International Journal of Robotics and Control Systems
Vol. 5, No. 4, 2025, pp. 2118-2139
ISSN 2775-2658
http://pubs2.ascee.org/index.php/ijrcs

A Survey of RRT* Variants and Enhancements for Robotic Path Planning

Hameed Salman Hameed [a,1,*], Firas Abdulrazzaq Raheem [a,2], Omar Farouq Lutfy [a,3]

[a] College of Control and Systems Engineering, University of Technology - Iraq, Baghdad 10066, Iraq
[1] cse.22.03@grad.uotechnology.edu.iq; [2] Firas.A.Raheem@uotechnology.edu.iq; [3] omar.f.lutfy@uotechnology.edu.iq
* Corresponding Author

ARTICLE INFO

Article history
Received June 13, 2025
Revised August 23, 2025
Accepted October 31, 2025

Keywords
Collision-Checking;
Kinodynamic;
Path Planning;
Path Smoothing;
RRT*;
Dynamic Environment

ABSTRACT

Sampling-based planners like Rapidly-exploring Random Trees Star (RRT*) have proven powerful for robotic path planning. However, practical deployments expose limitations in convergence speed, dynamic feasibility, collision-checking overhead, and adaptability to changing environments. This paper introduces a novel practice-oriented taxonomy and decision map that aligns RRT* variants with constraints (dynamic, kinodynamic, and safety) and computes budgets. First, we review guided sampling techniques, including Informed RRT*, bridge-test sampling, and region-biased strategies that focus computational effort on promising regions, accelerating convergence. Next, we examine kinodynamic extensions such as Kinodynamic RRT* and LQR-RRT*, which embed system dynamics and optimal-control heuristics directly into the tree growth, yielding smooth, dynamically-feasible trajectories for under actuated manipulators. We then explore collision-checking optimizations, from lazy evaluation to multi-resolution batching, which reduce expensive obstacle-testing calls without sacrificing optimality guarantees. Moreover, dynamic-environment variants (Dynamic RRT*, ERRT, and hybrid RRT*-D* Lite) are surveyed to demonstrate efficient incremental re-planning under moving obstacles. Finally, we discuss post-processing methods, including CHOMP and shortcut smoothing that further refine raw RRT* paths into execution-ready trajectories. By synthesizing these improvements, we identify open challenges and propose a unified framework integrating guided sampling, kinodynamic control, and real-time re-planning for next-generation robotic path planners.

1. Introduction

Path planning computes collision-free, dynamically feasible trajectories that move a robot from start to goal while respecting kinematics, task constraints, and real-time limits in uncertain, cluttered environments. Key challenges include high dimensionality, no convex obstacles, moving hazards, and model uncertainty; success is judged by feasibility, path optimality/smoothness, computation time, and robustness. The RRT and its asymptotically-optimal variant RRT* have revolutionized sampling-based motion planning by offering probabilistic completeness and, in the case of RRT*, convergence toward the optimal solution as the number of samples grows. Specifically, it is a random sampling-based method used for path planning in robotic systems, generating random nodes in the space and incrementally building a tree to find a near-optimal, collision-free path [1], [2].

The process ensures unbiased exploration of the space, expanding towards unexplored regions while avoiding collisions [3]: Yet, vanilla RRT* suffers in practice from four key limitations [4]:

1. Slow convergence in high-dimensional or cluttered spaces, since uniform sampling explores large unpromising regions.
2. Neglecting dynamics, yielding kinematically feasible but dynamically infeasible trajectories for real robots.
3. Collision-checking overhead, as every new edge incurs expensive geometry tests.
4. Static-world assumption, which fails when obstacles move or new obstacles appear at runtime.

This survey reviews major work efforts that tackle these limitations, grouping them into five categories: guided sampling, kinodynamic extensions, collision-checking optimizations, dynamic-environment adaptation, and post-processing smoothing. It also introduces a survey of improvements to RRT* that minimizes the limitations in each section from 2020 to 2025 regarding the algorithms and methods used, hardware, and applications in this domain. Furthermore, we highlight open research directions and suggest how these advances could be composed into a unified real-time planner. Fig. 1 is a taxonomy of RRT* enhancements in robotic path planning, organized into six major categories of improvements. In particular, this figure represents a high-level map of all the main research threads for making RRT* faster, more robust, and better suited to real robotic systems, whether by biasing its sampling, guiding it with heuristics, hybridizing with other planners, adapting on the fly to moving obstacles, cleaning up the raw paths, or cutting down collision-check costs. More specifically, this study clarifies the latest advancements in path planning by the RRT*. This knowledge is essential for practitioners, scholars, and engineers in the field of robotics, providing a foundation for further RRT* improvement. The success of route planning requires understanding RRT algorithms and their diverse applications.

