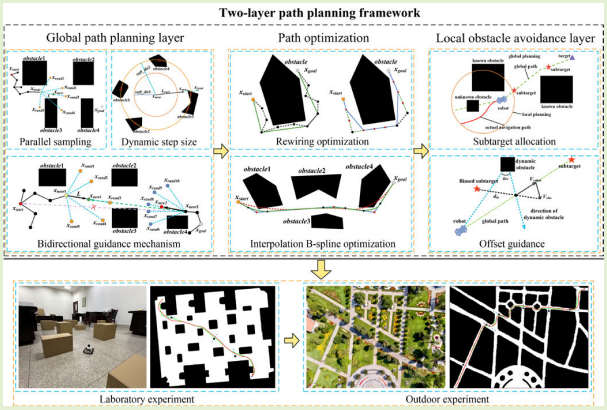
A Path Planning Framework for Robots Based on Improved Parallel Sampling RRT and Offset Guidance DWA

Yaowei Hu, Xufei Chen, Pingping Tang, *Member, IEEE*, Hui Zhang, *Member, IEEE*, Jiong Jin, *Member, IEEE*, and Shiwen Mao, *Fellow, IEEE*

***Abstract*—The path planning is one of the critical technologies for robots to achieve the autonomous operation, enabling to quickly find a safe path in dynamic environments. However, relying on global path planning alone cannot avoid dynamic obstacles, while only using the local path planning may lead to falling into local minima. Therefore, a two-layer robot path planning method suitable for dynamic environments is proposed. This two-layer strategy consists of an efficient global path planning layer and a safe local dynamic obstacle avoidance layer. In the first layer, a parallel sampling and bidirectional guidance rapidly exploring random tree algorithm (PB-RRT) is proposed to search for the global path. To enhance efficiency, the parallel heuristic sampling is introduced to replace the random sampling in bidirectional rapidly exploring random tree (Bi-RRT), and an evaluation function incorporating distance and corner factors is designed to select optimal sampling nodes for adaptive expansion, making the sampling process directional and avoiding over-exploration of space. A bidirectional guidance mechanism further accelerates the merging of the two trees by fully utilizing newly generated node information. Then, a path optimization (PO) method is proposed to improve the length and smoothness of the initial path and obtain the key nodes of the path. In the second layer, the key nodes obtained from the first layer are used as dynamic subtargets, and the safe dynamic window approach (SDWA) is used to achieve the dynamic obstacle avoidance. To further enhance safety, an offset guidance method is proposed to flexibly steer the robot around dynamic obstacles. Extensive experiments show that the average planning time of PB-rapidly exploring random tree (RRT) is reduced by 67.7% compared with Bi-RRT, while the path quality is improved and the path length is reduced by 27.1%. The proposed method also effectively avoids dynamic obstacles in environments with obstacle densities exceeding 60% and achieves the maximum of minimum distance to dynamic obstacles, validating the feasibility and safety of the method.**



***Index Terms*—Dynamic path planning, parallel sampling, path optimization (PO), rapidly exploring random tree (RRT).**

I. Introduction

PATH planning, as both a key technology and a major challenge for mobile robots, serves as the foundation for intelligent operations [1]. It not only ensures the safe obstacle avoidance and efficient navigation in complex and dynamic environments but also acts as a core enabling technology in fields such as industrial automation and autonomous driving [2]. To allow mobile robots to operate safely and effectively, it is essential to generate a collision-free path from the initial state to the goal state within the workspace, subject to given constraints [3]. Depending on whether environmental information are fully known, the path planning problem can be classified into global and local planning [4]. Similarly, based on the nature of the obstacles, it can be divided into static and dynamic scenarios [5]. Relying solely on global path planning algorithms, such as the A* algorithm [6], ant colony optimization (ACO) algorithm [7], and rapidly exploring random trees (RRT) [8], may result in failure when dynamic obstacles are present [3], [9]. Conversely, using only local planning algorithms, such as the artificial potential field (APF) [10] and the dynamic window approach (DWA) [11],

Received 13 May 2025; revised 2 July 2025; accepted 3 August 2025. Date of publication 12 August 2025; date of current version 15 September 2025. This work was supported in part by NSF under Grant CNS-2415208, in part by NSFC under Grant 62071005 and Grant 62071252, in part by the National Key Research and Development Program of China under Grant 2020YFB2104004, in part by the Key Research and Development Program of Jiangsu Province under Grant BE2021725, in part by the Natural Science Foundation of Anhui Province under Grant 2308085Y02 and Grant 2208085MF155, in part by the Natural Science Foundation of the Higher Education Institutions of Anhui Province under Grant KJ2021A0124, and in part by the Provincial Quality Engineering for Education of Anhui Province under Grant 2024cxcysj027. The associate editor coordinating the review of this article and approving it for publication was Prof. Fuqiang Gu. *(Corresponding author: Pingping Tang.)*

Please see the Acknowledgment section of this article for the author affiliations.

Digital Object Identifier 10.1109/JSEN.2025.3596346

can lead to local minima issues in complex environments, preventing the robot from reaching the goal [5], [12], [13].

Sampling-based algorithms, such as the RRT algorithm, are among the most commonly used methods in the global path planning [14]. This method avoids the geometric modeling of the space and has been shown to guarantee the probabilistic completeness [15]. However, the randomness and uncertainty of sampling lead to numerous invalid sampling nodes, preventing the path from being asymptotically optimal [16]. To balance the path quality and convergence speed, researchers have conducted extensive studies. Gammell et al. [17] replaced the global uniform sampling with sampling within an elliptical space to restrict the sampling range, enabling the path to quickly converge to the asymptotically optimal solution. Kuffner and LaValle [18] proposed the bidirectional RRT (Bi-RRT) algorithm, which grows two random trees from the starting and endpoints, reducing the path planning time. Wang et al. [19] proposed a greedy expansion strategy aimed at dynamic nodes, which incorporates an initial node rejection mechanism and sampling space constraints during the random sampling process, effectively reducing the backward sampling and improving the speed and efficiency of sampling. Huang and Zhang [20] integrated the bidirectional RRT with convergence points between the starting and endpoints to guide tree growth. Li and Chen [21] combined the RRT-Connect and RRV algorithms, dividing the state space into three regions for local sampling, which increases the probability of exploring unexplored areas and accelerates space exploration. Salzman and Halperin [22] integrated the RRT and RRT* algorithms, allowing a switch between the RRT and RRT* algorithms by introducing an approximation factor to control the state. Wang et al. [23] introduced the B2U-RRT algorithm, which uses bidirectional search to find heuristic paths, with the end tree guiding the starting tree toward the goal to generate the final path. Jiang et al. [24] replaced random sampling with mixed constraint sampling, positioning the sampled nodes closer to passable areas between obstacles. Zhang et al. [25] proposed the IBi-RRT algorithm, which adopts the idea of random node priority selection to reduce randomness in expansion, and uses an adjustable attraction size strategy to improve search speed. Hu et al. [26] proposed a path deformation strategy to distance path points from collision points, generating paths without collisions and smoothing the path with a steering function. Chen et al. [27] proposed the DT-RRT* algorithm, utilizing the RRT algorithm to explore the space, building another tree to optimize the path once a feasible solution is found, and adding a Re-SearchParent process to enhance the optimization efficiency.

The global path planning can find a path from the start to the endpoint, but it often fails to avoid locally dynamic obstacles [6]. Therefore, more efficient and real-time local path planning methods are needed in dynamic environments [28]. Qi et al. [29] proposed a path replanning method base on the Pareto theory is introduced to search the state tree for an alternative node to avoid dynamic obstacles. Szczepanski [30] introduced the safe APF for local path planning, which uses an attractive potential field for target attainment and replaces the traditional repulsive potential field with a general obstacle field. Dobrevski and Skočaj [31] proposed an adaptive DWA combined with deep reinforcement learning, where neural networks predict parameters in the cost function to adapt to specific environments. Yasuda et al. [32] introduced an efficient DWA that addresses navigation challenges for robots in uncertain environments using deterministic sampling. Nevertheless, the above algorithms cannot fully solve the problem of easily falling into local optima in complex dynamic environments.

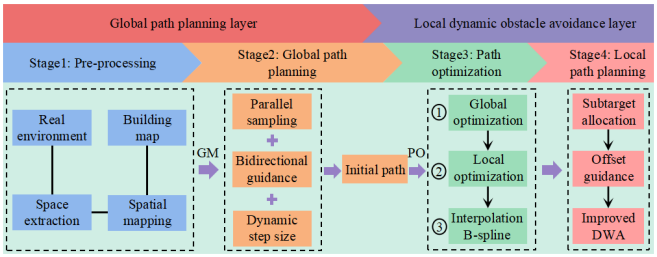


Fig. 1. Method framework diagram. The two-layer path planning method comprises a global path planning layer and a local dynamic obstacle avoidance layer. The global planning layer first preprocesses the environment to generate a global map (GM). Based on this map, an initial path is generated by the global planner and subsequently refined through PO to enhance its quality. Key nodes are then extracted from the optimized path. In the local dynamic obstacle avoidance layer, these key nodes serve as dynamic subtargets for the safe DWA (SDWA). In addition, an offset guidance method is applied to further improve the efficiency of the dynamic obstacle avoidance.

