vs PRM vs A^* vs RRT Time Comparison for Four Robots: As shown in Table 20 and Figure 10, The time comparison for four robots demonstrates that the PRM–GA algorithm consistently achieves the lowest or near-lowest completion times, with robots finishing their paths in 8.54 to 11.82 s. PRM and A^* show slightly higher times across all robots, indicating that deterministic planning methods perform effectively but are less optimized than PRM–GA. In contrast, RRT requires considerably longer durations, ranging from 10.96 to 16.75 s, due to its randomized exploration and less direct paths.

TABLE 20 | Time comparison for four robots in seconds (s).

Robot	PRM–GA	PRM	A^*	RRT
1	11.46	11.54	11.90	16.75
2	11.53	11.78	11.90	13.88
3	8.54	9.14	9.07	10.96
4	11.82	12.12	11.90	13.99

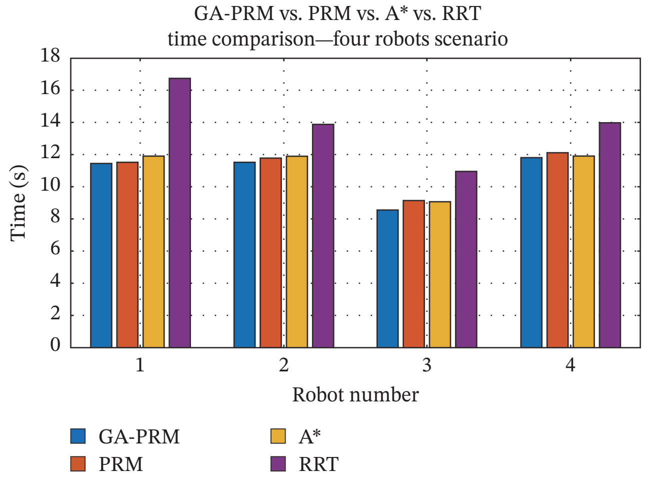

FIGURE 10 | Time comparison of GA–PRM vs PRM Vs A^* vs RRT methods for four robots.

4. Path Planning for Five Robots: The improved PRM–GA algorithm was initially applied to five robots using the reference configuration shown in Table 14. For this case, only Robots 1, 2, 3, 4, and 5 were considered, each assigned a distinct color to simplify visual tracking.

Figure 11 displays the paths generated for each robot using the optimized PRM–GA method in five robots scenario (Figure 12).

a. GA–PRM vs PRM vs A^* vs RRT Energy Comparison for Five Robots: The energy consumption analysis for five robots are shown in Table 21 and Figure 12. The results indicate that the PRM–GA algorithm remains the most energy-efficient, with energy usage ranging from 25.10 to 72.52 J. PRM and A^* consume significantly more energy, with PRM reaching up to 152.12 J for Robot 2, while A^* ranges between 74.41 and 102.01 J. RRT consistently shows the highest energy consumption, varying from 335.29 to 843.26 J.

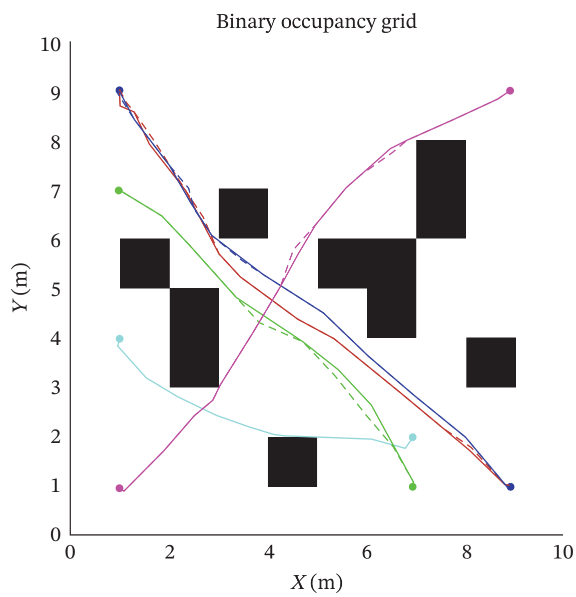

FIGURE 11 | Improved PRM–GA path for five robots.

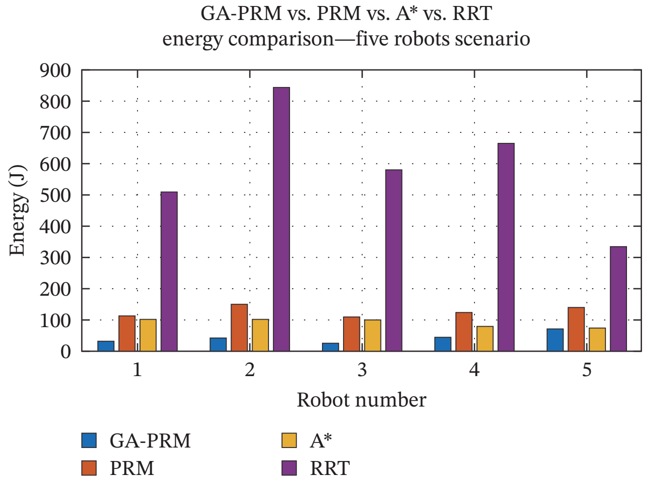

FIGURE 12 | Energy comparison of GA–PRM vs PRM vs A* vs RRT methods for five robots.

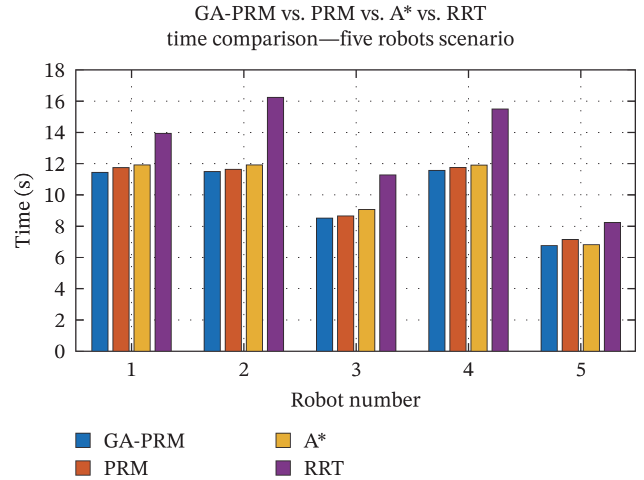

FIGURE 13 | Time comparison of GA–PRM vs PRM vs A* vs RRT methods for five robots.

TABLE 21 | Energy comparison for five robots in Joule (J).

Robot	PRM–GA	PRM	A*	RRT
1	32.12	112.70	102.01	509.78
2	41.99	152.12	102.01	843.26
3	25.10	110.35	99.13	579.60
4	44.36	124.21	79.54	665.68
5	72.52	141.93	74.41	335.29

b. GA–PRM vs PRM vs A* vs RRT Time Comparison for Five Robots: Table 22 and Figure 13 show the path planning times for five robots using the PRM and PRM–GA methods whereas from Figure 14 the results are graphically shown. According to the results, PRM–GA algorithm consistently achieves the lowest or near-lowest completion times, ranging from 6.73 to 11.55 s. PRM and A* demonstrate slightly higher times across most robots, indicating effective performance of deterministic methods, though generally less optimized than PRM–GA. RRT exhibits the longest completion times, ranging from 8.23 to 16.23 s. Overall, these results confirm that PRM–GA provides the most efficient path planning in terms of time across all five robots.

5. Path Planning for Six Robots: The improved PRM–GA algorithm was initially applied to six robots using the reference configuration shown in Table 14. For this case, only Robots 1, 2, 3, 4, 5, and 6 were considered, each as a distinct color to simplify visual tracking.

TABLE 22 | Time comparison for five robots in seconds (s).