This study highlights the advantages and disadvantages of various improvements, providing researchers and professionals with additional insights to determine which improvement methods to employ. In this regard, the scientific community benefits from systematic reviews that simplify the process of discovering limitations and advantages to improve the RRT* for path planning in robotic systems and build upon prior studies, aiding scholars in gaining a more profound understanding of path planning in robotics [5]. Finally, open challenges and future research directions are outlined to guide further advancements in robotic path planning.

The remainder of this paper is structured as follows. Section 2 presents advanced sampling strategies, including Informed RRT*, Bridge-Test Sampling, and Region-Biased Techniques. Section 3 addresses kinodynamic extensions, covering both Kinodynamic RRT* and LQR-RRT*. Section 4 discusses collision-checking optimizations, where Lazy Evaluation and Multi-Resolution/Batch Checking are explored. Section 5 introduces dynamic-environment adaptation (DEA) strategies, including Dynamic RRT*, Execution-Extended RRT*, and the Hybrid RRT*-D* Lite framework. Section 6 highlights post-processing and path smoothing methods. Section 7 presents the experimental results and discussion, evaluating the proposed approaches in static and dynamic environments. Finally, Section 8 concludes the paper and outlines future research directions.

2. Guided Sampling Techniques

Guided sampling methods bias sample placement to regions likely to improve path quality or bridge narrow passages [6]. The RRT and its kinodynamic variations were improved by Ortiz-Haro et al. [7], who placed a strong emphasis on adaptive sampling and trajectory optimization. Specifically, their results demonstrated how the RRT algorithms are always being improved to satisfy the needs of more intricate robotic applications. In addition, Wang et al. [8] and Uwacu et al. [9]'s recent contributions demonstrated a persistent emphasis on enhancing RRT using cutting-edge sampling techniques, including adaptive point selection and hierarchical skeleton-guided exploration. These works highlight the importance of effective algorithms that can handle the complexities of high-dimensional planning spaces.