To address the limitations of global path planning in dynamic environments and the lack of guidance in local path planning, which can lead to local minima in complex environments, this article proposes a two-layer robot path planning method. The framework of the two-layer method is shown in Fig. 1. The parallel sampling and bidirectional guidance RRT (PB-RRT) algorithm is proposed to quickly generate the global path, with key nodes from the optimized path extracted as dynamic subtargets. An SDWA with an offset guidance method is then introduced for local path planning to avoid dynamic obstacles. The main contributions of this article are as follows.

1) Based on the two-layer method, the PB-RRT algorithm and path optimization (PO) method are used to quickly identify the global path, followed by the SDWA algorithm for real-time local obstacle avoidance.
2) In the global path planning layer, the RRT sampling method is improved. Proposing the parallel heuristic sampling instead of the original random sampling to make the sampling process directional and avoid excessive exploration of the space. The evaluation function, incorporating distance and angle factors is established, and adaptive step sizes are dynamically selected based on the surrounding environment, providing directionality to the sampling process. Finally, the bidirectional guidance mechanism is introduced to fully utilize information from newly generated nodes to accelerate the merging process of two trees and quickly obtain the initial path.

3) The initial path is optimized by the PO method, and an interpolation B-spline path smoother is proposed to achieve a higher degree of fitting, resulting in a higher quality global path.
4) In the local path planning layer, the SDWA algorithm is proposed to solve the inherent problems of the traditional DWA algorithm. The local area is replanned based on the first-layer dynamic subtargets and offset guidance method to avoid dynamic obstacles.

The remainder of this article is organized as follows. Section II introduces environmental modeling and the path planning problem. Section III presents the improvement strategies and probabilistic completeness proof of the PB-RRT algorithm. Section IV introduces the detailed optimization of SDWA algorithm. Section V introduces the two-layer path planning method. Section VI, we present simulation experiments to verify the algorithm and architecture proposed in this article. The conclusion is presented in Section VII.

II. Problem Statements

Different from static path planning, dynamic path planning primarily involves finding an effective path for a robot in an ever-changing environment. The map contains dynamic obstacles, and as the robot moves, it continuously gathers information about its surroundings through sensors, detecting both static and dynamic obstacles. The goal is to plan a path from the point, where a conflict with obstacles arises to a local target point [33].

To clearly express the dynamic path planning problem, we abstract the working environment into a configuration space. The configuration space for the path planning problem is defined as $X \in R^n$, the obstacle space is defined as X_{obs}, and the free space is defined as $X_{\text{free}} = X_{\text{obs}}/X$. Initially, in the first layer of the global path planning process, an initial path is found within the known free space σ. Therefore, the global path planning problem can be defined as follows:

$$\begin{aligned} & t : [0,T] \rightarrow \sigma \\ & \text{s.t. } \sigma\,(0) = x_{\text{start}} \\ & \sigma\,(T) = x_{\text{goal}} \\ & \sigma\,(t) \in X_{\text{free}}\ \forall t \in [0,T] \end{aligned} \tag{1}$$

where x_{start} is the starting node in the state space, x_{goal} is the goal node, t is the time stamp, $\sigma(0)$ is the initial state of the robot, $\sigma(T)$ is the final state, and $\sigma(t)$ is the state at any moment. After the global path planning is completed, the critical nodes from the generated path are extracted sequentially and used as subtargets to guide the robot for local path planning. In the second layer of the local path planning process, the obstacle space X_{obs} includes both static and dynamic obstacles. The free space X_{free} and obstacle space X_{obs} are updated through real-time obstacle detection. When the planned global path conflicts with an obstacle, local replanning is triggered to generate a new path such that $\sigma(t) \in X_{\text{free}}$, where any $t \in [0, T]$. As the robot reaches a subtarget node, it selects the next one as the new local target. This process continues until the robot arrives at the goal node.

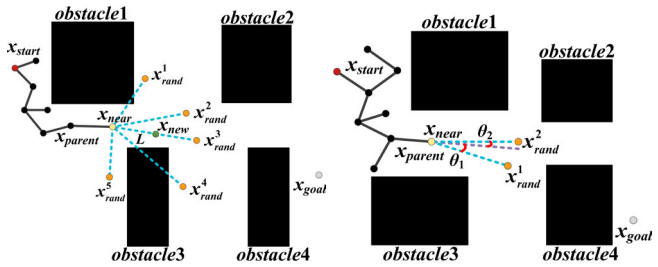


Fig. 2. Parallel heuristic sampling strategy. x^i_{rand} is the sampling node, x_{near} is the nearest node, x_{parent} is the parent node, and θ_i is the angle between three adjacent nodes x_{parent}, x_{near}, and x^i_{rand}. (a) Parallel sampling. (b) Extended node selection.

III. Path Planning Based on the PB-RRT Algorithm

A. Parallel Heuristic Sampling

The RRT algorithm only randomly samples one node per iteration, which limits the efficiency of the algorithm and the quality of the sampling nodes. To improve the sampling efficiency, a parallel heuristic sampling strategy is proposed. This approach entails randomly sampling multiple nodes during each iteration, and then, the sampling nodes are evaluated using an evaluation function to identify the optimal node for expansion. Fig. 2(a) illustrates the parallel sampling of five nodes, where the evaluation function is computed for each sampled node, and the node with the lowest evaluation value is selected for expansion.

The evaluation function $F(x)$ in Algorithm 1 calculates the path length from the sampled node to both the starting point and the target point, facilitating more rapid growth of the random tree toward the target. In addition, it computes the angles between three adjacent path nodes to minimize unnecessary turns created during sampling, thereby reducing path costs and enhancing path quality. Since distance and angle are metrics from different ranges, normalization is necessary to unify these distinct data sources into a common reference frame. The normalization method is as follows:

$$T_n\left(\theta_{x_{\text{parent}},x_{\text{near}},x^i_{\text{rand}}}\right) = S_A\left(\theta_{x_{\text{parent}},x_{\text{near}},x^i_{\text{rand}}}\right) \tag{2}$$

$$A\left(x^i_{\text{rand}},x_{\text{start}}\right) = S_d\left(d_{x^i_{\text{rand}},x_{\text{start}}}\right) \tag{3}$$

$$N\left(x^i_{\text{rand}},x_{\text{goal}}\right) = S_D\left(D_{x^i_{\text{rand}},x_{\text{goal}}}\right) \tag{4}$$

$$S_A\left(\theta_{x_{\text{parent}},x_{\text{near}},x^i_{\text{rand}}}\right) = \frac{\theta_{x_{\text{parent}},x_{\text{near}},x^i_{\text{rand}}}}{\sum_{i=1}^{n}\theta_{x_{\text{parent}},x_{\text{near}},x^i_{\text{rand}}}} \tag{5}$$

$$S_d\left(d_{x^i_{\text{rand}},x_{\text{start}}}\right) = \frac{d_{x^i_{\text{rand}},x_{\text{start}}}}{\sum_{i=1}^{n} d_{x^i_{\text{rand}},x_{\text{start}}}} \tag{6}$$

$$S_D\left(D_{x^i_{\text{rand}},x_{\text{goal}}}\right) = \frac{D_{x^i_{\text{rand}},x_{\text{goal}}}}{\sum_{i=1}^{n} D_{x^i_{\text{rand}},x_{\text{goal}}}} \tag{7}$$

where x^i_{rand} is the sampling node generated for each iteration, $d_{x^i_{\text{rand}},x_{\text{start}}}$ is the Euclidean distance between points x^i_{rand} and x_{start}, while $D_{x^i_{\text{rand}},x_{\text{goal}}}$ is the distance between points x^i_{rand} and x_{goal}. $S_d(d_{x^i_{\text{rand}},x_{\text{start}}})$ and $S_D(D_{x^i_{\text{rand}},x_{\text{goal}}})$ are

Algorithm 1 Parallelsampling

for $i = 1$ *to* n **do**
  $x^i_{rand} \leftarrow$ RandomSampling(Map);
  $x_{nearest} \leftarrow$ Nearest(x_{rand}, T);
  dis_goal $\leftarrow$ Distance between x^i_{rand} and x_{goal};
  dis_start $\leftarrow$ Distance between x^i_{rand} and x_{start};
  angle $\leftarrow$ CulculateAngle(x_{parent}, $x_{nearest}$, x^i_{rand});
  $A(x)$, $N(x)$, $T_n \leftarrow$
   Normalize $d_{x^i_{rand},x_{start}}$, $D_{x^i_{rand},x_{goal}}$
   and $\theta_{x_{parent},x_{near},x^i_{rand}}$;
$x_{best} \leftarrow$ Calculate the value of $F(x)$ and select the best node with the minimum value of $F(x)$ in x^i_{rand};
$x_{new} \leftarrow$ Extend(T,x_{best},L_{step});

the normalization functions corresponding to these distances. The variable $\theta_{x_{\text{parent}},x_{\text{near}},x^i_{\text{rand}}}$ is the angle between adjacent nodes, and $S_A(\theta_{x_{\text{parent}},x_{\text{near}},x^i_{\text{rand}}})$ is the normalization function for the angle.