Robot	PRM–GA	PRM	A*	RRT
1	11.44	11.72	11.90	13.94
2	11.46	11.61	11.90	16.23
3	8.51	8.65	9.07	11.30
4	11.55	11.71	11.90	15.48
5	6.73	7.11	6.83	8.23

Figure 15 displays the paths generated for each robot the optimized PRM–GA method in six robots scenario

a. GA–PRM vs PRM vs A* Vs RRT Energy Comparison for Six Robots: The results indicated in Table 23 and Figure 16 confirm that the PRM–GA algorithm has the minimum energy consumption ranging from 23.46 J (Robot 3) to 76.40 J (Robot 2), reflecting its ability to generate optimized paths for multiple robots. PRM and A* consume substantially more energy across all robots, with PRM reaching a maximum of 180.81 J for Robot 4, while A* ranges between 74.41 and 102.01 J, indicating moderate efficiency but less optimization compared to PRM–GA. RRT consistently shows the highest energy usage, varying from 296.75 to 648.31 J, which suggests that its paths are generally longer or less energy-optimal.
b. GA–PRM vs PRM vs A* vs RRT Time Comparison for Six Robots: As per Table 24 and Figure 17, the time comparison for six robots indicates that the PRM–GA algorithm

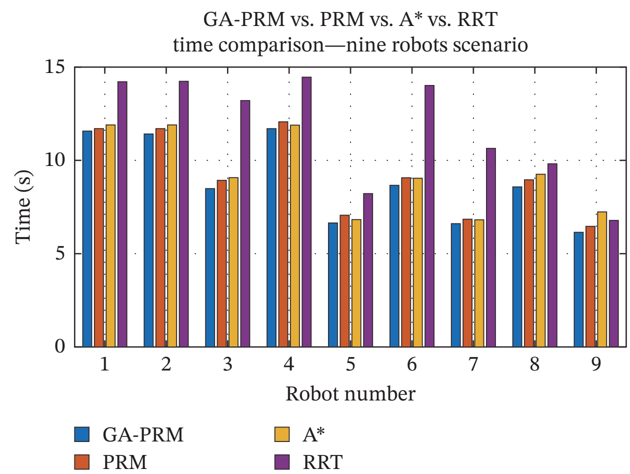

FIGURE 14 | Time comparison of GA–PRM vs PRM vs A* vs RRT methods for nine robots.

TABLE 23 | Energy comparison for six robots in Joule (J).

Robot	PRM–GA	PRM	A^*	RRT
1	39.22	97.56	102.01	648.31
2	76.40	156.05	102.01	641.45
3	23.46	52.98	99.13	325.73
4	43.11	180.81	79.54	420.14
5	27.27	146.50	74.41	296.75
6	28.46	66.40	76.66	505.21

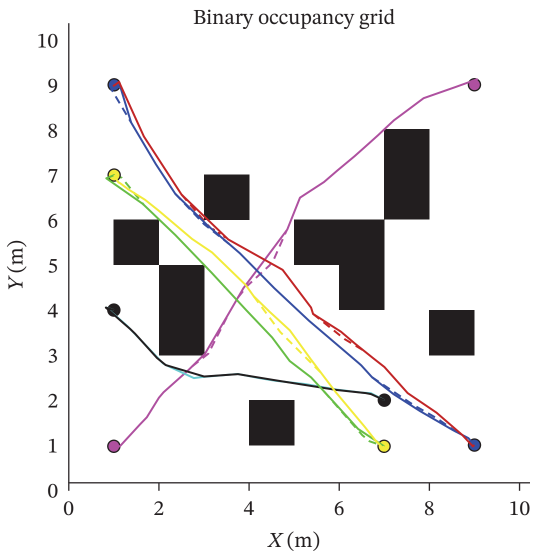

FIGURE 15 | Improved PRM–GA path for six robots.

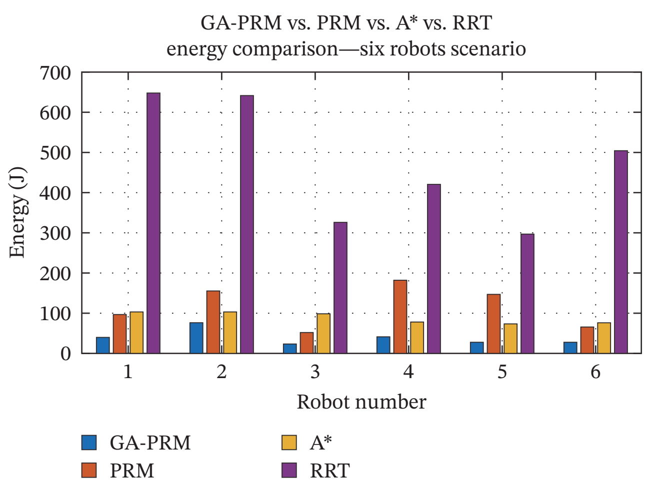

FIGURE 16 | Energy comparison of GA–PRM vs PRM vs A^* vs RRT methods for six robots.

consistently achieves the lowest completion times across most robots, ranging from 6.46 s for Robot 5 to 11.62 s for Robot 2. For PRM, the times range from 7.07 to 12.13 s, while A^* ranges from 6.83 to 11.90 s. RRT shows times between 7.07 and 12.13 s, which are generally higher than PRM–GA but similar to PRM in some cases. Specifically, Robot 1 completes in 11.43 s with PRM–GA

TABLE 24 | Time comparison for six robots in seconds (s).

Robot	PRM–GA	PRM	A^*	RRT
1	11.43	11.50	11.90	11.50
2	11.62	11.69	11.90	11.69
3	8.51	8.54	9.07	8.54
4	11.53	12.13	11.90	12.13
5	6.46	7.07	6.83	7.07
6	8.51	8.68	9.07	8.68

compared to 11.50 s for both PRM and RRT, and 11.90 s for A^*. Robot 3 completes in 8.51 s with PRM–GA, slightly faster than 8.54 s for PRM and RRT, and 9.07 s for A^*. Overall, these results confirm that PRM–GA provides the most time-efficient path planning, maintaining consistent performance across robots with different starting positions.

6. Path Planning for Seven Robots: The improved PRM–GA algorithm was initially applied to seven robots using the reference configuration shown in Table 14. For this case, only Robots 1, 2, 3, 4, 5, 6, and 7 were considered, each assigned a distinct color to simplify visual tracking.

Figure 18 displays the paths generated for each robot using the optimized PRM–GA method in seven robots scenario

a. GA–PRM vs PRM vs A^* vs RRT Energy Comparison for Seven Robots: The energy consumption analysis for seven robots as indicated in Table 25 and Figure 19 shows that the PRM–GA algorithm remains the most energy-efficient, with values ranging from 17.57 J for Robot 6 to 75.15 J for Robot 4. PRM and A^* require substantially higher energy, with PRM ranging from 24.52 to 151.63 J and A^* from 74.41 to 102.01 J. RRT exhibits the highest energy usage, varying between 339.82 and 624.87 J across the robots. For instance, Robot 1 consumes 63.02 J with PRM–GA compared to 102.60 J for PRM, 102.01 J for A^*, and 624.87 J for RRT, while Robot 6 shows the lowest PRM–GA consumption at 17.57 J.

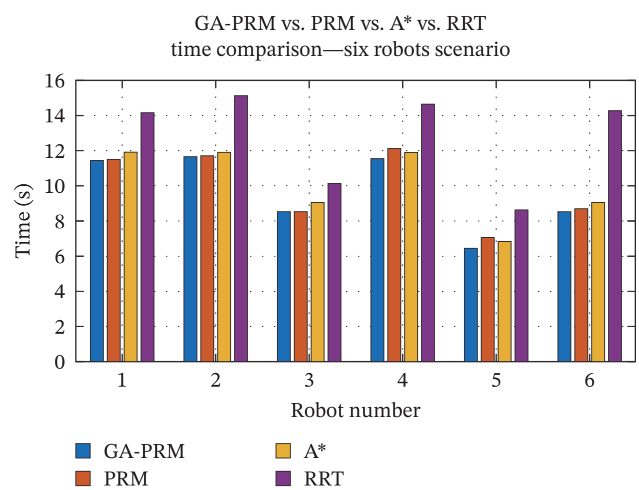

FIGURE 17 | Time comparison of GA–PRM vs PRM vs A^* vs RRT methods for six robots.

TABLE 25 | Energy comparison for seven robots in Joule (J).

Robot	PRM–GA	PRM	A^*	RRT
1	63.02	102.60	102.01	624.87
2	45.82	113.32	102.01	417.82
3	20.02	72.32	99.13	544.34
4	75.15	150.91	79.54	576.34
5	34.47	104.33	74.41	460.14
6	17.57	24.52	76.66	503.67
7	50.80	151.63	96.87	339.82

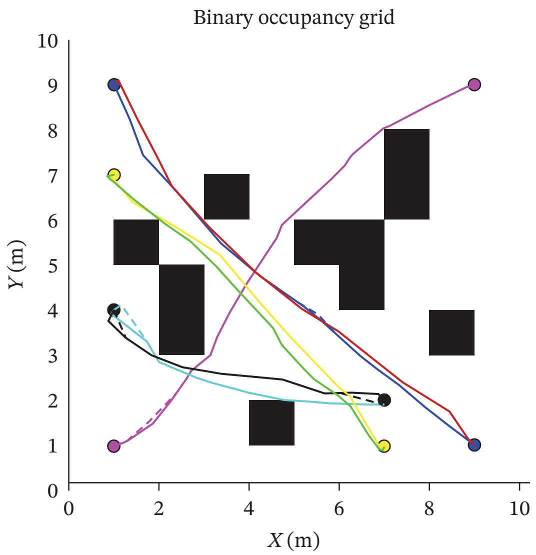

FIGURE 18 | Improved PRM–GA path for seven robots.

b. GA–PRM vs PRM vs A^* vs RRT Time Comparison for Seven Robots: Table 26 and Figure 20 show the time comparison results for seven robots scenario. The results demonstrate that the PRM–GA algorithm consistently achieves the lowest or near-lowest completion times, ranging from 6.46 s for Robot 5 to 11.77 s for Robot 4. PRM times range from 6.69 to 11.90 s, while A^* varies between 6.83 and 11.90 s. RRT records the highest times, from 10.10 to 15.48 s. For example, Robot 1 completes in 11.47 s with PRM–GA, compared to 11.71 s with PRM, 11.90 s with A^*, and 14.94 s with

TABLE 26 | Time comparison for seven robots in seconds (s).