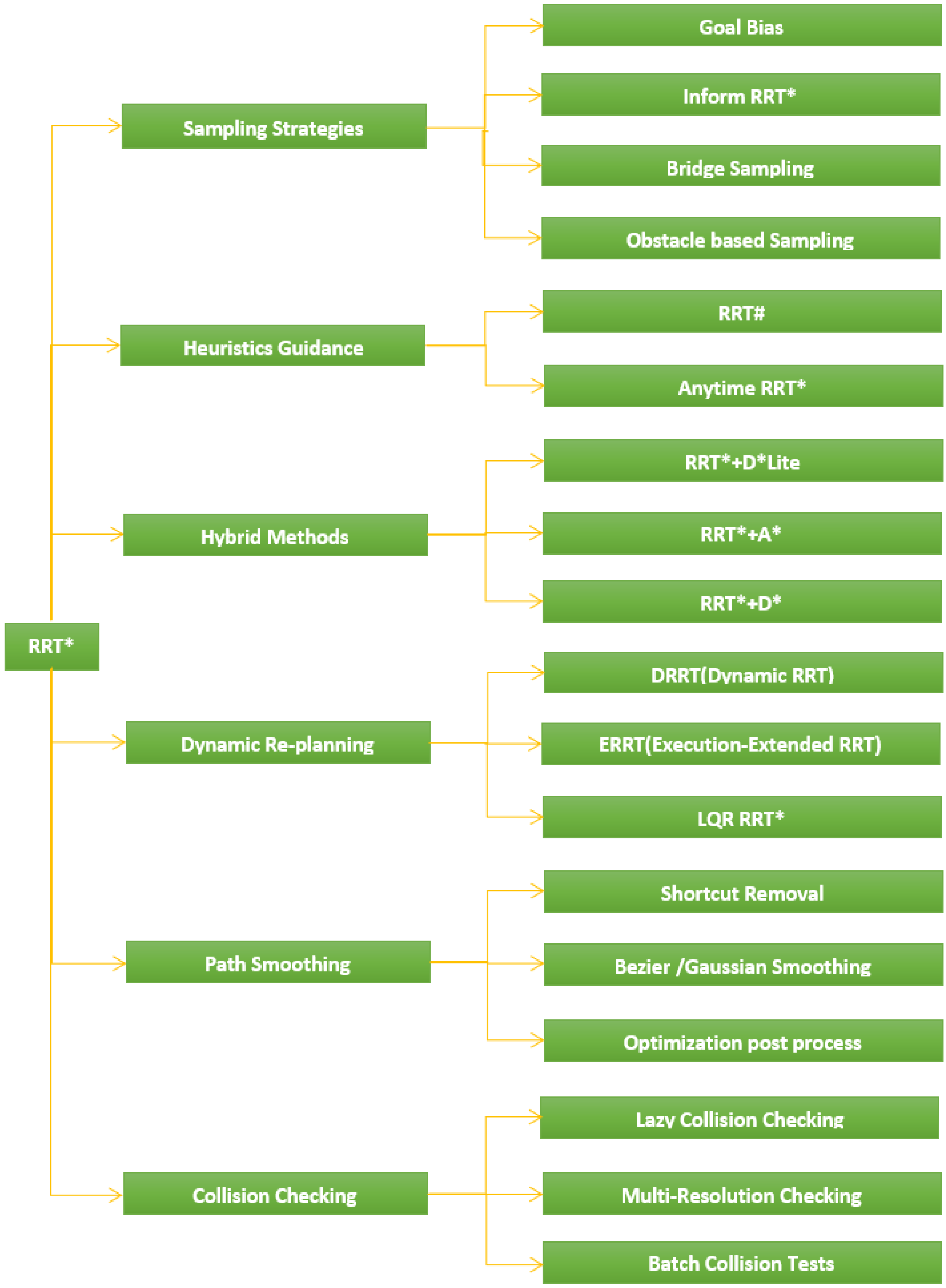

Fig. 1. The taxonomy of RRT* enhancements in robotic path planning of RRT*

2.1. Informed RRT*

Once an initial solution of cost C_max is found, the Informed RRT* restricts future sampling to the prolate hyper spheroid defined by all points x satisfying the following condition [10].

$$\|X - X_{Start}\| + \|X - X_{Goal}\| \leq C_{max} \tag{1}$$

This focus eliminates sampling in regions that cannot beat the current best cost, dramatically improving convergence rate without introducing new tuning parameters. Using direct sampling following the discovery of the first solution, the authors in [10] offered the informed RRT*-Connect. The suggested approach, in contrast to RRT*-Connect, only examines states that could offer superior solutions to the current one. The issues with the Informed-RRT* algorithm are uneven sampling and unsmooth paths; thus, Yuan et al. [11] proposed an improved Informed-RRT* algorithm founded on an adaptive growth strategy and elliptical region-based variable weight sampling technique, with trajectory optimization, to accelerate planners utilizing both the basic RRT and the RRT*. Ryu et al. [12] proposed an improved Informed RRT* using grid map skeletonization for the initial path, yielding earlier optimization and reduced runtime with lower variance compared to the conventional method. The authors in [13] explored the use of informed sampling and path optimization, producing a family of algorithms known as optimized informed RRTs. The Dynamic Informed Bias RRT*-Connect is a novel method proposed by Li et al. [14]. This technique presented a dynamic bias point set that directs dual tree development with precision targets. It is based on potential and explicit informed bias sampling. Furthermore, they improved the algorithmic heuristics assessment framework by proposing two novel measures that reflect the algorithm's properties. Through the integration of the improved Informed-RRT* algorithm with the Dynamic Window Approach (DWA) algorithm, Wu et al. [3] presented an improved path planning technique to reduce redundant nodes and accelerate convergence by improving an Artificial Potential Field (APF) within the elliptical sampling area. Dai et al. [15] proposed an improved method to solve the problem of weak path performance and reduced navigation efficiency of the Informed-RRT*. In particular, the greedy algorithm is applied in the path planning approach, and the search range of the potentially best parent node turns into the built path, rather than the tree of nodes.

2.2. Bridge-Test Sampling

Narrow passages often trap uniform-samplers. Bridge-test sampling draws two random points; if both lie in collision but their midpoint is free, that midpoint is added to the tree. This "bridge-test" effectively finds samples in constricted regions between obstacles.

The mechanism includes the following:

1. With probability$\mathrm{P_{BTS}}$, draw two uniform samples, p1,and p2,in the configuration space.
2. If both p1 and p2 lie in collision, compute the midpoint as follows:

$$m = \frac{1}{2}(P_1 + P_2) \tag{2}$$

3. If m is collision-free, accept mmm as the new sample X_rand; otherwise, fall back to a uniform sample.
4. Proceed with the usual RRT* extend/rewire steps using $\mathrm{X_{rand}}$.