When selecting the optimal sampling node, both distance and angle constraints are considered. Adaptive factors α and k are utilized to adjust the influence of the distance and the turning angle on the evaluation function. The evaluation function $F(x)$ is defined as follows:

$$F(x) = \alpha \cdot A(x) + (1 + d_n/D_n) \cdot N(x) + k \cdot T_n \tag{8}$$

$$A(x) = \left\| x^i_{\text{rand}} - x_{\text{start}} \right\|, \; N(x) = \left\| x_{\text{goal}} - x^i_{\text{rand}} \right\| \tag{9}$$

$$T_n = \theta\left(x_{\text{parent}},x_{\text{near}},x^i_{\text{rand}}\right) \tag{10}$$

where D_n is the Euclidean distance from the starting point to the target point, d_n is the Euclidean distance from the current node to the target point, T_n is the angle between three adjacent nodes, α is the distance factor, and k is the turning angle factor. As shown in Fig. 2(b), the Euclidean distance between points x^2_{rand} and x_{goal}, x_{start} is greater than the distance between points x^1_{rand} and x_{goal}, x_{start}, but the angle θ_2 between the adjacent path segment, is smaller than θ_1, Therefore, by considering both the distance and angle costs, the path is generated between points x^2_{rand} and x_{near}.

B. Dynamic Step Size

The fixed step size extension in the traditional RRT algorithm generates many unnecessary nodes in simple environments, while extensive sampling through obstacle areas in complex environments significantly decreases search efficiency. Therefore, this article proposes a dynamic step size extension strategy to enhance the search efficiency and path quality in various environments. The method detects the distribution of obstacles around the extended nodes and dynamically selects the step length accordingly. As shown in Fig. 3, the basic approach involves drawing a circle centered at the nearest node $x_{\text{nearest}}(x_0,y_0)$, with a safety distance $R_{\text{safe\_dis}}$ as the radius, and extracting points from different regions of the circle. The adaptive circular domain is set as

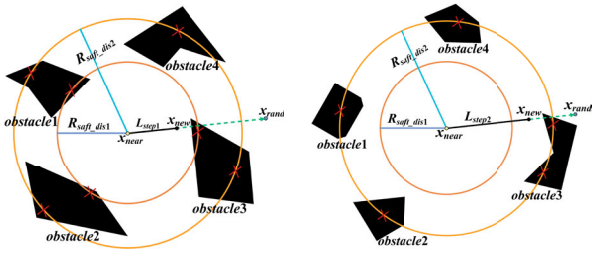


Fig. 3. Dynamic step size extension strategy. (a) Expansion step size in the dense obstacle environment. (b) Expansion step size in the sparse obstacle environment.

follows:

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + R_{\text{safe\_dis}} \begin{bmatrix} \cos\theta_i \\ \sin\theta_i \end{bmatrix}$$
$$\text{s.t. } n = \text{ceil}\left(2\pi \cdot R_{\text{safe\_dis}}\right)$$
$$\theta_i = (i/n) \cdot 2\pi, i = 1, \ldots, n \tag{11}$$

where $R_{\text{safe\_dis}}$ represents the safety distance. $R_{\text{safe\_dis1}}$ and $R_{\text{safe\_dis2}}$ are defined as two safety distances, whose values are specified as follows:

$$R_{\text{safe\_dis1}} = L_{\text{step1}} + 0.5L_{\text{step2}} \tag{12}$$

$$R_{\text{safe\_dis2}} = L_{\text{step3}} + 0.5L_{\text{step2}} \tag{13}$$

where L_{step1} is the small expansion step size, L_{step2} is the normal step size, and L_{step3} is the large step size. As shown in Fig. 3, first, $R_{\text{safe\_dis2}}$ is used as the safety distance, and verifies whether the points within the circular area lie outside the state space or on an obstacle. If no obstacles are detected, the large step size L_{step3} is selected for expansion. Otherwise, $R_{\text{safe\_dis1}}$ is used as the safety distance, and check the points within the circular area again. The normal step size L_{step2} or small step size L_{step1} is chosen for expansion.

C. Bidirectional Guidance Mechanism

Different from the random tree growth method of the traditional bidirectional RRT (Bi-RRT) algorithm, a bidirectional guidance mechanism is proposed that simultaneously establishes two random exploration trees at the starting and target points. The starting tree grows toward the target point and provides a heuristic guidance for the target tree, guiding it to grow toward the starting tree, enabling the rapid merging of the two trees. As shown in Fig. 4, the starting tree node x^1_{new} is generated using the parallel heuristic sampling strategy, while the target tree node x^2_{new} is generated using the newly generated path node x^1_{new}, as a local target point through the parallel heuristic sampling strategy. The evaluation function $H(x)$ of the target tree for selecting the best extension node can be defined as follows:

$$H(x) = \varepsilon \cdot B(x) + (1 + l_n/L_n) \cdot S(x) + m \cdot T_n \tag{14}$$

$$B(x) = \left\| x^i_{\text{rand}} - x_{\text{goal}} \right\|, \; S(x) = \left\| x^{T1}_{\text{new}} - x^i_{\text{rand}} \right\| \tag{15}$$

where L_n is the Euclidean distance from the target point to the new node of starting tree x^{T1}_{new}, l_n is the Euclidean distance from the current node to x^{T1}_{new}, ε is the distance factor, and m is

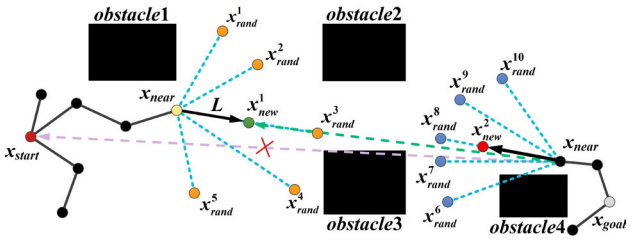


Fig. 4. New node based on the bidirectional guidance mechanism. The purple dashed line indicates the tree expansion direction of the traditional Bi-RRT algorithm, and the green dashed line shows the tree expansion direction with the bidirectional tree guidance mechanism.

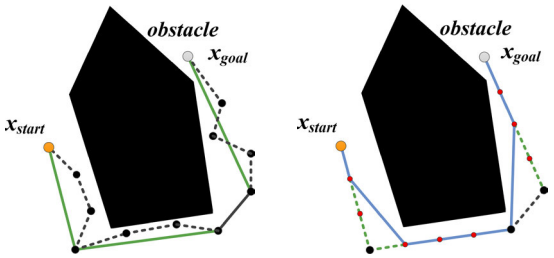


Fig. 5. Rewiring PO. (a) Global optimization. The black path represents the initial path, and the green path represents the global rewiring path. (b) Local optimization. The red nodes represent the path nodes inserted by the algorithm, and the blue path indicates the local rewiring path.

the turning angle factor. By shifting the target of the expansion of the target tree from the starting point to the newly generated node of the starting tree, the bidirectional tree connection is completed more quickly.

D. Path Optimization

This article proposes a PO method to filter out noncritical nodes and improve the quality of the initial path. First, the initial path is traced back from the target point, with the target point serving as the starting node of the segment. Subsequent path nodes are sequentially connected, and the algorithm checks whether the connecting segment collides with any obstacles. If no collision is detected, all path nodes between the two connected nodes are removed. If a collision occurs, the child node of the current node is taken as the new starting point of the segment, and the process is repeated until the path is fully traced back to the starting point. The initial path globally rewired process is shown in Fig. 5(a).

After global rewiring, the number of path nodes is significantly reduced, but the path still consists of segments from the initial path. As shown in Fig. 5(b), local rewiring is then introduced to further enhance the path quality. Since the path nodes are relatively sparse after global rewiring, interpolation is proposed for the newly generated path segments, with the interpolation distance set to the minimum step size. This creates a denser set of nodes for local rewiring. Starting from the initial point, the method similar to the global rewiring is used to further reduce the path cost and resolve issues of overlapping paths.

The B-spline curve is introduced to optimize the path and generate a smooth path. However, after local rewiring, the limited number of path nodes results in poor alignment between the smoothed B-spline path and the original path, increasing collision risks. Therefore, we propose an interpolation B-spline smoother for better path fitting. If the segment length between adjacent nodes is less than a threshold, a control point is inserted at the midpoint of the segment. Otherwise, control points are generated at a regular step size from both nodes. As shown in Fig. 6, the path smoothed by the interpolation B-spline achieves a closer fit to the original path compared with the conventional B-spline.

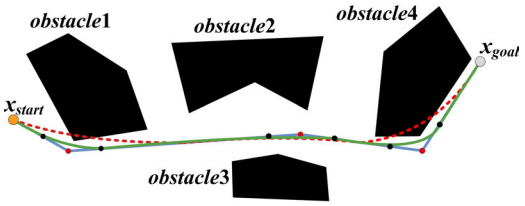


Fig. 6. B-spline curve optimization. The red path represents the result of conventional B-spline curve optimization, and the green path shows the result of interpolation B-spline curve.

E. Theoretical Analysis of Algorithm

The probabilistic completeness of an algorithm must meet the following condition: if a feasible path exists from the starting point to the goal point, the probability of the algorithm finding this feasible path approaches one as the number of random samples tends to infinity [34]. The PB-RRT algorithm is probabilistically complete. For any path planning problem $\{x_{\text{start}}, X_{\text{free}}, X_{\text{obs}}, x_{\text{goal}}\}$ with an existing feasible solution σ, as the number of samples approaches infinity, the probability of the PB-RRT algorithm finding a feasible path approaches one, as described in

$$\lim_{i\to\infty} P\left(\left\{x_{\text{start}}, x_{\text{goal}} \in V_i^{\text{PB-RRT}} \cap \sigma \text{ subject to } \sigma \subseteq T_i^{\text{PB-RRT}}\right\}\right) = 1 \tag{16}$$

where i is the number of iterations, $T_i^{\text{PB-RRT}}$ is the random tree generated by the i times sampling, $V_i^{\text{PB-RRT}}$ is the set of nodes in the random tree, and P is the probability of the PB-RRT algorithm finding a feasible path.