Robot	PRM–GA	PRM	A^*	RRT
1	11.47	11.71	11.90	14.94
2	11.45	11.52	11.90	14.44
3	8.51	8.65	9.07	12.49
4	11.77	11.90	11.90	15.48
5	6.46	6.69	6.83	10.10
6	8.50	8.51	9.07	13.43
7	6.65	7.12	6.83	10.23

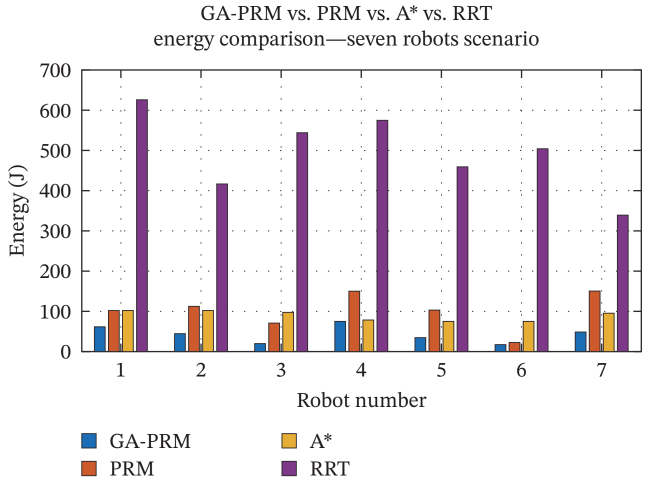

FIGURE 19 | Energy comparison of GA–PRM vs PRM vs A^* vs RRT methods for seven robots.

RRT, whereas Robot 5 achieves 6.46 s with PRM–GA, slightly better than 6.69 s with PRM and 6.83 s with A^*. Overall, PRM–GA demonstrates the most time efficient and consistent performance, making it ideal for multi-robot path planning.

7. Path Planning for Eight Robots: The improved PRM–GA algorithm was initially applied to eight robots using the reference configuration shown in Table 14. For this case, only Robots 1, 2, 3, 4, 5, 6, 7, and 8 were considered, each assigned a distinct color to simplify visual tracking.

Figure 21 displays the paths generated for each robot using the optimized PRM–GA method in eight robots scenario.

a. GA–PRM vs PRM vs A^* vs RRT Energy Comparison for Eight Robots: The energy consumption results as per Table 27 and Figure 22 reveal that PRM–GA consistently consumes the least energy, ranging from 17.57 J for Robot 6 to 75.15 J for Robot 4. PRM shows higher energy usage, between 24.52 and 151.63 J, while A^* varies from 74.41 to 102.01 J. RRT exhibits the highest energy consumption,

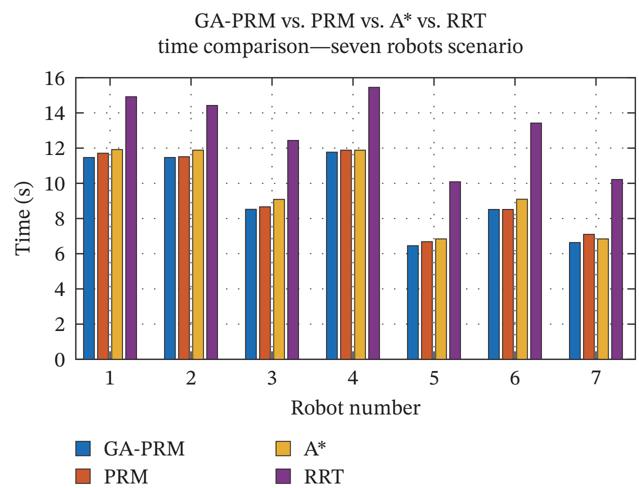

FIGURE 20 | Time comparison of GA–PRM vs PRM vs A^* vs RRT methods for seven robots.

TABLE 27 | Energy comparison for eight robots in Joule (J).

Robot	PRM–GA	PRM	A*	RRT
1	63.02	102.60	102.01	624.87
2	45.82	113.32	102.01	417.82
3	20.02	72.32	99.13	544.34
4	75.15	150.91	79.54	576.34
5	34.47	104.33	74.41	460.14
6	17.57	24.52	76.66	503.67
7	50.80	151.63	96.87	339.82
8	23.53	75.34	76.89	459.54

TABLE 28 | Time comparison for eight robots in seconds (s).

Robot	PRM–GA	PRM	A*	RRT
1	11.47	11.71	11.90	14.94
2	11.45	11.52	11.90	14.44
3	8.51	8.65	9.07	12.49
4	11.77	11.90	11.90	15.48
5	6.46	6.69	6.83	10.10
6	8.50	8.51	9.07	13.43
7	6.65	7.12	6.83	10.23
8	8.56	8.71	9.24	12.41

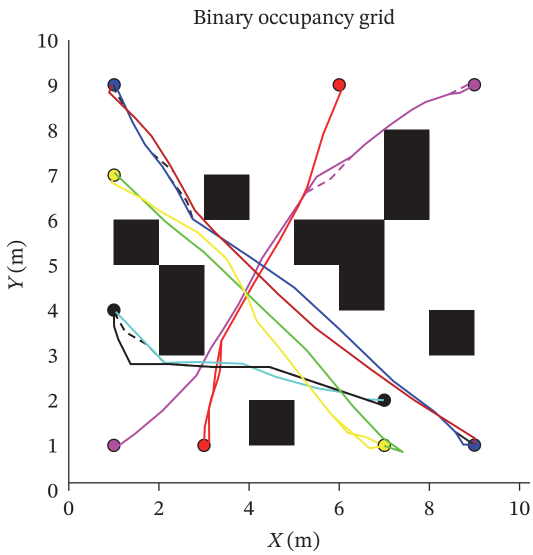

FIGURE 21 | Improved PRM–GA path for eight robots.

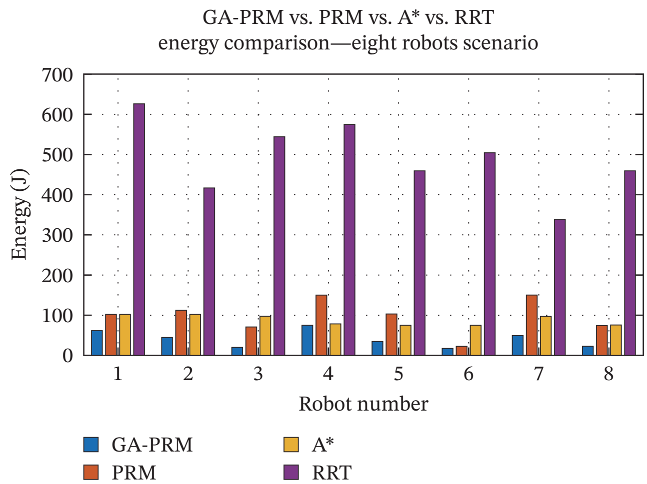

FIGURE 22 | Energy comparison of GA–PRM vs PRM vs A* vs RRT methods for eight robots.

ranging from 339.82 to 624.87 J. For instance, Robot 1 uses 63.02 J with PRM–GA compared to 102.60 J with PRM, 102.01 J with A*, and 624.87 J with RRT, while Robot 6 consumes only 17.57 J with PRM–GA versus 24.52 J with PRM and 76.66 J with A*. Overall, PRM–GA demonstrates the most energy efficient and consistent performance across all eight robots.

b. GA–PRM vs PRM vs A* vs RRT Time Comparison for Eight Robots: The path time results for eight robots are presented in Table 28 and Figure 23. The results highlight that the PRM–GA algorithm consistently achieves the lowest or near-lowest completion times, ranging from 6.46 s for Robot 5 to 11.77 s for Robot 4. PRM times vary between 6.69 and 11.90 s, while A* ranges from 6.83 to 11.90 s. RRT records the highest times, from 10.10 to 15.48 s. For example, Robot 1 completes in 11.47 s with PRM–GA, compared to 11.71 s with PRM, 11.90 s with A*, and 14.94 s with RRT, whereas Robot 5 achieves 6.46 s with PRM–GA, slightly better than 6.69 s with PRM and 6.83 s with A*. Overall, PRM–GA demonstrates the most time-efficient and consistent performance across all eight robots, making it suitable for multi-robot path planning applications.