Key parameters include:

1. Bridge probability $\mathrm{P_{BTS}} \in [0.1, 0.3]$ balances narrow-passage probing versus global exploration.
2. Hybridization with weight $1 - \mathrm{P_{BTS}}$ on uniform sampling preserves coverage and asymptotic optimality.

Tu et al. [16] suggested an enhanced RRT algorithm. Initially, the simplified bridge test and the point cloud clustering technique were used to identify the map's narrow pathways and position the root nodes within them. Then, using the root nodes, starting locations, and endpoints, many random trees are created. The random trees progressively grow and link to one another as the number of samples rises. Qiu et al. [17] proposed an innovative path planning strategy, termed Dynamic Bridging Rapidly Exploring Random Tree, which aims to facilitate secure and rapid path navigation.

Vol. 5, No. 4, 2025, pp. 2118-2139

2.3. Region-Biased Strategies

Region-biased RRT* adapts the sampling distribution online by identifying promising homotopy classes or graph regions where past samples yielded large cost reductions. New samples concentrate in these "hotspots," further accelerating early convergence.

The mechanism includes the following:

1. Skeleton extraction: Compute a coarse representation (e.g., medial-axis or visibility graph) of free-space corridors.
2. Region definition: Define moving sampling regions that track the advancing frontier of the RRT* along the skeleton edges.
3. Biased sampling: With probability P_{RBSP}, draw X_{rand} uniformly within the current region; otherwise, sample globally to maintain coverage.
4. Iteration: Update the region as the tree expands, continuously biasing toward skeleton-based corridors.

Key parameters include:

Region-biased sampling strategies guide the growth of the RRT tree by focusing sampling efforts on particular regions of the search space to improve efficiency and convergence speed. The main parameters that define these strategies include Region Definition, Bias Probability, Region Size and Shape, Adaptive Parameters, and Sampling Distribution [18], [19]. These parameters control the balance between exploration of the space and intensification near promising areas, directly impacting the convergence speed and the success rate of the RRT* algorithm using region-biased sampling. Liu et al. [20] introduced the variable probability goal-biased strategy to guide the generation of random sampling points and reduce the number of sampling points. The adaptive dynamic step size update function is utilized to adjust the expansion amplitude of random trees and overcome the shortcomings of blind exploration. Fan et. al. [21] proposed UAV trajectory planning based on a bi-directional APF-RRT* algorithm with goal-bias using a goal-biased strategy to reduce the number of iterations and compared Informed-RRT*, Bi-RRT*, and improved P-RRT* algorithms. Zhang et al. [22] proposed an improved heuristic Bi-RRT algorithm suitable for obstacle avoidance of vehicles in an unknown dynamic environment.

3. Kinodynamic Extension

Handling differential constraints is critical for robots with dynamics, under-actuated joints, or torque limits. The classic RRT* algorithm is designed for path planning in static environments, assuming simple holonomic motion models, which means that the planner only considers geometric constraints without explicitly considering the system's dynamics (like velocity, acceleration, or forces). However, many robotic systems have dynamic constraints (e.g., velocity limits, acceleration limits, and non-holonomic constraints, such as car-like robots or robotic arms).

3.1. Kinodynamic RRT*

Kinodynamic RRT* generalizes the nearest-neighbor and rewiring steps using the true cost-to-go under linear differential constraints. Each candidate extension solves a two-point boundary-value problem (often via a linear-quadratic regulator) [23]. An extension of RRT* guarantees asymptotic optimality for systems with differential constraints using optimal controllers to connect sampled states.

The mechanism includes the following:

Dynamics model $\dot{x}=Ax+Bu$ (or local linearization of $\dot{x}=f(x,u)$).

Optimal connector: Use a fixed-final-state–free-final-time controller (e.g., LQR or two-point BVP solution) to precisely and optimally steer between any two states.

Sampling and rewiring: It remains identical to RRT* except that "nearest" and "cost" use the true control-effort metric, and "steer" executes the optimal controller.

Guarantee: Under controllability and mild regularity, KD-RRT* converges almost surely to the minimum-cost trajectory respecting the system's dynamics [24].