For the PB-RRT algorithm, the parallel heuristic sampling strategy improves traditional random sampling by incorporating parallel sampling points, which guide the random tree to grow toward the target point and reduce invalid sampling. This improvement does not compromise the convergence of the algorithm and ensures that it can at least find the same path solution as the RRT algorithm. The bidirectional guidance mechanism accelerates the merging process of two trees without affecting probabilistic completeness. Since the probabilistic completeness of both the RRT and Bi-RRT algorithms has been established [22], it proves that the PB-RRT algorithm is probabilistically complete. Experimental results demonstrate that as the complexity of the map increases, the number of samples required by the PB-RRT algorithm to find a feasible path also increases, but remains lower than that required by the Bi-RRT algorithm.

For the path planning problem, the time complexity of the PB-RRT algorithm is $O(n\log(n))$ and the space complexity is $O(n\log(n))$ when the number of iterations is n.

The overall algorithmic complexity depends on random sampling, nearest node search, and collision detection. The parallel sampling strategy samples multiple nodes per iteration for evaluation function computation, and its time complexity is $O(n)$. The time cost of the dynamic step size expansion method also depends on nearest node search and collision detection; hence, its complexity is $O(n\log(n))$. For the bidirectional guidance mechanism, the sampling and expansion processes for both the starting and goal trees are the same as those in the single tree structure, so the computational complexity of the initial path planning phase remains $O(n\log(n))$. For PO, the time complexity of reselecting path nodes, detecting connections, and inserting control points for B-spline smoothing is less than or equal to $O(n)$. Therefore, the overall time complexity of the algorithm remains $O(n\log(n))$. Regarding space complexity, storing random tree nodes requires $O(n)$ space, and storing nearby nodes for each new node requires $O(n\log(n))$ space; thus, the space complexity of the algorithm is $O(n\log(n))$ according to the large O rule.

Therefore, the PB-RRT algorithm exhibits the same theoretical complexity as the RRT algorithm. Experimental results further demonstrate that the PB-RRT algorithm achieves faster convergence compared with Bi-RRT algorithms, while also requiring fewer stored path nodes.

IV. Automatic Obstacle Avoidance Based on the SDWA Algorithm

The traditional DWA has limitations such as a conservative velocity window and an ineffective evaluation function mechanism. These shortcomings make the robot prone to falling into local optima and lead to poor robustness in complex environments. To address these issues, this article proposes SDWA. As shown in Fig. 7, the adaptive dynamic window dynamically adjusts the exploration window size during the movement of robots. All feasible regions δ are first identified, and the predicted path region δ_s, which is closest to the dynamic subtargets, is selected as the optimal feasible region. The robot evaluates various path options by balancing multiple evaluation factors in the path evaluation function within this optimal feasible region. This process enables the robot to determine the optimal movement direction effectively. The path evaluation function is defined as follows:

$$G(v,\omega) = k_1\cdot \mathrm{Head}(v,\omega) + k_2\cdot \mathrm{Dist}(v,\omega) + k_3\cdot \mathrm{Vel}(v,\omega) + k_4\cdot \mathrm{Dist_{GP}}(v,\omega) + k_5\cdot \mathrm{Goal}(v,\omega) \tag{17}$$

where $\mathrm{Head}(v,\omega)$ is the heading function, $\mathrm{Dist}(v,\omega)$ is the obstacle threat function, $\mathrm{Vel}(v,\omega)$ is the velocity function, $\mathrm{Dist_{GP}}(v,\omega)$ is the global path deviation function, and $\mathrm{Goal}(v,\omega)$ is the terminal convergence function. The coefficients k_1–k_5 are the weight factors.

The obstacle threat evaluation function is improved by dividing into the distance between the robot and static obstacles and the distance between the robot and dynamic obstacles. It improves the safety of the navigation path by enabling more precise handling of various obstacles. The obstacle threat

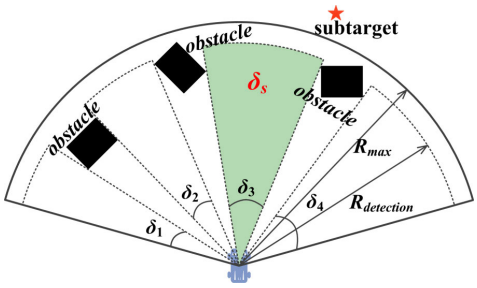


Fig. 7. Adaptive dynamic window. $R_{\max}$ is the maximum radius of the window, and $R_{\text{detection}}$ is the maximum detection distance of the robot. The areas δ_1–δ_4 are the regions, where the robot can travel, while Area δ_s represents the optimal region.

evaluation function is represented as follows:

$$\mathrm{Dist}(v,\omega) = \lambda\cdot \mathrm{Dist\_}S(v,\omega) + (1-\lambda)\cdot \mathrm{Dist\_}D(v,\omega) \tag{18}$$

$$\mathrm{Dist\_}S(v,\omega) = \begin{cases} D_s, & \text{if} \quad D_s \geq L \\ 0, & \text{otherwise} \end{cases} \tag{19}$$

$$\mathrm{Dist\_}D(v,\omega) = \begin{cases} D_d, & \text{if} \quad D_d \geq S \\ 0, & \text{otherwise} \end{cases} \tag{20}$$

$$D_s = \min\left\|x_p - x_s\right\|,\ D_d = \min\left\|x_p - x_d\right\| \tag{21}$$

$$S = v_t\cdot\Delta t,\ L = 2R_r \tag{22}$$

where D_s is the shortest distance from the predicted path endpoint to a static obstacle and D_d is the shortest distance from the predicted path endpoint to a dynamic obstacle. S is the distance the robot travels within the prediction time Δt, R_r is the radius of rotation of the robot, and λ is weight coefficient. Moreover, a global path deviation evaluation function is incorporated to enhance the tracking performance of robots on the global path while avoiding local obstacles. The global path deviation function can be described as follows:

$$\mathrm{Dist_{GP}}(v,\omega) = \begin{cases} \mathrm{dist_{gp}}, & \text{if} \quad \mathrm{dist_{obs}} \geq L \\ 0, & \text{otherwise} \end{cases} \tag{23}$$

$$\mathrm{dist_{obs}} = \min\left\|x_p - x_{uo}\right\|,\ \mathrm{dist_{gp}} = \min\left\|x_p - x_{gp}\right\| \tag{24}$$

where $\mathrm{dist_{obs}}$ is the shortest distance between the robot and dynamic obstacles at the current moment. $\mathrm{dist_{gp}}$ is the shortest distance from the end of the predicted path to the global path. The terminal convergence function is introduced to assign higher weights to paths that are closer to the target, ensuring that the robot does not linger near the goal and reaching the target rapidly. The terminal convergence function is given by

$$\mathrm{Goal}(v,\omega) = \begin{cases} \mathrm{dist_{st}}, & \text{if} \quad \mathrm{dist_{rs}} \leq R_{\text{dete}} \\ 0, & \text{otherwise} \end{cases} \tag{25}$$

$$\mathrm{dist_{st}} = \min\left\|x_p - x_{st}\right\|,\ \mathrm{dist_{rs}} = \min\left\|x_r - x_{st}\right\| \tag{26}$$

where $\mathrm{dist_{st}}$ is the shortest distance from the predicted path endpoint to the target, $\mathrm{dist_{rs}}$ is the shortest distance between the robot and the target, and R_{dete} is the obstacle detection range of the sensors.

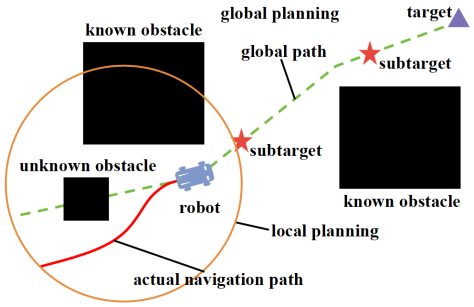


Fig. 8. Two-layer structural diagram. The green dashed line represents the path generated by the global path planning, the red pentagrams represent the subtarget point, and the red line represents the path generated by the local path planning of the robot.

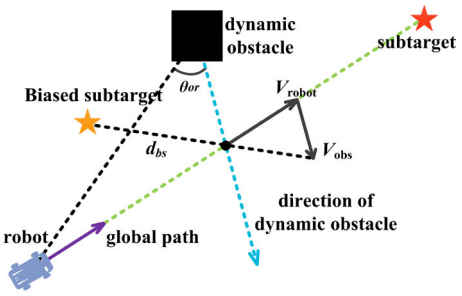


Fig. 9. Subtarget offset guidance diagram. The blue line represents the direction of obstacle movement, the purple line represents the current direction of robot movement, the green dashed line represents the path generated after global planning, the red pentagram represents the subtarget point, and the orange pentagram represents the subtarget point after offset guidance.

The inability to adjust the speed in time may cause the turning angle of robots to be insufficient in environments with numerous obstacles, increasing the risk of collision. To address this issue, the robot operates at a lower linear velocity and a higher angular velocity in dense obstacle areas, enabling it to follow the optimal path more closely and navigate to the target point more quickly and safely. The adaptive adjustments for linear and angular velocities are expressed as follows:

$$v(t+1) = v(t) + \alpha_1 \frac{\delta_s(t+1) - \delta_s(t)}{\delta_{\text{all}}} v_{\max} \tag{27}$$

$$\omega(t+1) = \omega_{\max} - \alpha_2 \frac{v(t)}{v_{\max}} \omega(t) \tag{28}$$

where v is the linear velocity of robot, ω is the angular velocity, δ_s is the optimal feasible region, δ_{all} is the maximum exploration area, and α_1 and α_2 are the adjustment parameters.