8. Path Planning for Nine Robots: The improved PRM–GA algorithm was initially applied to nine robots using the reference configuration shown in Table 14. For this case, only Robots 1,2,3,4,5,6,7,8 and 9 were considered, each assigned a distinct color to simplify visual tracking.

Figure 24 displays the paths generated for each robot using the optimized PRM–GA method in nine robots scenario.

a. GA–PRM vs PRM vs A* vs RRT Energy Comparison for Nine Robots: The path energy results for nine robots demonstrated as per Table 29 and Figure 25 show that PRM–GA consistently consumes the least energy, ranging from 18.79 J for Robot 3 to 107.71 J for Robot 4. PRM shows higher energy usage, between 110.42 and 180.46 J, while A* varies from 74.41 to 102.01 J. RRT exhibits the highest energy consumption, ranging from 204.03 to 596.35 J. For example, Robot 1 consumes 79.47 J with PRM–GA compared to 141.04 J with PRM, 102.01 J with A*, and 596.35 J with RRT, while Robot 3 uses only 18.79 J with PRM–GA versus 135.57 J with PRM and 99.13 J with A*. Overall, PRM–GA demonstrated consistent improvements across most cases, with marked energy savings for several robots.

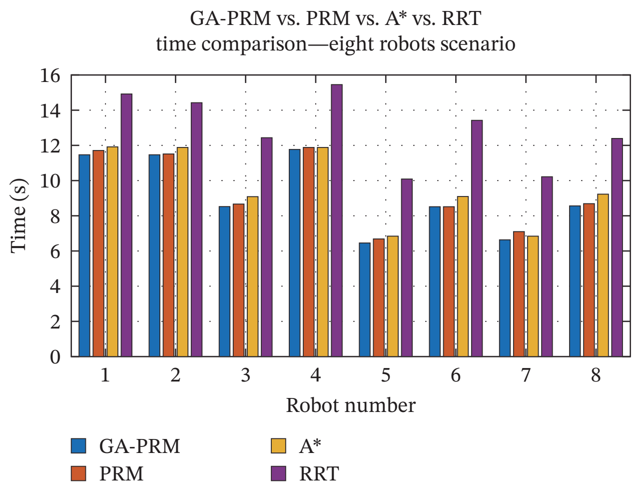

FIGURE 23 | Time comparison of GA–PRM vs PRM vs A* vs RRT methods for eight robots.

TABLE 29 | Energy comparison for nine robots in Joule (J).

Robot	PRM–GA	PRM	A*	RRT
1	79.47	141.04	102.01	596.35
2	32.52	110.42	102.01	436.83
3	18.79	135.57	99.13	492.47
4	107.71	180.46	79.54	538.53
5	64.53	130.89	74.41	311.39
6	81.15	163.89	76.66	560.70
7	40.16	110.83	96.87	565.08
8	28.08	154.19	76.89	375.23
9	42.79	152.94	97.32	204.03

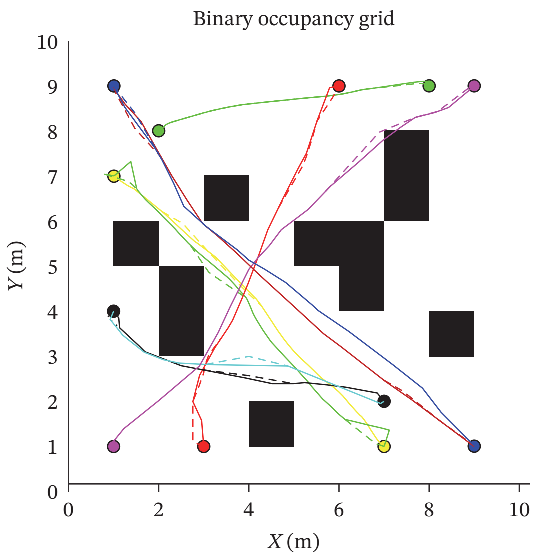

FIGURE 24 | Improved PRM–GA path for nine robots.

b. GA–PRM vs PRM vs A* vs RRT Time Comparison for Nine Robots: The time efficiency results for the nine robots as in Table 30 and Figure 14 reveal that the PRM–GA algorithm

TABLE 30 | Time comparison for nine robots in second (s).

Robot	PRM–GA	PRM	A*	RRT
1	11.58	11.70	11.90	14.22
2	11.42	11.74	11.90	14.24
3	8.50	8.93	9.07	13.20
4	11.70	12.06	11.90	14.45
5	6.65	7.07	6.83	8.23
6	8.69	9.09	9.07	14.05
7	6.61	6.84	6.83	10.63
8	8.58	8.95	9.24	9.80
9	6.17	6.48	7.24	6.79

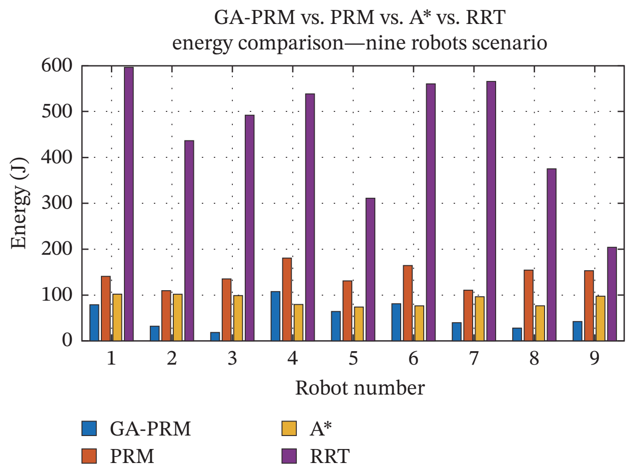

FIGURE 25 | Energy comparison of GA–PRM vs PRM vs A* vs RRT methods for nine robots.

consistently achieves the lowest or near-lowest completion times, ranging from 6.17 s for Robot 9 to 11.70 s for Robot 4. PRM times vary between 6.48 and 12.06 s, while A* ranges from 6.83 to 11.90 s. RRT records the highest times in most cases, from 6.79 to 14.45 s. For example, Robot 1 completes in 11.58 s with PRM–GA, compared to 11.70 s with PRM, 11.90 s with A*, and 14.22 s with RRT, while Robot 5 achieves 6.65 s with PRM–GA, slightly faster than 7.07 s with PRM and 6.83 s with A*. Overall, PRM–GA demonstrates the most time efficient and consistent performance across all nine robots, making it suitable for multi-robot path planning.

5.1 | Sensitivity Analysis of Algorithm Parameters

To evaluate the robustness and scalability of the proposed hybrid PRM–GA framework, a comprehensive sensitivity analysis was conducted on key algorithmic parameters under varying robot counts (from 2 to 9 robots). The objective was to examine how changes in parameter settings and team size affect overall performance, including path success rate, total energy consumption, computation time, and average path length.

The analysis considered three main parameter groups:

- PRM node density and connection radius,
- GA population size and number of generations, and
- GA crossover and mutation probabilities.

Each parameter was varied independently while maintaining others at their nominal values to isolate its individual influence. For statistical reliability, each configuration was executed 30 independent times with different random seeds, and the mean and standard deviation of performance metrics were recorded.

Table 31 summarizes the parameter ranges used in the sensitivity analysis.

Table 32 presents the influence of PRM parameters for multi-robot settings. Increasing the number of nodes from 500 to 1000 improves the success rate and reduces energy consumption for all robot counts, but beyond 1000 nodes, computational time increases sharply with marginal benefits. The optimal connection range was found to be 1.0, balancing connectivity and efficiency.

Table 33 illustrates the effect of GA parameters, including population size, number of generations, crossover, and mutation rates. Results were averaged across 2–9 robots. The best performance was consistently observed for population size = 50–100, generations = 400, crossover = 0.8, and mutation = 0.05. These settings achieved near optimal fitness with reasonable computation time. Extremely low mutation rates led to premature convergence, while excessively high mutation introduced instability and slower convergence.

Across all tested configurations (2–9 robots), the hybrid PRM–GA framework maintained high success rates and consistent energy efficiency. The algorithm exhibited minimal sensitivity to moderate parameter variations, demonstrating robust and reliable performance. The optimal parameter set was identified as PRM nodes = 1000, connection range = 1.0, GA population = 50–100, generations = 400, crossover = 0.8, and mutation = 0.05. These values provide an effective balance between solution quality, computational cost, and scalability for multi-robot applications.

TABLE 31 | Parameter ranges used in the sensitivity analysis.