Chen et al. [25]. created an inferencing system using fuzzy logic to avoid obstacles in a changing environment. They provided Fuzzy-Kinodynamic RRT, a technique that uses the classic RRT algorithm to create a dynamic route. To decrease the dimensionality of the sample space, D. Zheng and Tsiotras [26] proposed the use of a partial-final-state-free (PFF) optimum controller in the kinodynamic RRT*. The suggested accelerated kinodynamic RRT*, known as Kino-RRT*, only samples a portion of the state space rather than the entire state space, and the PFF optimum controller chooses the remaining states. They developed a rewiring technique in template-based RRT that effectively finds high-quality pathways while preserving a quick sampling speed, thanks to YANG et al. [27], who suggested a method that exceeds the baseline and may identify better answers faster, according to the experiment conducted in an artificial environment. Ježek et al. [28] proposed a novel approximate method called the Kinodynamic Rapidly-exploring Random Forest (KRRF) to find a collision-free multi-goal trajectory that satisfies the motion constraints of the robot. The KRRF simultaneously grows kinodynamic trees from all targets towards all other targets while using the other trees as a heuristic to boost the growth. Ge et al. [29] proposed a method based on the kinodynamic RRT* algorithm. This method hinges on a state-space model for fixed-wing UAV flight dynamics using feedback linearization. Tange et al. [30] introduced a sampling-based kinodynamic planning algorithm for quadrotors, which provides a dynamically practical trajectory in a complex environment. They designed a method to restrict the sampling state using the vector field to construct a cone in the sampling stage of the RRT* to generate a trajectory as smoothly as possible.

3.2. LQR-RRT*

LQR-RRT* (Linear Quadratic Regulator Rapidly-exploring Random Tree Star) is an extension of the RRT* algorithm designed for planning optimal and dynamically feasible paths for systems with linear dynamics or linearized approximations. It is an advanced path planning algorithm that combines the strengths of the RRT* algorithm with optimal control principles. The LQR-RRT* advances kinodynamic planning by [31]:

1. Metric: Using the infinite-horizon LQR value function V(x) around each node as the distance metric to samples.
2. Steering: Simulating the LQR feedback law to extend the tree toward a random sample in a dynamics-respecting way.

The mechanism includes the following:

1. Dynamics model: Locally linearize $\dot{x} = (x, u)$ about the current node to obtain A and B.
2. LQR solution: Compute the infinite-horizon LQR gain K and the cost-to-go matrix S by solving the continuous-time algebraic Riccati equation.
3. Nearest and Near: Use the LQR value function $V(x) = X^T SX$ as the distance metric for both the nearest neighbor and the neighbor-radius queries.
4. Steer: Integrate the LQR feedback law $U = -K(X - X_{target})$ over a short horizon to generate a dynamically feasible trajectory segment toward the sampled state.
5. Rewire: Re-evaluate and rewire existing nodes using the same LQR-based cost-to-go, preserving asymptotic optimality under kinodynamic constraints.

This approach requires no hand-tuned steering parameters and naturally produces energy-efficient, smooth trajectories for under actuated manipulators. Yang et al. [32] provided a new motion-planning framework that incorporates CBFs, the Linear Quadratic Regulator (LQR), and sampling-based techniques. This method eliminates explicit collision checking during samplings and does not

Vol. 5, No. 4, 2025, pp. 2118-2139

involve solving the QPs for control synthesis. Instead, it rejects dangerous paths using CBF and generates optimum controls using LQR. Moreover, Zhang et al. [33] suggested a motion planning and tracking control for autonomous vehicles based on an improved RRT* and Particle Swarm Optimization Linear Quadratic Regulator to tackle slow convergence and excessive randomness in RRT*. In addition, Doerr Linares [34] developed path planning and control algorithms using (LQR-RRT*), path smoothing, and a closed-loop nonlinear receding horizon control optimizer for a robotic Astrobee free-flyer for tracking the trajectory. Zhong et al. [35] considered the path planning and control of flexible spacecraft in on-orbit service. They used the RRT* algorithm and the LQR to generate a path considering both dynamic performance and collision avoidance. Liu et al. [36] proposed an innovative preoperative path planning algorithm based on an improved MDP-LQR-RRT*, and the workflow of the MDP-LQR-RRT*, which introduces a workflow that uses the LQR controller to generate path points and utilizes the Markov decision process model to refine the path. Liu et al. [37] proposed a combined spatiotemporal trajectory planning and control method based on a reachable set and optimization technique to address the shortcomings of existing reachable set methods in terms of unreasonable risk characterization and lack of risk consideration in the spatiotemporal corridor generation process. Particularly, a risk assessment model that takes into account the uncertainty of the predicted location distribution and a reachable set spatiotemporal corridor generation strategy that integrates the risk field were proposed.