V. Two-Layer Path Planning Method

This article integrates the efficient path planning of the PB-RRT algorithm capabilities with the real-time adaptability of SDWA in complex and dynamic environments. As shown in Fig. 8, the PB-RRT algorithm handles global path planning in known environments, while SDWA avoids dynamic obstacles during navigation. After completing the heuristic global path planning, the nodes on the optimized heuristic path are sequentially extracted to set as the local subtargets for the SDWA. This enables the robot to track the global path in real time while avoiding dynamic obstacles. Since the robot reduces speed as it approaches the local target point, a dynamic subtargets strategy is proposed. The strategy calculates the distance between the current position of the robot and the dynamic subtargets. If this distance is less than a preset threshold, the dynamic subtargets shift to the next point along the sequence of subtargets. Otherwise, it remains at the current subtargets.

Based on the two-layer method, we propose an approach that adheres to collision avoidance principles, enhancing the safety of robot navigation. This method leverages the characteristic of the two-layer strategy, which always maintains a dynamic subtarget, to guide the robot in avoiding dynamic obstacles by adjusting the position of the dynamic subtargets. As shown in Fig. 9, when the robot approaches the boundary of a dynamic obstacle range, the movement direction of the dynamic obstacle and the angle θ_{or} between the line connecting the robot and the obstacle are determined. If the angle θ_{or} lies within the range of $(-\pi/2, \pi/2)$, the dynamic subtargets is offset to guide the robot. The direction of the adjustment is opposite to the vector sum of the velocity of the robot and the dynamic obstacle. The global path subtarget is then translated from the point, where the dynamic obstacle intersects the global path by a distance d_{bs}, in the direction of the offset, thereby guiding the robot to avoid the dynamic obstacle flexibly. If the angle θ_{or} lies outside the range of $(-\pi/2, \pi/2)$ or exceeds the influence range of dynamic obstacle, the dynamic subtarget reverts to the global path, directing the robot back to the global path.

VI. Experiments and Analysis

To verify the effectiveness and superiority of the proposed method, four environmental maps were constructed to compare the planning performance of the PB-RRT with Bi-RRT [18], IBi-RRT [25], and RB-RRT [19]. The path quality at various stages of the PO was compared with demonstrate the necessity of the PO method. Then, the application of the proposed method in the dynamic path planning was analyzed. Finally, the real environments are modeled to generate the movement path of the robot, validating the effectiveness of the method in practical scenarios. The simulations are performed in MATLAB R2022b, and conducted in a PC with 1.9 GHz Intel[1] Core[2] i5-1340P CPU and 16G RAM.

A. Comparative Experimental Analysis

This article conducts a comparative experiment in various environmental maps, including a random map, a corridor map, a warehouse map, and a maze map, each measuring 500 × 500. These maps provide different scenarios for the robot's path planning. The random map featured irregularly shaped obstacles placed randomly, the corridor map contained narrow paths, the warehouse map consisted of neatly arranged rectangular shelves, and the maze map included dense obstacles and misleading paths.

[1]Registered trademark.
[2]Trademarked.

TABLE I
RESULTS FOR PLANNING IN MULTIPLE ENVIRONMENTS

Environment	Algorithm	Average time(s)	Min time(s)	Max time(s)	Standard deviation	Mean samples	Mean length	Mean nodes
Random	Bi-RRT	2.002	1.611	2.754	0.383	96.7	951.0	39.6
	IBi-RRT	1.563	1.328	2.010	0.193	54.6	786.7	24.3
	RB-RRT	0.832	0.671	1.034	0.164	50.2	716.8	6.3
	PB-RRT	**0.636**	**0.499**	**0.826**	**0.089**	**36.0**	**694.2**	**5.2**
Corridor	Bi-RRT	1.880	1.569	2.287	0.205	89.5	883.0	36.9
	IBi-RRT	1.673	1.242	1.994	0.190	50.5	784.6	23.7
	RB-RRT	0.856	0.654	1.052	0.453	49.3	720.8	5.2
	PB-RRT	**0.638**	**0.473**	**0.874**	**0.096**	**35.7**	**703.3**	**4.6**
Warehouse	Bi-RRT	3.181	2.423	3.940	0.366	170.0	1025.2	52.6
	IBi-RRT	2.187	1.545	2.865	0.379	90.0	916.2	30.2
	RB-RRT	0.996	0.765	1.328	0.431	76.7	846.7	6.5
	PB-RRT	**0.778**	**0.609**	**0.950**	**0.155**	**46.8**	**831.7**	**5.3**
Maze	Bi-RRT	5.179	4.332	6.347	0.454	258.5	1189.5	60.9
	IBi-RRT	4.912	2.742	6.832	1.005	173.4	966.2	32.3
	RB-RRT	2.321	1.786	3.167	0.624	120.5	933.6	6.4
	PB-RRT	**1.559**	**1.099**	**2.223**	**0.345**	**90.2**	**913.3**	**5.1**

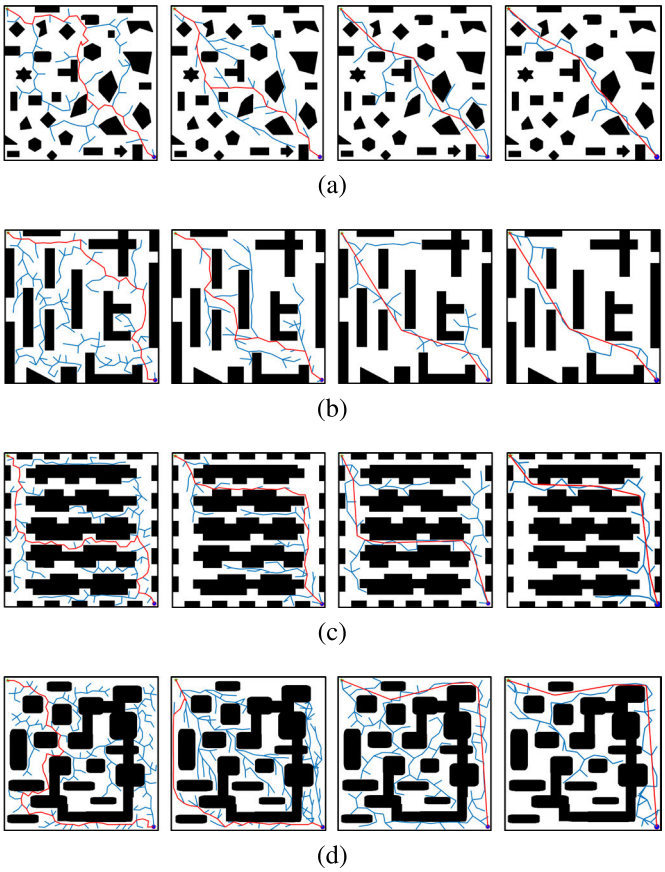


Fig. 10. Comparison of search paths for various algorithms. From left to right in each row are the planning results of Bi-RRT, IBi-RRT, RB-RRT, and PB-RRT. (a) Random. (b) Corridor. (c) Warehouse. (d) Maze.

Fig. 10 shows the results of each algorithm in four different environments under 50 samples. It is evident that the Bi-RRT algorithm lacks directional sampling, resulting in excessive exploration of the space, a wide distribution of sampling points, and the generation of numerous redundant nodes. The IBi-RRT algorithm minimizes invalid expansions but still generates many invalid sampling nodes in densely obstructed areas, resulting in slow expansion in complex environments. The RB-RRT algorithm integrates a node rejection mechanism with a greedy expansion strategy to focus sampling by limiting exploration in irrelevant regions of the state space. However, in densely obstructed areas, the node rejection mechanism may hinder the tree from discovering a feasible path, leading to a large number of invalid samples. In contrast, the PB-RRT algorithm demonstrates clear purpose and direction in expansion across all maps, with the starting tree and target tree guiding each other, reducing invalid sampling in local spaces, and achieving better paths with fewer sampling nodes.

Table I records the average time, standard deviation, number of sampled nodes, path length, and number of path nodes required to converge to the optimal solution across 50 independent experiments. Results show that in any map environment, the PB-RRT algorithm consistently achieves superior paths with fewer sampled nodes and shorter planning times. For instance, on the random map, the PB-RRT algorithm reduces average planning time by 68.2% compared with Bi-RRT, 59.3% compared with IBi-RRT, and 23.6% compared with RB-RRT. The reduction in average planning time is even more pronounced on the warehouse and maze maps. Moreover, the PB-RRT algorithm exhibits the smallest standard deviation in average planning time across all maps, demonstrating its stable performance and robust adaptability. As converging to the optimal solution, PB-RRT requires significantly fewer sampled nodes and achieves shorter path lengths than Bi-RRT, IBi-RRT, and RB-RRT.

The PB-RRT algorithm also conducts a global path with substantially the fewer average nodes and the shortest path length compared with other algorithms. The average number of nodes in the final path generated by the PB-RRT algorithm is significantly reduced compared with other algorithms, with only an average of five nodes. This represents reductions of 89%, 82%, and 17% compared with the Bi-RRT (46 nodes), IBi-RRT (28 nodes), and RB-RRT (6 nodes), respectively. It can be seen that the PB-RRT algorithm can efficiently explore the space using fewer samples while finding more optimal path solutions.