Parameter	Range tested
PRM node count	500, 750, 1000, 1250
PRM connection range	0.5, 1.0, 1.5
GA population size	20, 50, 100
GA generations	100, 200, 400, 800
Crossover probability	0.6, 0.8, 0.9
Mutation probability	0.01, 0.05, 0.10
Number of robots	2, 3, 5, 7, 9

TABLE 32 | Sensitivity results for PRM parameters (averaged over 2–9 robots).

PRM nodes	Success rate (%)	Energy (J)	Computation time (s)
500	82.7 ± 5.1	53.9 ± 2.0	8.7 ± 0.6
750	90.8 ± 3.9	49.8 ± 1.6	10.9 ± 0.8
1000	96.4 ± 2.3	47.2 ± 1.2	12.8 ± 0.7
1250	97.0 ± 2.1	46.9 ± 1.1	16.8 ± 1.0

5.2 | Computational Complexity

Table 34 presents the average run times recorded for different numbers of robots, illustrating how the computational complexity of the PRM–GA algorithm scales with increasing robot count. These times represent the mean duration required to complete

TABLE 33 | Sensitivity results for GA parameters (averaged over 2–9 robots).

Population	Generations	Crossover	Mutation	Success rate (%)	Energy (J)	Computation time (s)
20	100	0.6	0.01	78.2 ± 5.3	56.1 ± 2.4	5.5 ± 0.3
20	100	0.8	0.05	82.5 ± 4.9	54.8 ± 2.1	5.8 ± 0.4
20	200	0.8	0.05	84.9 ± 4.6	54.5 ± 2.1	7.5 ± 0.5
20	400	0.8	0.10	83.7 ± 4.8	54.9 ± 2.2	8.3 ± 0.5
50	100	0.6	0.01	88.1 ± 3.9	50.8 ± 1.9	7.8 ± 0.5
50	200	0.8	0.05	92.4 ± 3.1	48.5 ± 1.5	9.6 ± 0.6
50	400	0.8	0.05	95.6 ± 2.4	47.6 ± 1.3	11.6 ± 0.8
50	800	0.8	0.05	96.7 ± 1.9	46.6 ± 1.0	17.4 ± 1.1
100	100	0.6	0.01	90.3 ± 3.2	49.2 ± 1.4	10.1 ± 0.6
100	200	0.8	0.05	94.2 ± 2.7	47.8 ± 1.2	12.0 ± 0.8
100	400	0.8	0.05	96.2 ± 2.0	46.9 ± 1.2	13.1 ± 0.9
100	400	0.9	0.05	95.9 ± 2.1	47.0 ± 1.2	13.5 ± 0.9
100	400	0.8	0.10	94.7 ± 2.3	47.5 ± 1.3	13.4 ± 0.9

TABLE 34 | Average computational time for multi-robot path planning.

Number of robots	Average run time (s)
2	15.72
3	23.58
4	30.14
5	41.87
6	48.23
7	59.46
8	70.12
9	90.53

path planning after running each scenario 10 times. All other parameters were kept constant, so the observed trend reflects the direct impact of robot density on computational performance.

As Table 34 indicates, computational time increases nonlinearly with the number of robots. For smaller robot counts, the algorithm executes efficiently, with times below 24 s. However, as the number of robots grows, the computational burden rises sharply due to the added complexity of collision avoidance, path optimization, and multi-robot interactions.

This trend highlights that multi-robot path planning imposes significant computational overhead which can be identified as a drawback of the method. However, the performance can be further improved by optimizing algorithmic efficiency or employing parallel processing techniques for larger teams.

5.3 | Convergence Behavior Based on Generation Sensitivity

The effect of the generation budget (100–800 generations) was analyzed to study convergence. From the results shown from Table 33, the PRM–GA algorithm exhibited different amounts of total energy reduction when the generation count changes. Increasing the generations from 200 to 400 generations yields a substantial improvement in energy reduction (13%) and success rate (11%). But going from 400 to 800 generations results in only marginal gains (2%), confirming the claim of diminishing returns beyond 400 generations. This indicates that the algorithm effectively converges within 200–400 generations, beyond which further iterations provide minimal performance gains.

5.4 | Statistical Performance Analysis of the Proposed PRM–GA Approach

The performance of the proposed GA–PRM hybrid algorithm was evaluated against three standard path-planning benchmarks: PRM, A*, and RRT. To compare performances, paired *t*-tests were conducted on two key metrics: energy consumption and path completion time. These tests were used to determine whether the observed improvements achieved by PRM–GA are statistically significant across varying robot counts. Given the consistent nature of the simulation environment and the clear performance trends across different robot densities, the *t*-test provides a reliable measure of the performance gap between the GA–PRM hybrid and the baseline algorithms.

It should be noted that the robot movements in the simulation are independent and random. The initial positions of the robots were randomly generated, ensuring that the movements of the robots were independent.

1. Energy Consumption: A Paired *T*-Test (Two Sample for Means) was used at a 95% confidence level ($\alpha = 0.05$) to ensure that the energy savings of GA–PRM were statistically significant. Figure 26a shows how the means and variances of energy use change for scenarios with 2–9 robots. The mean is measured in J and the variance in J^2, and the graph is presented on a logarithmic scale. The comparative examination of energy consumption between

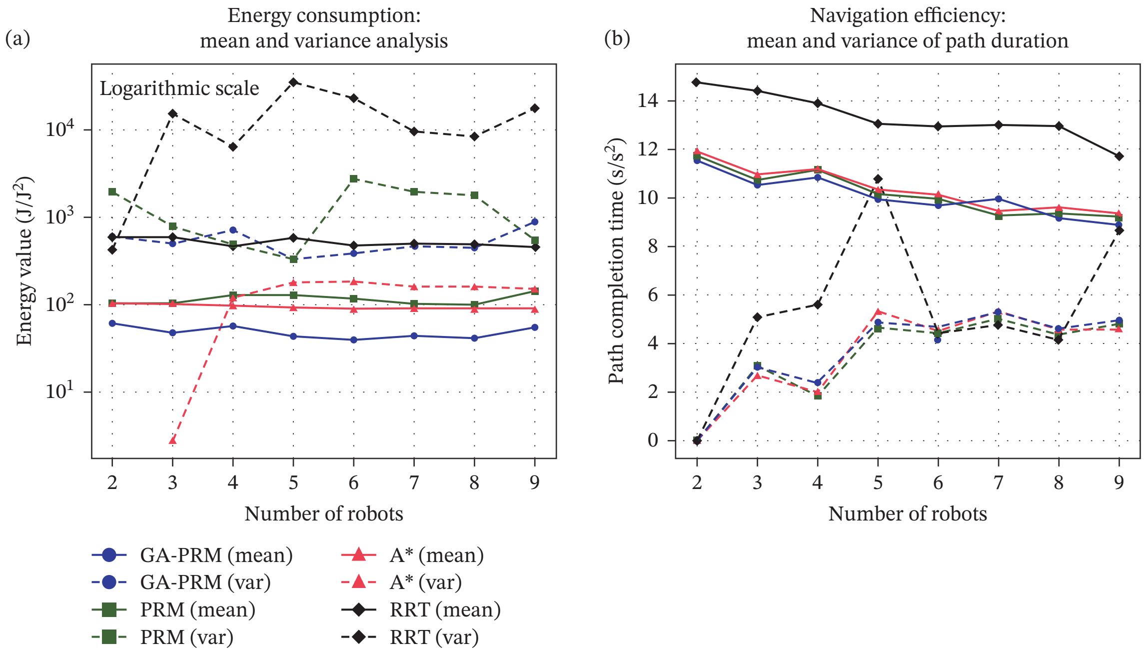

FIGURE 26 | Performance comparison for varying robot counts: (a) Energy consumption and (b) Path completion time.

the proposed GA–PRM hybrid and the benchmark algorithms, PRM, A^*, and RRT, demonstrate a steady trend of enhanced efficiency for the hybrid approach. The GA–PRM approach used the least amount of energy in all tests, which included 2–9 robots. In scenarios with only 2–3 robots, the performance differential is clear but not very wide. The GA–PRM, on the other hand, shows a significant scalability as the number of robots grows. At the 9-robot scenario, the average energy for GA–PRM is 55.02 J, while the average energy for PRM is 142.25 J and the average energy for RRT is 453.40 J. This suggests that normal PRM and RRT have trouble with the bigger search space and the need to avoid collisions when coordinating multiple robots. However, the GA layer improves the optimization of the roadmap to find the most energy-efficient paths. The variance metrics also show that these results are reliable. RRT was quite unstable, with a variance of 17,741.26 for 9 robots. This demonstrates that its path solutions are very inconsistent when it comes to energy cost. The A^* algorithm had the least variance since it was deterministic, but the GA–PRM had a competitive and stable variance profile (e.g., 328.20 for 5 robots) that was always lower than the normal PRM. This balance of low mean energy and controlled variance suggests that the GA–PRM hybrid is not only efficient but also robust enough for deployment in dynamic multi-robot environments where predictable energy usage is critical for task planning.