4. Collision-Checking Optimizations

This optimization refers to strategies and techniques designed to decrease the cost and enhance the efficiency of collision detection during the implementation of the RRT* algorithm [38]-[40]. It is often the most computationally expensive step because it involves geometric tests against the environment for each new edge or node. Frequent and exhaustive collision checks slow down the algorithm, especially in complex environments or high-dimensional spaces. Collision-checking optimizations aim to minimize unnecessary collision tests without compromising the correctness or optimality of the RRT* algorithm. In particular, collision queries often dominate runtime [41], [42]. Two main streams address this issue, including:

4.1. Lazy Evaluation

The Lazy RRT* postpones collision checking until a candidate path has been found. Once a complete path is extracted, only that path's edges are checked, and colliding edges are repaired via rewiring. This reduction in per-sample tests preserves optimality while cutting the overhead.

4.2. Multi-Resolution and Batch Checking

Multi-resolution schemes first perform coarse-grained checks (e.g., bounding-box tests) on many edges in the batch, refining only those flagged as potentially colliding. Batched spatial data structures (e.g., sweep-and-prune) further amortize the cost of collision detection over multiple edges.

The mechanism includes the following:

Lazy Evaluation: Defer expensive collision checks until a candidate path is nearly complete, caching edge-feasibility results and only validating when necessary.

1. Adaptive Lazy Checking: Dynamically adjust when and which edges to test based on heuristic estimates of collision probability, further cutting wasted checks.
2. Probabilistic/Batched Checking: Use fast, approximate tests (e.g., learned clearance predictors or broad-phase checks) in batches to prune safe or unsafe edges before exact testing.

Key parameters include:

1. Lazy threshold: How many times an edge can be deferred before a forced check.
2. Batch size: The number of edges grouped per the coarse-phase test.

3. Adaptive heuristics: Metrics (e.g., edge length and proximity to obstacles) guiding selective validation.

Yu and Gao [43] proposed a novel learning-based method for decreasing collision checking to accelerate path planning by training graph neural networks that perform path exploration and path smoothing. Furthermore, Shi et al. [44] proposed a hybrid safety certificate approach in workspace and configuration space to improve collision checking efficiency for large-scale complex path planning problems. Henning and Schildbach [45] and Henning and Schildbach [35] presented a novel method of obstacle detection for non-holonomic systems in constrained environments. This new approach was evaluated against other collision-checking methods commonly used alongside various path planners. Zha and W [46] presented a sampling-driven approach that utilizes collisions to improve motion planning. The method is founded on the RRT algorithm* and utilizes rapid motion primitive generation and collision detection for multicopters.

5. Dynamic-Environment Adaptation (DEA)

Dynamic-Environment Adaptation for RRT* involves enhancing the classic RRT* algorithm to efficiently handle environments where obstacles or conditions change over time, which means that the path planner can update, repair, or replan the tree as the environment dynamically evolves without restarting from scratch. In this context, robots in real environments must handle moving obstacles and map changes [47], [48]. These techniques enable RRT* to react to moving obstacles or map updates by repairing or re-growing rather than re-planning from scratch [49].

The mechanism includes the following:

1. Tree Pruning and Repair (DRRT): Mark and remove only the invalidated edges/nodes when obstacles move, and then regrow the trimmed tree until the goal is reachable again.
2. Self-Repairing Anytime RRT (SMARRT): Continuously monitor a feasibility horizon ahead of the robot, prune infeasible segments on the fly, and quickly resample within a multi-resolution utility map for fast re-planning [50].
3. Real-Time Kinodynamic RRT (RRTX): Maintain a dynamic cache of collision-checked edges and propagate cost-to-go updates incrementally as the map changes, preserving kinodynamic feasibility in real time.

5.1. Dynamic RRT* (DRRT)

It is a re-planning extension of the RRT* that preserves and repairs most of the existing tree under obstacle-map changes by pruning only the invalidated branches and re-growing as needed, enabling fast, incremental updates in dynamic environments [51].