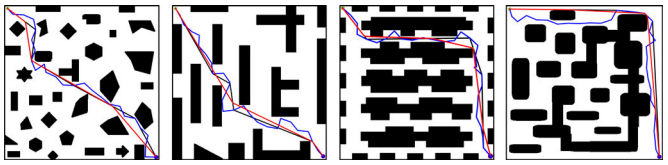


Fig. 11. Comparison of initial PO. The blue path represents the initial path, the black path is the global rewiring optimization, and the red path indicate the locally rewired and B-spline curve optimized path.

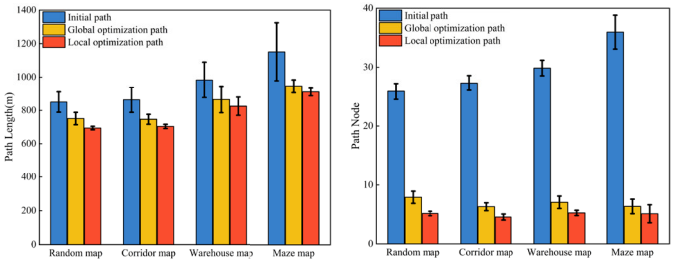


Fig. 12. Comparison of results at different stages of PO in different maps. (a) Path length. (b) Path node.

B. PO Results

The results of the optimized path experiment are shown in Fig. 11. The initial path contains numerous redundant nodes, resulting in an extended path length. After applying global rewiring optimization, the number of path nodes and the path length are significantly reduced, but the optimized path still retains some unnecessary segments from the initial path. Local rewiring optimization further shortens the path length and removes redundant nodes. Fig. 11 shows that the final paths after PO are shorter and smoother in different environmental maps. Compared with the initial paths, the quality of the paths is significantly improved.

Fig. 12 shows comparisons of path length and path nodes across different stages of PO in four scenarios. Obviously, the results clearly show that PO significantly enhances path quality, evident in the substantial reduction of both path length and the number of path nodes. It can be seen that the PO strategies are efficient, particularly in complex environments. The global path after PO is more suitable for robot navigation.

C. Experimental Results in Dynamic Environment

In the random map and the corridor map, several dynamic obstacles were introduced that move back and forth along blue paths, with their speeds changing randomly every three seconds. The blue paths are manually drawn and randomly generated in the simulation environment based on the map layout and dynamic obstacle design requirements. The resulting paths are then discretized into a series of continuous control points and smoothed using B-spline curves fitting to generate the final nonlinear paths. The Bi-RRT-DWA algorithm, IBi-RRT-DWA algorithm, RB-RRT-DWA algorithm, and PB-RRT-SDWA algorithm were employed for dynamic path planning to verify the performance of the proposed method. The comparison results of various algorithms are shown in Fig. 13. The Bi-RRT-DWA algorithm, IBi-RRT-DWA algorithm, and RB-RRT-DWA algorithm generate paths with significant fluctuations due to poor global path quality or unreasonable speed control when encountering obstacles. In contrast, the path generated by the PB-RRT-SDWA algorithm exhibits good smoothness due to dynamic subtarget guidance.

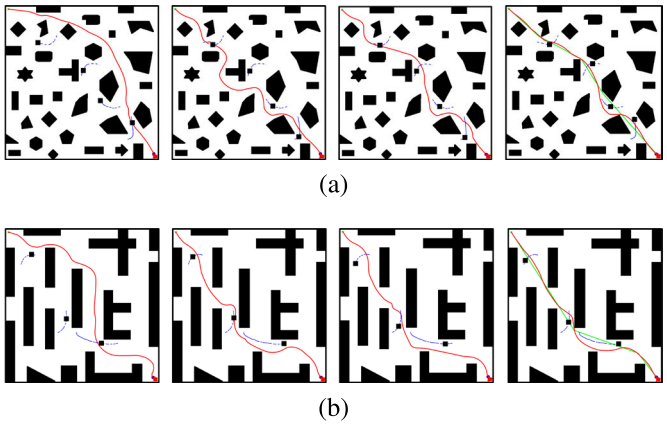


Fig. 13. Results of the robot in scenes with moving obstacles. From left to right in each row are the planning results of Bi-RRT-DWA, IBi-RRT-DWA, RB-RRT-DWA, and PB-RRT-SDWA. The green path is the global path, and the red path is the real-time local path. (a) Random. (b) Corridor.

TABLE II
RESULTS FOR PLANNING IN DYNAMIC ENVIRONMENTS

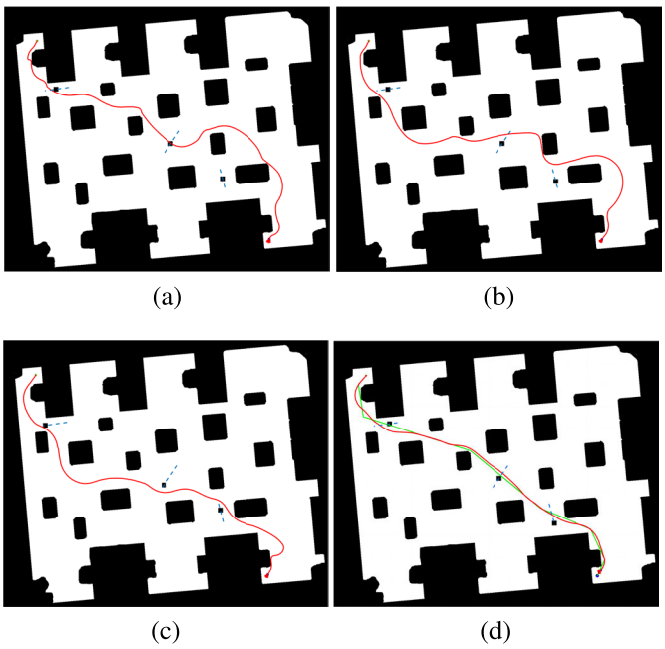
Environment	Algorithm	Time(s)	Path length	$D_{\min}$
Random	Bi-RRT-DWA	298.4	875.7	3.8
	IBi-RRT-DWA	199.3	804.3	5.2
	RB-RRT-DWA	178.6	777.8	6.1
	PB-RRT-SDWA	**142.1**	**746.2**	**8.4**
Corridor	Bi-RRT-DWA	258.7	844.3	5.3
	IBi-RRT-DWA	189.1	808.3	4.8
	RB-RRT-DWA	159.4	783.6	6.3
	PB-RRT-SDWA	**126.3**	**748.2**	**8.2**

Table II presents the average data obtained from 50 sets of dynamic path planning experiments, including planning time, path length, and the minimum distance to dynamic obstacles $D_{\min}$. The PB-RRT-SDWA algorithm demonstrated the shortest planning time, the shortest path length, and the biggest minimum distance to dynamic obstacles. For instance, in random maps, the planning time of the PB-RRT-SDWA algorithm was reduced by 52.4%, 28.7%, and 20.4% compared with the Bi-RRT-DWA, IBi-RRT-DWA, and RB-RRT-DWA algorithms, respectively. In addition, the path length generated by the PB-RRT-SDWA algorithm was shortened by 14.8%, 7.2%, and 4.1% compared with the Bi-RRT-DWA, IBi-RRT-DWA, and RB-RRT-DWA algorithms, respectively. It can be seen that the dynamic subtarget strategy and rational velocity control can significantly reduce both planning time and path length. Moreover, the offset guidance strategy enables the robot to maintain a relatively safe distance from dynamic obstacles.

The real indoor environment is implemented to demonstrate the effectiveness of the two-layer method in practical applications. As shown in Fig. 14(b), the testing area is a 10 × 7 m^2 office with a map resolution of 1 cm. The linear

(a) (b)

Fig. 14. Real experimental scenarios. (a) Indoor environment (office). (b) Outdoor environment (park).



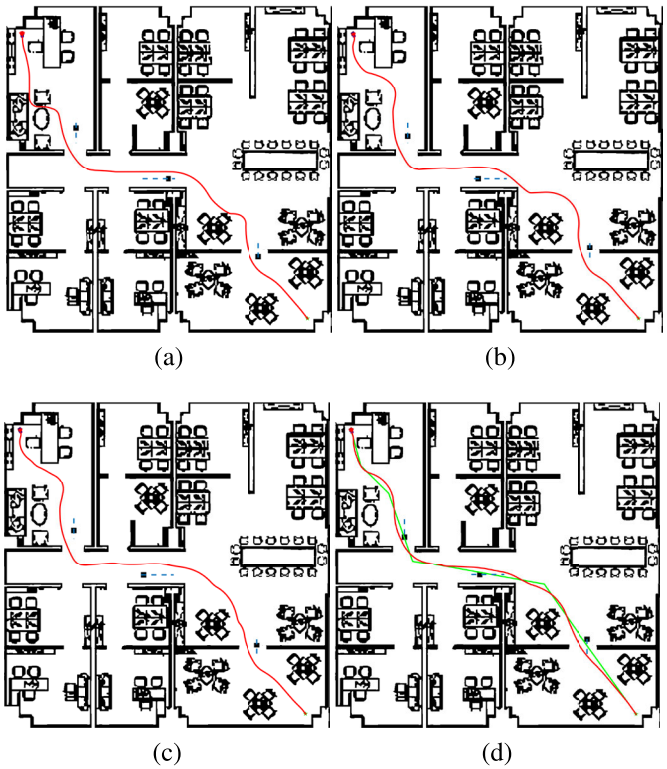
(a) (b) (c) (d)

Fig. 15. Results of robots in an office scenario. The green path is the global path, and the red path is the real-time local path. (a) Bi-RRT-DWA. (b) IBi-RRT-DWA. (c) RB-RRT-DWA. (d) PB-RRT-SDWA.

velocity and angular velocity of the robot are constrained to $v \in (0, 0.4$ m/s) and $\omega \in (-3.14, 3.14$ rad/s). As shown in Fig. 15, the experimental results demonstrate that the PB-RRT algorithm generates higher quality global paths during the global path planning stage. In the local path planning stage, the PB-RRT-SDWA algorithm performs local planning based on subtargets derived from the global path. Upon entering the collision range of a dynamic obstacle, it can promptly replan and select an optimal path to avoid the obstacle. Once the robot exits the influence range of the dynamic obstacle, it swiftly returns to the global path and safely reaches the target point. As shown in Table III, the planning time of the PB-RRT-SDWA algorithm was reduced by 41.7%, 36.7%, and 20.1% compared with the Bi-RRT-DWA, IBi-RRT-DWA, and RB-RRT-DWA algorithms, respectively. In addition, the path length generated by the PB-RRT-SDWA algorithm was shortened by 15.8%, 7.8%, and 5.4% compared with the Bi-RRT-DWA, IBi-RRT-DWA, and RB-RRT-DWA algorithms, respectively. It can be seen that the PB-RRT-SDWA algorithm can generate smoother paths and consistently maintains a relatively safe distance from dynamic obstacles compared with the other three algorithms.