The paired t-test findings are shown in Table 35. At a significance level of $\alpha = 0.05$, the null hypothesis suggesting no difference between the mean energy of GA–PRM and other algorithms was evaluated against the alternative hypothesis that GA–PRM considerably decreases energy consumption. The RRT comparison results demonstrate that the $P\,(T \le t)$ two-tail value is always below 0.05 for all robot counts, with values as low as 1.19×10^{-5} for 9 robots. This, along with significant negative t-stat values (e.g., −181.98 for 2 robots), shows that the GA–PRM hybrid is inherently more efficient than the RRT technique. The statistical comparison with PRM and A^* shows how important the optimization layer of GA–PRM is in situations with a lot of data. The p-values show no significant difference at 2 robots ($p = 0.195$) when compared to PRM. However, as the number of robots increases,

TABLE 35 | Paired two-sample T-test for means: energy consumption (J).

No. of robots	Algorithm	Pearson corr.	Hyp. mean diff.	df	t Stat	$P(T_i = t)$ one-tail	t Crit one-tail	$P(T_i = t)$ two-tail
2	PRM	1.0000	0	1	−3.1559	0.0977	6.3138	0.1954
	A^*	NA	0	1	−2.3995	0.1257	6.3138	0.2514
	RRT	1.0000	0	1	−181.9840	0.0017	6.3138	0.0035
	PRM	0.6325	0	2	−4.2932	0.0251	2.9200	0.0502
3	A^*	0.4096	0	2	−4.2721	0.0253	2.9200	0.0507
	RRT	−0.0462	0	2	−7.3762	0.0089	2.9200	0.0179
	PRM	0.8495	0	3	−9.8789	0.0011	2.3534	0.0022
4	A^*	−0.9273	0	3	−2.1114	0.0626	2.3534	0.1252
	RRT	−0.6435	0	3	−8.1321	0.0019	2.3534	0.0039
	PRM	0.6526	0	4	−12.5379	0.0001	2.1318	0.0002
5	A^*	−0.8078	0	4	−3.5971	0.0114	2.1318	0.0228
	RRT	−0.4853	0	4	−6.1587	0.0018	2.1318	0.0035
	PRM	0.5723	0	5	−4.3068	0.0038	2.0150	0.0077
6	A^*	0.4554	0	5	−6.7315	0.0005	2.0150	0.0011
	RRT	0.6832	0	5	−7.6071	0.0003	2.0150	0.0006
	PRM	0.8113	0	6	−5.2239	0.0010	1.9432	0.0020
7	A^*	0.1571	0	6	−5.3096	0.0009	1.9432	0.0018
	RRT	0.2176	0	6	−12.5287	7.91×10^{-6}	1.9432	1.58×10^{-5}
	PRM	0.8206	0	7	−5.9159	0.0003	1.8946	0.0006
8	A^*	0.2639	0	7	−6.2059	0.0002	1.8946	0.0004
	RRT	0.2498	0	7	−14.3776	9.37×10^{-7}	1.8946	1.87×10^{-6}
	PRM	0.6152	0	8	−10.9191	2.19×10^{-6}	1.8595	4.39×10^{-6}
9	A^*	−0.4149	0	8	−2.8327	0.0110	1.8595	0.0221
	RRT	0.3754	0	8	−9.5526	5.97×10^{-6}	1.8595	1.19×10^{-5}

the p-value lowers to 0.050 at 3 robots and reaches a very substantial 4.39×10^{-6} at 9 robots. The GA–PRM also becomes statistically significant against A* starting with the 5-robot scenario ($p = 0.001$). These t-test results are crucial as they prove that the GA–PRM's performance advantage is not a result of random sampling luck but is a systematic improvement. The data shows that as the number of robots increases, the hybrid strategy becomes statistically necessary for reducing energy consumption.

2. Path Completion Time: The means and variances of path completion time, for different robot counts expressed in s and s2, respectively, are shown in Figure 26b. These findings provide a direct measure of path efficiency, in contrast to computational overhead. The descriptive statistics show that the PRM–GA hybrid consistently allows robots to finish their tasks in the shortest amount of time in all tested scenarios. The PRM–GA hybrid has a mean path time of 11.53 s in a 2-robot setup, while PRM, A*, and RRT have mean navigation times of 11.77, 11.90, and 14.78 s, respectively. With a mean completion time of 8.88 s, the PRM–GA hybrid continues to lead when the complexity increases to a 9-robot system, whereas benchmark algorithms need longer times of 9.21 s for PRM, 9.33 s for A*, and 11.73 s for RRT. These findings show that the evolutionary optimization layer in the PRM–GA hybrid outperforms grid-based and traditional sampling techniques in terms of output by actively optimizing the path shape to minimize path time rather than just identifying a valid path.

An additional indicator of the algorithms' dependability in generating effective routes is the variance in path completion time. The PRM–GA hybrid shows minimal and stable variance for lower robot counts (2–4 robots), such as 0.018 for the 2-robot example, demonstrating very predictable navigation performance. RRT, on the other hand, has greater volatility, indicating that although it is capable of finding paths, the subsequent path time and quality of those paths differ considerably in busy environments. The PRM–GA hybrid's stability implies that, in contrast to merely randomized sampling methods, the evolutionary optimization layer consistently converges on effective solutions.

A Paired T-Test (Two Sample for Means) was performed at a 95% confidence level ($\alpha = 0.05$) to assess the statistical significance of these gains in path efficiency. The findings are shown in Table 36.

TABLE 36 | Paired two-sample T-test for means: path completion time (s).

No. of robots	Algorithm	Pearson corr.	Hyp. mean diff.	df	t Stat	$P(T_i = t)$ one-tail	t Crit one-tail	$P(T_i = t)$ two-tail
2	PRM	1.0000	0	1	−3.6408	0.0853	6.3138	0.1707
	A*	NA	0	1	−3.8242	0.0814	6.3138	0.1628
	RRT	1.0000	0	1	−44.4060	0.0072	6.3138	0.0143
	PRM	0.9984	0	2	−3.2409	0.0417	2.9200	0.0835
3	A*	0.9988	0	2	−5.9530	0.0135	2.9200	0.0271
	RRT	0.9477	0	2	−8.2259	0.0072	2.9200	0.0145
	PRM	0.9965	0	3	−2.8713	0.0320	2.3534	0.0640
4	A*	0.9949	0	3	−3.6837	0.0173	2.3534	0.0347
	RRT	0.7876	0	3	−4.0927	0.0132	2.3534	0.0264
	PRM	0.9992	0	4	−4.8313	0.0042	2.1318	0.0084
5	A*	0.9979	0	4	−4.8731	0.0041	2.1318	0.0082
	RRT	0.9676	0	4	−5.4209	0.0028	2.1318	0.0056
	PRM	0.9922	0	5	−2.3292	0.0336	2.0150	0.0673
6	A*	0.9987	0	5	−9.3316	0.0001	2.0150	0.0002
	RRT	0.9922	0	5	−2.3292	0.0336	2.0150	0.0673
	PRM	0.9983	0	6	−3.3175	0.0080	1.9432	0.0161
7	A*	0.9972	0	6	−5.8698	0.0005	1.9432	0.0011
	RRT	0.9659	0	6	−16.6450	1.50×10^{-6}	1.9432	3.00×10^{-6}
	PRM	0.9983	0	7	−3.7049	0.0038	1.8946	0.0076
8	A*	0.9960	0	7	−6.2245	0.0002	1.8946	0.0004
	RRT	0.9663	0	7	−19.2460	1.30×10^{-7}	1.8946	2.50×10^{-7}
	PRM	0.9990	0	8	−9.7888	5.00×10^{-6}	1.8595	1.00×10^{-5}
9	A*	0.9919	0	8	−4.7448	0.0007	1.8595	0.0014
	RRT	0.8437	0	8	−5.3603	0.0003	1.8595	0.0007

With two-tail p-values as low as 2.5×10^{-7} in the 8-robot instance, the PRM–GA hybrid clearly outperforms RRT statistically in every situation. Strong proof that the PRM–GA hybrid is more time-efficient than the RRT approach is provided by these low p-values and strong negative t-stat values (e.g., −44.41 for 2 robots). Starting from the 3 robot scenario ($p = 0.027$) and continuing through 9 robots ($p = 0.00068$), the PRM–GA hybrid demonstrates statistically significant time savings against the conventionally effective A* algorithm, demonstrating that the GA optimization successfully refines the roadmap beyond the discrete constraints of a standard grid search. The comparison against standard PRM specifically highlights the advantages of the GA optimization layer. At very low densities (2–4 robots), the difference in path time is not statistically significant; however, it becomes significant at the 5-robot mark ($p = 0.0084$) and continues to be highly significant after that, reaching a p-value of 1.0×10^{-5} for the 8 robot configuration. The mean path completion time for the suggested hybrid approach is consistently shorter than that of the comparative benchmarks, as shown by the negative t-statistic values in every result. The PRM–GA hybrid is more robust and efficient at identifying high-quality, time-efficient pathways as the coordination task increases, as evidenced by the divergence in statistical significance at higher robot counts, despite the fact that the algorithms perform similarly in simpler cases. Consequently, it is an ideal choice for tasks that are difficult to complete and require a faster completion.