1. Root-at-Goal Growth: By rooting the tree at the goal and growing toward the start, only the few branches intersecting new obstacles need pruning when the map changes.
2. Selective Pruning: Upon detecting a moved or newly appeared obstacle, identify all edges that collide and remove just those sub trees [52].
3. Regrowth: Continue sampling from the remaining tree toward the start (or goal), quickly restoring connectivity without rebuilding the entire roadmap.
4. Asymptotic Optimality: Retains RRT*'s optimal-path guarantees in the limit of infinite samples, even under repeated re-planning events [53].

The DRRT prunes subtrees whose paths intersect new obstacle regions and regrows them via additional sampling. By only re-planning the affected branches, the DRRT avoids full-tree rebuilds and maintains asymptotic optimality under slow obstacle motion. Jain and Kashyap [54] introduced a superior UAV collision management strategy that merges the Improved RRT* algorithm for global path planning with the Dynamic Window. The approach for real-time adjustments permits the UAV to flexibly alternate between extended planning for fixed obstacles and accurate trajectory optimization in dynamic settings. Cui et al. [55] proposed a reverse tree-guided RRT (RT-RRT) algorithm to perform navigation tasks in dynamic environments efficiently. The method begins by building a reverse tree anchored at the goal state to find an initial path. If a collision is detected along the path, the RT-RRT builds a forward tree with the current robot state as its root in the same configuration space, until it links with the reverse tree to discover a new path. Moreover, Katiyar and Dutta [56] presented a CG-Space-based dynamic path planning and obstacle avoidance algorithm for a ten-DOF wheeled mobile robot (Rover) navigating through 3D rough terrains. In addition, Li et al. [57] proposed a dynamic RRT* with bridge guidance (DBRRT*) to accelerate feasible path searching by adding a dynamic sampling method and an improved bridge test. To validate the effectiveness of the DBRRT*, a series of experiments on a 6-DOF manipulator was conducted on some typical path planning algorithms. Lopes et al. [58] transposed an existing algorithm developed for 2D environments to 3D, leveraging a heuristic to optimize the generated paths in terms of path length, memory consumed, and execution time. Along with this scalability to 3D scenarios, a modification was introduced that trades off some execution time for a substantial improvement in the path length [59].

5.2. Execution-Extended RRT* (ERRT)

The ERRT maintains two trees, one for planning, and one for execution, growing the execution tree only along validated branches while the planning tree continues sampling. This parallelism enables the robot to move safely even as new samples improve future paths. A re-planning-enhanced RRT variant interleaves planning and execution by leveraging a waypoint cache and adaptive cost-penalty search to rapidly repair and enhance paths in dynamic settings. The mechanism includes the following:

1. Waypoint Cache: After each successful plan, store its sequence of states in a fixed-size cache. During re-planning, with probabilityP_{wp}, bias the planner's "target" toward one of these cached waypoints, reusing useful sub-paths.
2. Adaptive Cost-Penalty Search: Introduce a dynamic penalty (beta) on the distance-to-root component of the nearest-neighbor metric, increase beta when improving a cached plan to shorten paths, and decrease it when planning fails to maintain exploration.
3. Interleaved Execution: Execute the current best path incrementally while the planner continues to refine and repair it in the background, ensuring that the robot never stalls completely [60].

Wang et al. [61] proposed a novel and efficient path-planning algorithm using the B-RRT search. The RRT extension function was used to arrange the sampled states according to kinodynamic constraints. Additionally, Wang et al. [62] introduced the B-RRT* algorithm that seeks to achieve faster convergence and more efficient paths. To address the challenges of path planning in intricate environments with concavities, for complex scenarios involving convex regions, narrow channels, maze-like paths, and several obstacles, an adaptive bidirectional RRT* (AEB-RRT*) extension was introduced.

5.3. The Hybrid RRT* and the RRT*-D*Lite

Combining RRT*'s global sampling with D*Lite incremental graph search yields a two-layer scheme. The RRT* constructs a global roadmap; when obstacles invalidate edges, the D* Lite repairs only the impacted region. This hybrid algorithm excels in highly dynamic workspaces where frequent minor updates occur [62]-[64]. Li et al. [65] suggested an enhanced D Lite algorithm to resolve issues related to sharp turns, redundant nodes, and proximity to obstacles in the path, thereby improving the path's safety. The D Lite algorithm defines a safety buffer between the vehicle and the obstacles. Hameed et al. [66] presented an APF-IRRT*-HS method, combining RRT*, APF, and Halton sequence for optimized path planning in dynamic environments, showing improvements in sampling efficiency, path length, and computation time. Moreover, Hameed et al. [67] proposed a hybrid algorithm, namely, the APF-IRRT*-SB, which integrates Artificial Potential Field (APF) with RRT*, enhancing computational efficiency and path quality [68]. The method showed significant improvements in both path length and computation time. Lin and Zhang [69] proposed an HBAI-