TABLE III
RESULTS FOR PLANNING IN REAL DYNAMIC ENVIRONMENTS

Environment	Algorithm	Time(s)	Path length	$D_{\min}$
Office	Bi-RRT-DWA	430.8	1884	6.1
	IBi-RRT-DWA	397.1	1723	6.8
	RB-RRT-DWA	314.2	1668	7.3
	PB-RRT-SDWA	**251.3**	**1587**	**9.2**
Floor	Bi-RRT-DWA	400.7	1534	7.9
	IBi-RRT-DWA	374.2	1487	7.2
	RB-RRT-DWA	291.4	1456	8.7
	PB-RRT-SDWA	**236.9**	**1395**	**9.8**
Park	Bi-RRT-DWA	656.8	4881	8.1
	IBi-RRT-DWA	594.3	4331	9.6
	RB-RRT-DWA	506.7	4060	10.7
	PB-RRT-SDWA	**428.3**	**3892**	**12.3**



(a) (b) (c) (d)

Fig. 16. Results of robots in a larger office scenario. The green path is the global path, and the red path is the real-time local path. (a) Bi-RRT-DWA. (b) IBi-RRT-DWA. (c) RB-RRT-DWA. (d) PB-RRT-SDWA.

Then, we select a larger office scenario, as shown in Fig. 16. With dynamic subtarget guidance, the PB-RRT-SDWA algorithm maintains its navigation direction in complex environments, while the subtarget offset strategy enables the robot to flexibly avoid dynamic obstacles. Compared with other algorithms, the PB-RRT-SDWA algorithm demonstrates superior speed control during turns, allowing it to rapidly avoid dynamic obstacles, promptly return to the global path, and smoothly reach the goal. As shown in Table III, the PB-RRT-SDWA algorithm achieves the shortest planning time, the most optimal path length, and the greatest minimum distance from dynamic obstacles.

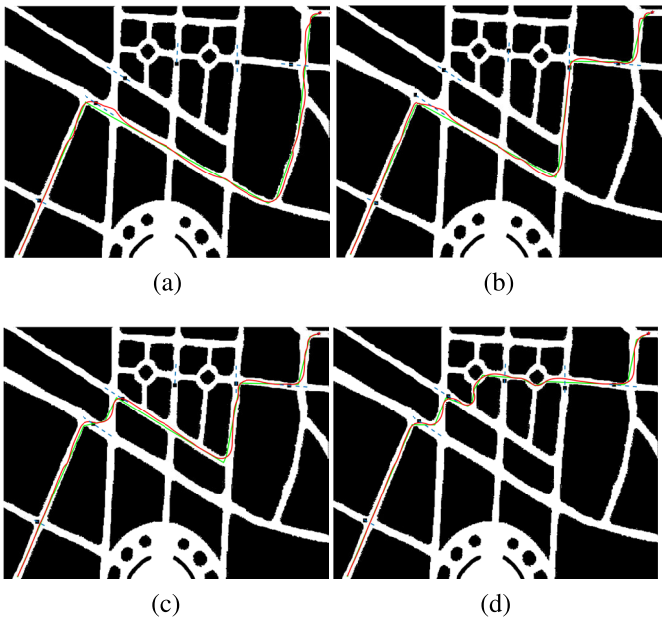


Fig. 17. Results of robots in a real outdoor environment. The green path is the global path, and the red path is the real-time local path. (a) Bi-RRT-DWA. (b) IBi-RRT-DWA. (c) RB-RRT-DWA. (d) PB-RRT-SDWA.

As shown in Fig. 14(b), a real complex outdoor environment is selected for robot path planning and obstacle avoidance experiment. The performance of each algorithm in real outdoor environments is presented in Fig. 17. It is evident that the PB-RRT-SDWA algorithm produces smoother paths and exhibits greater stability and reliability in complex environments compared with other algorithms. The experimental results summarized in Table III further demonstrate that the two-layer path planning method is effective in practical applications.

VII. Conclusion

This article proposes an efficient and safe two-layer path planning method for complex and dynamic environments. In the first layer, the PB-RRT algorithm is employed to find the global path, which utilizes parallel sampling points and evaluation functions to achieve the optimal adaptive expansion. Then, the bidirectional guidance mechanism is introduced to accelerate the merging of bidirectional random trees. In addition, PO is introduced to generate a path that the robot can follow. The optimized path is used to extract key nodes as dynamic subtargets. The second layer introduces SDWA, which is combined with an offset guidance method to identify feasible paths for robots in dynamic environments. Experimental results in various complex dynamic environments show that the two-layer method can effectively generate an optimal path and safely avoid dynamic obstacles to reach the target.

Acknowledgment

Yaowei Hu, Xufei Chen, and Pingping Tang are with the School of Physics and Electronic Information, Anhui Normal University, Wuhu 241002, China (e-mail: hyw0911@ahnu.edu.cn; chenxufei@ahnu.edu.cn; tpping@ahnu.edu.cn).

Hui Zhang is with the School of Communication and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China (e-mail: zhhjoice@126.com).

Jiong Jin is with the School of Engineering, Swinburne University of Technology, Melbourne, VIC 3122, Australia (e-mail: jiongjin@swin.edu.au).

Shiwen Mao is with the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849 USA (e-mail: smao@ieee.org).

References

[1] R. Zhang et al., "Efficient and near-optimal global path planning for AGVs: A DNN-based double closed-loop approach with guarantee mechanism," *IEEE Trans. Ind. Electron.*, vol. 72, no. 1, pp. 681–692, Jan. 2025.

[2] J. Wang, T. Li, B. Li, and M. Q.-H. Meng, "GMR-RRT: Sampling-based path planning using Gaussian mixture regression," *IEEE Trans. Intell. Vehicles*, vol. 7, no. 3, pp. 690–700, Sep. 2022.

[3] S. Shan, J. Shao, H. Zhang, S. Xie, and F. Sun, "Research and validation of self-driving path planning algorithm based on optimized A-artificial potential field method," *IEEE Sensors J.*, vol. 24, no. 15, pp. 24708–24722, Aug. 2024.

[4] L. Zhang, Y. Zhang, and Y. Li, "Mobile robot path planning based on improved localized particle swarm optimization," *IEEE Sensors J.*, vol. 21, no. 5, pp. 6962–6972, Mar. 2021.

[5] C. Han, Z. Yu, X. Shi, and J. Fan, "Dynamic path planning for rapidly expanding autonomous vehicles in complex environments," *IEEE Sensors J.*, vol. 25, no. 1, pp. 1216–1229, Jan. 2025.

[6] Z. Lin, K. Wu, R. Shen, X. Yu, and S. Huang, "An efficient and accurate A-star algorithm for autonomous vehicle path planning," *IEEE Trans. Veh. Technol.*, vol. 73, no. 6, pp. 9003–9008, Jun. 2024.

[7] H. Yang, J. Qi, Y. Miao, H. Sun, and J. Li, "A new robot navigation algorithm based on a double-layer ant algorithm and trajectory optimization," *IEEE Trans. Ind. Electron.*, vol. 66, no. 11, pp. 8557–8566, Nov. 2019.

[8] Y. Liu, Z. Hou, J. Qu, X. Liu, and Q. Fan, "Optimized RRT planning with CMA-ES for autonomous navigation of magnetic microrobots in complex environments," *IEEE/ASME Trans. Mechatronics*, vol. 29, no. 6, pp. 4826–4835, Dec. 2024.

[9] J. Yin et al., "Reliable global path planning of off-road autonomous ground vehicles under uncertain terrain conditions," *IEEE Trans. Intell. Vehicles*, vol. 9, no. 1, pp. 1161–1174, Jan. 2024.

[10] H. Li, W. Liu, C. Yang, W. Wang, T. Qie, and C. Xiang, "An optimization-based path planning approach for autonomous vehicles using the DynEFWA-artificial potential field," *IEEE Trans. Intell. Vehicles*, vol. 7, no. 2, pp. 263–272, Jun. 2022.

[11] H. Yang, X. Xu, and J. Hong, "Automatic parking path planning of tracked vehicle based on improved A* and DWA algorithms," *IEEE Trans. Transport. Electrific.*, vol. 9, no. 1, pp. 283–292, Mar. 2023.