The statistical significance of these findings should be interpreted cautiously due to the sample sizes ($n = 2$–9) utilized for both the energy consumption and path planning time analyses. Although the paired t-tests offer insightful information, the conclusions may not be as reliable due to the limited sample sizes. Instead of being definitive, the observed variations in energy usage and path planning periods should be regarded as preliminary and suggestive. Larger sample sizes in future studies will be necessary to confirm and strengthen the validity of these conclusions. Only paired t-tests were used to assess statistical significance in this investigation. In order to increase the statistical results' dependability and interpretability, post-hoc tests and confidence intervals will be used in further research.

6 | Discussion and Future Work

The results clearly show the effectiveness of the proposed hybrid PRM–GA algorithm for multi-robot path planning across varying numbers of robots. By integrating the global exploration ability of PRM with the optimization capability of the GA, the proposed approach consistently obtains significant performance compared to PRM, A*, and RRT techniques.

In terms of energy consumption, across all robot configurations, the proposed hybrid method generates the most energy efficient paths than all the other three methods involved in the study. This improvement can be attributed to the GA's ability to optimize path selection and refine paths generated by PRM, thereby eliminating unnecessary turns and reducing overall travel cost.

Moreover, the proposed PRM–GA method indicates improved path completion times. Although A* demonstrates stable and predictable behavior in structured environments, its performance is limited by grid constraints. Similarly, RRT tends to generate longer and less optimal paths due to its random exploration nature, resulting in higher path completion times and energy consumption. The traditional PRM method performs better than RRT but lacks the optimization capability required for further improvement. In contrast, the proposed hybrid method effectively balances exploration and optimization, resulting in efficient and scalable performance. Furthermore, as the number of robots increases, PRM–GA approach indicates consistent scalability, maintaining lower energy consumption and lower path completion times compared to the other approaches. This proves that the hybrid approach is well suited for complex multi-robot scenarios where coordination and efficiency are essential.

Overall, the results validate that the proposed PRM–GA approach outperforms both classical (A*) and sampling-based (PRM, RRT) algorithms by achieving a better trade-off between energy efficiency and path execution time. This makes it a promising solution for real world multi-robot navigation problems, particularly in energy constrained environments.

6.1 | Future Directions

While the proposed PRM–GA approach demonstrates significant improvements in energy efficiency and path execution times, several directions can be explored to further enhance its applicability. Future work can focus on real world implementation using physical robot platforms to validate the practicality of the proposed hybrid approach under dynamic and uncertain conditions.

This approach can also be extended to incorporate dynamic obstacle avoidance, enabling robots to adapt in real time to changing environments. Integrating learning based techniques, such as reinforcement learning, could further improve adaptability and decision making capability in complex scenarios. Additionally, future research can explore scalability for large scale multi-robot systems, including communication constraints and decentralized coordination strategies. Incorporation of energy aware communication and task allocation mechanisms could further optimize overall system performance.

Further, the proposed algorithm can be improved by integrating hybrid optimization techniques or adaptive parameter tuning methods to improve convergence speed and robustness across diverse environments.

7 | Conclusion

In conclusion, the proposed PRM–GA algorithm demonstrates clear improvements in energy efficiency compared to traditional PRM, as well as A* and RRT. The results confirm that the hybrid approach consistently produces more energy efficient paths, making it suitable for real world robotic navigation tasks.

However, while the energy consumption is significantly reduced, the path time still requires further improvement. Although PRM–GA performs competitively with A* and better than RRT in most cases, achieving a balanced optimization between energy and time remains a challenge.

This study highlights the importance of designing an effective multi-objective fitness function that considers energy, path length, and collision avoidance. Future work will focus on dynamic path planning and adaptive optimization techniques to further enhance performance in complex environments.

Funding

No funding was received for this manuscript.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

1. 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* 9 (2021): 119310–119342.

2. H. Gao, D. Liu, and J. Hu, "A Survey on Path Planning for Mobile Robot Systems," in *2023 IEEE 12th Data Driven Control and Learning Systems Conference (DDCLS)*, (IEEE, 2023): 1176–1181.

3. S. J. Al-Kamil and R. Szabolcsi, "Optimizing Path Planning in Mobile Robot Systems Using Motion Capture Technology," *Results in Engineering* 22 (2024): 102043.

4. A. Meysami, S. Kelouwani, J.-C. Cuilliere, V. Francois, A. Amamou, and B. Allani, "An Efficient Indoor Large Map Global Path Planning for Robot Navigation," *Expert Systems with Applications* 248 (2024): 123388.

5. L. M. Dalbah, M. A. Al-Betar, M. A. Awadallah, and R. A. Zitar, "A Modified Coronavirus Herd Immunity Optimizer for Capacitated Vehicle Routing Problem," *Journal of King Saud University - Computer and Information Sciences* 34, no. 8 (2022): 4782–4795.

6. L. M. Dalbah, M. A. Al-Betar, and M. A. Awadallah, "Jaya Clustering-Based Algorithm for Multiobjective IoV Network Routing Optimization," *Soft Computing* 28, no. 6 (2024): 5639–5665.

7. P. K. Das and P. K. Jena, "Multi-Robot Path Planning Using Improved Particle Swarm Optimization Algorithm Through Novel Evolutionary Operators," *Applied Soft Computing* 92 (2020): 106312.

8. S. Tian, Y. Li, Y. Kang, and J. Xia, "Multi-Robot Path Planning in Wireless Sensor Networks Based on Jump Mechanism PSO and Safety Gap Obstacle Avoidance," *Future Generation Computer Systems* 118 (2021): 37–47.

9. L. Zhang, K. Cai, Z. Sun, et al., "Motion Planning for Robotics: A Review for Sampling- Based Planners," *Biomimetic Intelligence and Robotics* 5, no. 1 (2025): 100207.

10. A. Sanyal, M. N. Zafar, J. Mohanta, and M. F. Ahmed, "Path Planning Approaches for Mobile Robot Navigation in Various Environments: A Review," in *Advances in Interdisciplinary Engineering*, Lecture Notes in Mechanical Engineering, (Springer, 2021): 555–572.

11. A. Vagale, R. Oucheikh, R. T. Bye, O. L. Osen, and T. I. Fossen, "Path Planning and Collision Avoidance for Autonomous Surface Vehicles I: A Review," *Journal of Marine Science and Technology* 26, no. 4 (2021): 1292–1306.

12. E. W. Dijkstra, "A Note on Two Problems in Connexion With Graphs," in *Edsger Wybe Dijkstra: His Life, Work, and Legacy*, (Association for Computing Machinery, 2022): 287–290.

13. P. Hart, N. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transactions on Systems Science and Cybernetics* 4, no. 2 (1968): 100–107.

14. M. Hawa, "Light-Assisted A* Path Planning," *Engineering Applications of Artificial Intelligence* 26, no. 2 (2013): 888–898.

15. 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* 227 (2023): 120254.

16. T. XiangRong, Z. Yukun, and J. XinXin, "Improved A-Star Algorithm for Robot Path Planning in Static Environment," *Journal of Physics: Conference Series* 1792 (2021): 012067.

17. L. Zhang and Y. Li, "Mobile Robot Path Planning Algorithm Based on Improved a Star," *Journal of Physics: Conference Series* 1848 (2021): 012013.

18. W. Meng, W. Li, Y. Jin, H. Qi, and M. Yue, "Improved A* Algorithm Based on Node Expansion and Adaptive Fuzzy Controller," in *2021 27th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*, (IEEE, 2021): 78–83.

19. V. Yevsieiev, A. Abu-Jassar, S. Maksymova, and A. Alkhalaileh, "Route Constructing for a Mobile Robot Based on the D-Star Algorithm," (2024).

20. Y. Xue, X. Cai, and F. Neri, "A Multi-Objective Evolutionary Algorithm With Interval Based Initialization and Self-Adaptive Crossover Operator for Large-Scale Feature Selection in Classification," *Applied Soft Computing* 127 (2022): 109420.

21. Y. Zhou, W. Zhang, J. Kang, X. Zhang, and X. Wang, "A Problem-Specific Non-Dominated Sorting Genetic Algorithm for Supervised Feature Selection," *Information Sciences* 547 (2021): 841–859.

22. X. Yu, Z. Hu, X. Wang, and W. Luo, "Ranking Teaching–Learning-Based Optimization Algorithm to Estimate the Parameters of Solar Models," *Engineering Applications of Artificial Intelligence* 123 (2023): 106225.

23. S. Wang, W. Hu, I. Riego, and Y. Yu, "Improved Surrogate-Assisted Whale Optimization Algorithm for Fractional Chaotic Systems' Parameters Identification," *Engineering Applications of Artificial Intelligence* 110 (2022): 104685.

24. Y. Xue, Y. Wang, J. Liang, and A. Slowik, "A Self-Adaptive Mutation Neural Architecture Search Algorithm Based on Blocks," *IEEE Computational Intelligence Magazine* 16, no. 3 (2021): 67–78.

25. Y. Xue, Y. Tong, and F. Neri, "An Ensemble of Differential Evolution and Adam for Training Feed-Forward Neural Networks," *Information Sciences* 608 (2022): 453–471.

26. A. Tuncer and M. Yildirim, "Dynamic Path Planning of Mobile Robots With Improved Genetic Algorithm," *Computers & Electrical Engineering* 38, no. 6 (2012): 1564–1572.

27. M. Nazarahari, E. Khanmirza, and S. Doostie, "Multi-Objective Multi-Robot Path Planning in Continuous Environment Using an Enhanced Genetic Algorithm," *Expert Systems with Applications* 115 (2019): 106–120.

28. S. Agrawal, B. K. Patle, and S. Sanap, "A Systematic Review on Metaheuristic Approaches for Autonomous Path Planning of Unmanned Aerial Vehicles," *Drone Systems and Applications* 12 (2024): 1–28.

29. A. H. Hasan and A. M. Mosa, "Multi-Robot Path Planning Based on Max–Min ant Colony Optimization and D* Algorithms in a Dynamic Environment," in *2018 International Conference on Advanced Science and Engineering (ICOASE)*, (IEEE, 2018): 110–115.

30. C. Miao, G. Chen, C. Yan, and Y. Wu, "Path Planning Optimization of Indoor Mobile Robot Based on Adaptive Ant Colony Algorithm," *Computers & Industrial Engineering* 156 (2021): 107230.

31. N. H. Abbas and F. M. Ali, "Path Planning of an Autonomous Mobile Robot Using Directed Artificial Bee Colony Algorithm," *International Journal of Computer Applications* 96, no. 11 (2014): 11–16.

32. M. A. Contreras-Cruz, V. Ayala-Ramirez, and U. H. Hernandez-Belmonte, "Mobile Robot Path Planning Using Artificial Bee Colony and Evolutionary Programming," *Applied Soft Computing* 30 (2015): 319–328.

33. F. Xu, H. Li, C.-M. Pun, et al., "A New Global Best Guided Artificial Bee Colony Algorithm With Application in Robot Path Planning," *Applied Soft Computing* 88 (2020): 106037.

34. S. Aslan, "An Immune Plasma Algorithm With a Modified Treatment Schema for UCAV Path Planning," *Engineering Applications of Artificial Intelligence* 112 (2022): 104789.

35. D. Karaboga, et al., "An Idea Based on Honey Bee Swarm for Numerical Optimization," (2005).

36. P. Bhattacharjee, P. Rakshit, I. Goswami, A. Konar, and A. K. Nagar, "Multi-Robot Path-Planning Using Artificial Bee Colony Optimization Algorithm," in *2011 Third World Congress on Nature and Biologically Inspired Computing*, (IEEE, 2011): 219–224.

37. J.-H. Liang and C.-H. Lee, "Efficient Collision-Free Path-Planning of Multiple Mobile Robots System Using Efficient Artificial Bee Colony Algorithm," *Advances in Engineering Software* 79 (2015): 47–56.

38. Z. Wang, M. Li, L. Dou, Y. Li, Q. Zhao, and J. Li, "A Novel Multi-Objective Artificial Bee Colony Algorithm for Multi-Robot Path Planning," in *2015 IEEE International Conference on Information and Automation*, (IEEE, 2015): 481–486.

39. A. Nayyar, N. G. Nguyen, R. Kumari, and S. Kumar, "Robot Path Planning Using Modified Artificial Bee Colony Algorithm," in *Frontiers in Intelligent Computing: Theory and Applications*, 1014, (Springer, 2018): 25–36.

40. Y. V. Pehlivanoglu and P. Pehlivanoglu, "An Enhanced Genetic Algorithm for Path Planning of Autonomous UAV in Target Coverage Problems," *Applied Soft Computing* 112 (2021): 107796.

41. M. N. Ab Wahab, A. Nazir, A. Khalil, et al., "Improved Genetic Algorithm for Mobile Robot Path Planning in Static Environments," *Expert Systems with Applications* 249 (2024): 123762.

42. N. S. Abu, W. M. Bukhari, M. H. Adli, and A. Ma'arif, "Optimization of an Autonomous Mobile Robot Path Planning Based on Improved Genetic Algorithms," *Journal of Robotics and Control (JRC)* 4, no. 4 (2023): 557–571.

43. Z. Duraklı and V. Nabiyev, "A New Approach Based on Bezier Curves to Solve Path Planning Problems for Mobile Robots," *Journal of Computational Science* 58 (2022): 101540.

44. Y. Wu, M. Nie, X. Ma, Y. Guo, and X. Liu, "Co-Evolutionary Algorithm-Based Multi-Unmanned Aerial Vehicle Cooperative Path Plan- Ning," *Drones* 7, no. 10 (2023): 606.

45. A. G. Gad, "Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review," *Archives of Computational Methods in Engineering* 29, no. 5 (2022): 2531–2561.

46. J. Lu and Z. Zhang, "An Improved Simulated Annealing Particle Swarm Optimization Algorithm for Path Planning of Mobile Robots Using Mutation Particles," *Wireless Communications and Mobile Computing* 2021, no. 1 (2021): 2374712.

47. Q. Huang, Z. Sheng, Y. Fang, and J. Li, "A Simulated Annealing-Particle Swarm Optimization Algorithm for UAV Multi-Target Path Planning," in *2022 2nd International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, (IEEE, 2022): 906–910.

48. 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* 67 (2023): 101938.

49. Z. Ren, M. Wang, and Y. Hua, in *Proceedings of 2021 5th Chinese Conference on Swarm Intelligence and Cooperative Control*, (Springer, 2023).

50. A. G. Göktaş and S. Sezer, "A New Local Path Planning Approach by Synthesis of PRM and RRT* Algorithms for an Autonomous Mobile Robot," *Journal of Control, Automation and Electrical Systems* 36 (2025): 72–85.

51. Q. Li, Y. Xu, S. Bu, and J. Yang, "Smart Vehicle Path Planning Based on Modified PRM Algorithm," *Sensors* 22, no. 17 (2022): 6581.

52. B. Liao, F. Wan, Y. Hua, R. Ma, S. Zhu, and X. Qing, "F-RRT*: An Improved Path Planning Algorithm With Improved Initial Solution and Convergence Rate," *Expert Systems with Applications* 184 (2021): 115457.

53. J. Ding, Y. Zhou, X. Huang, K. Song, S. Lu, and L. Wang, "An Improved RRT* Algorithm for Robot Path Planning Based on Path Expan- Sion Heuristic Sampling," *Journal of Computational Science* 67 (2023): 101937.

54. S. Sabeeh and I. S. Al-Furati, "Efficient Path Planning in Medical Environments: Integrating Genetic Algorithm and Probabilistic Roadmap (GA-PRM) for Autonomous Robotics," *Iraqi Journal for Electrical and Electronic Engineering* 20, no. 2 (2024): 243–258.

55. 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* 2, no. 2 (2023): 12–37.

56. S. Zhang, Q. Hou, X. Zhang, X. Wu, and H. Wang, "A Novel Vectorized Curved Road Representation Based Aerial Guided Unmanned Vehicle Trajectory Planning," *Sensors* 23, no. 16 (2023): 7305.

57. T. Jathunga and S. Rajapaksha, "Improved Path Planning for Multi-Robot Systems Using a Hybrid Probabilistic Roadmap and Genetic Algorithm Approach," *Journal of Robotics and Control (JRC)* 6, no. 2 (2025): 715–733.

58. R.-C. Roman, R.-E. Precup, E. M. Petriu, and A.-I. Borlea, "Hybrid Data-Driven Active Disturbance Rejection Sliding Mode Control With Tower Crane Systems Validation," *Romanian Journal of Information Science and Technology* 27, no. 1 (2024): 50–64.

59. R. Chi, H. Li, D. Shen, Z. Hou, and B. Huang, "Enhanced P-Type Control: Indirect Adaptive Learning From Set-Point Updates," *IEEE Transactions on Automatic Control* 68, no. 3 (2023): 1600–1613.