[12] N. Wang and H. Xu, "Dynamics-constrained global–local hybrid path planning of an autonomous surface vehicle," *IEEE Trans. Veh. Technol.*, vol. 69, no. 7, pp. 6928–6942, Jul. 2020.

[13] J. Wang, X. Yuan, Z. Liu, W. Tan, X. Zhang, and Y. Wang, "Adaptive dynamic path planning method for autonomous vehicle under various road friction and speeds," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 10, pp. 10977–10987, 2023.

[14] Y. Zhang, H. Wang, M. Yin, J. Wang, and C. Hua, "Bi-AM-RRT: A fast and efficient sampling-based motion planning algorithm in dynamic environments," *IEEE Trans. Intell. Vehicles*, vol. 9, no. 1, pp. 1282–1293, Jan. 2024.

[15] B. Lindqvist, A. Patel, K. Löfgren, and G. Nikolakopoulos, "A tree-based next-best-trajectory method for 3-D UAV exploration," *IEEE Trans. Robot.*, vol. 40, pp. 3496–3513, 2024.

[16] W. Chi, Z. Ding, J. Wang, G. Chen, and L. Sun, "A generalized Voronoi diagram-based efficient heuristic path planning method for RRTs in mobile robots," *IEEE Trans. Ind. Electron.*, vol. 69, no. 5, pp. 4926–4937, May 2022.

[17] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed RRT: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2014, pp. 2997–3004.

[18] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Proc. IEEE Int. Conf. Robot. Autom. Symposia*, vol. 2, Apr. 2000, pp. 995–1001.

[19] L. Wang, Y. Zhang, and C. Guo, "Path planning for a prostate intervention robot based on an improved bi-RRT algorithm," *IEEE/ASME Trans. Mechatronics*, vol. 30, no. 1, pp. 668–678, Feb. 2025.

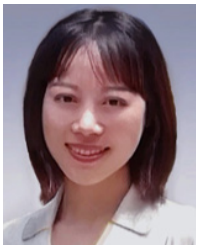
[20] Y. Huang and F. Zhang, "Variable curvature path planning for robot-assisted flexible needle insertion based on improved bi-RRT algorithm," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–14, 2024.
[21] B. Li and B. Chen, "An adaptive rapidly-exploring random tree," *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 2, pp. 283–294, Feb. 2022.
[22] O. Salzman and D. Halperin, "Asymptotically near-optimal RRT for fast, high-quality motion planning," *IEEE Trans. Robot.*, vol. 32, no. 3, pp. 473–483, Jun. 2016.
[23] J. Wang, W. Chi, C. Li, and M. Q.-H. Meng, "Efficient robot motion planning using bidirectional-unidirectional RRT extend function," *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 3, pp. 1859–1868, Jul. 2022.
[24] L. Jiang, S. Liu, Y. Cui, and H. Jiang, "Path planning for robotic manipulator in complex multi-obstacle environment based on improved_RRT," *IEEE/ASME Trans. Mechatronics*, vol. 27, no. 6, pp. 4774–4785, Dec. 2022.
[25] M. Zhang, Y. Chen, S. Luo, Q. Li, H. Zhao, and L. Zu, "Path planning of the robotic manipulator based on an improved bi-RRT," *IEEE Sensors J.*, vol. 24, no. 19, pp. 31245–31261, Oct. 2024.
[26] B. Hu, Z. Cao, and M. Zhou, "An efficient RRT-based framework for planning short and smooth wheeled robot motion under kinodynamic constraints," *IEEE Trans. Ind. Electron.*, vol. 68, no. 4, pp. 3292–3302, Apr. 2021.
[27] L. Chen, Y. Shan, W. Tian, B. Li, and D. Cao, "A fast and efficient double-tree RRT-like sampling-based planner applying on mobile robotic systems," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 6, pp. 2568–2578, Dec. 2018.
[28] W. Zhang, L. Shan, L. Chang, and Y. Dai, "SVF-RRT: A stream-based VF-RRT for USVs path planning considering ocean currents," *IEEE Robot. Autom. Lett.*, vol. 8, no. 4, pp. 2413–2420, Apr. 2023.
[29] J. Qi, H. Yang, and H. Sun, "MOD-RRT: A sampling-based algorithm for robot path planning in dynamic environment," *IEEE Trans. Ind. Electron.*, vol. 68, no. 8, pp. 7244–7251, Aug. 2021.
[30] R. Szczepanski, "Safe artificial potential field–novel local path planning algorithm maintaining safe distance from obstacles," *IEEE Robot. Autom. Lett.*, vol. 8, no. 8, pp. 4823–4830, Aug. 2023.
[31] M. Dobrevski and D. Skočaj, "Dynamic adaptive dynamic window approach," *IEEE Trans. Robot.*, vol. 40, pp. 3068–3081, 2024.
[32] S. Yasuda, T. Kumagai, and H. Yoshida, "Safe and efficient dynamic window approach for differential mobile robots with stochastic dynamics using deterministic sampling," *IEEE Robot. Autom. Lett.*, vol. 8, no. 5, pp. 2614–2621, May 2023.
[33] J. Wang, M. Q.-H. Meng, and O. Khatib, "EB-RRT: Optimal motion planning for mobile robots," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 4, pp. 2063–2073, Oct. 2020.
[34] L. Wang, Z. Wang, Z. Ying, X. Bai, and N. Xu, "A path planning framework based on an improved weighted heuristic RRT and optimization strategy," *IEEE Trans. Intell. Vehicles*, vol. 9, no. 1, pp. 1941–1952, Jan. 2024.

Yaowei Hu is currently pursuing the M.S. degree in electronic information with the School of Physics and Electronic Information, Anhui Normal University, Wuhu, China.

His research interests include autonomous navigation and control of robots, collision detection, and mathematical modeling.

Xufei Chen is currently pursuing the M.S. degree in electronic information with the School of Physics and Electronic Information, Anhui Normal University, Wuhu, China.

His research interests include swarm intelligence, robotics, and automation, as well as their applications in intelligent manufacturing and intelligent cities.



Pingping Tang (Member, IEEE) received the Ph.D. degree in communications engineering from Nanjing University of Posts and Telecommunications (NJUPT), Nanjing, China, in 2021.

She is currently an Associate Professor with the School of Electronic and Information Engineering, Anhui Normal University (AHNU), Wuhu, China. She has authored or co-authored a series of research papers in international journals and conferences. Her research interests include wireless networking, swarm intelligence computing, multimedia communications, network traffic identification, and intelligent network computing.

Hui Zhang (Member, IEEE) received the Ph.D. degree in information and communication engineering from Nanjing University of Posts and Telecommunications (NJUPT), Nanjing, China, in 2008.

He is currently a Professor and a Doctoral Supervisor with the Internet of Things Research Institute, NJUPT. His research interests include the future Internet of Things and artificial intelligence.

Jiong Jin (Member, IEEE) received the B.E. (Hons.) degree in computer engineering from Nanyang Technological University, Singapore, in 2006, and the Ph.D. degree in electrical and electronic engineering from the University of Melbourne, Melbourne, VIC, Australia, in 2011.

He is currently a Full Professor with the School of Engineering, Swinburne University of Technology, Melbourne. His research interests include network design and optimization, edge computing and intelligence, robotics and automation, and cyber–physical systems and the Internet of Things as well as their applications in smart manufacturing, smart transportation, and smart cities.

Dr. Jin was recognized as an Honorable Mention in the AI 2000 Most Influential Scholars List in IoT in 2021 and 2022. He is currently an Associate Editor of IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS and IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING.

Shiwen Mao (Fellow, IEEE) received the Ph.D. degree in electrical and computer engineering from Polytechnic University, Brooklyn, NY, USA, in 2004.

He is currently a Professor, an Earle C. Williams Eminent Scholar, and the Director of the Wireless Engineering Research and Education Center, Auburn University, Auburn, AL, USA. His research interests include wireless networks and multimedia communications.

Dr. Mao is a Distinguished Lecturer of the IEEE Communications Society and the IEEE Council of RFID. He is the Editor-in-Chief of IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING, a Member-at-Large of the IEEE Communications Society Board of Governors, and a Vice President of Technical Activities of the IEEE Council on Radio Frequency Identification (CRFID). He was a recipient of the IEEE ComSoc MMTC Outstanding Researcher Award in 2023, SEC 2023 Faculty Achievement Award for Auburn, IEEE ComSoc TC-CSR Distinguished Technical Achievement Award in 2019, Auburn University Creative Research and Scholarship Award in 2018, NSF CAREER Award in 2010, and several IEEE service awards. He was a co-recipient of the 2022 Best Journal Paper Award of IEEE ComSoceHealth Technical Committee, the 2021 Best Paper Award of Elsevier/KeAi Digital Communications and Networks Journal, the 2021 IEEE Internet of Things Journal Best Paper Award, the 2021 IEEE Communications Society Outstanding Paper Award, the IEEE Vehicular Technology Society 2020 Jack Neubauer Memorial Award, the 2018 Best Journal Paper Award and 2017 Best Conference Paper Award from IEEE ComSoc MMTC, and the 2004 IEEE Communications Society Leonard G. Abraham Prize in the Field of Communications Systems. He was the General Chair of IEEE INFOCOM 2022, a TPC Chair of IEEE INFOCOM 2018, and a TPC Vice-Chair of IEEE GLOBECOM 2022.