RRT* to overcome the limitations of finding optimal solutions, which restricts their use in specific scenarios, particularly in complex maze environments. Simultaneously, a dynamic modification approach for the segment intersection set was used to improve the overall path. Su et al. [70] proposed a modified RRT* algorithm, FFC-RRT*, to overcome the high degree of randomness in RRT*, which leads to a slow convergence of the initial solution and a suboptimal starting solution. The restricted sampling method was presented, enabling the random tree to be sampled within a defined area, and an adaptive combined sampling method was applied, helping the algorithm to better accommodate various environments [48].

A two-layer planner was proposed that marries RRT*'s asymptotically-optimal tree construction with D* Lite's efficient incremental graph search, enabling global optimality and fast local re-planning under dynamic obstacles.

The mechanism includes the following:

1. Global Roadmap (RRT*) builds an initial, asymptotically-optimal tree in the configuration space.
2. Local Repair (D* Lite) maintains a sparse graph over the same nodes; when obstacles move or new obstacles appear, the D* Lite incrementally updates only the affected edges to restore a collision-free, lowest-cost path.
3. The execution loop alternates between tree expansion (to improve optimality) and the D* Lite repairs (to adapt to changes), ensuring both path quality and responsiveness.

6. Post-Processing and Path Smoothing

After the RRT* or similar sampling-based planners find a path, the raw output path is usually a sequence of connected waypoints (nodes) forming a piecewise linear path. This path often contains unnecessary waypoints, zigzags, or detours due to the random nature of sampling and connection constraints. Post-processing involves applying techniques after path generation to improve this raw path by removing redundant waypoints, shortening the total length, and simplifying the path shape while maintaining collision-free constraints. On the other hand, path smoothing is a subset of post-processing focused on converting the jagged piecewise linear path into a smooth, continuous curve. This step is essential because real robots cannot instantaneously change direction sharply; smooth paths reduce mechanical stress, energy consumption, and tracking errors, and smooth trajectories are easier to follow with control algorithms. A collection of techniques are applied after the RRT* tree has been built to transform the raw waypoint sequence into a smooth, dynamically feasible trajectory suitable for execution.

Shortcut Smoothing: Iteratively attempt direct connections between non-consecutive waypoints; if collision-free and cost-reducing, remove the intermediate nodes.

1. CHOMP: Covariant Hamiltonian Optimization for Motion Planning uses functional gradients to minimize a cost combining trajectory smoothness and obstacle clearance, converging to low-cost, smooth paths even from infeasible seeds. Pamarthi and Agrawal [71] presented an optimized motion-planning framework for UAVs navigating dynamic environments, integrating an enhanced RRTX method with a CHOMP-based optimizer. It includes a dynamic scenario generator using a UAV simulator and barrier information to emulate obstacles and flight patterns.
2. STOMP: Stochastic Trajectory Optimization for Motion Planning generates and evaluates noisy perturbations around an initial path, combining successful samples to overcome local minima and optimize arbitrary cost functions without gradient information. Ecker et al. [72] presented an optimized global kinodynamic path planning for timber cranes utilizing the newly introduced via-point-based stochastic trajectory optimization method. They validated the effectiveness of their method by comparing it with the LQR-RRT*.

3. Spline-Based Fitting: Fit Bézier or B-spline curves through the waypoints optionally enforcing velocity/acceleration continuity and joint limits. Eshtehardian and Khodaygan [73], [74] proposed an approach based on the merge of RRT* and B-spline for smoothing the path generated by the RRT* algorithms. To avoid collision in the generated path, some corrections were also provided. Zhang and Gong [75] proposed an S-BRRT*, which is a novel pruning strategy put forward to conduct intelligent pruning of BRRT*'s zigzag path, including pruning under collision and path constraint to select optimal path nodes. Then, the smooth strategy was proposed using a cubic Bézier curve to fit the previous pruning path for the first time. Then, they fit for the second time under collision optimization to ensure the final feasible smooth path. Fei et al. [76] introduced a novel Informed SRRT path planning algorithm for robotics. This algorithm merges a local planner from SRRT, handling both external and internal constraints. They added two additional lines at the Bézier spline's endpoints, aiding the rewiring process.

Feng et al. [77] proposed a new autonomous underwater vehicle path planning approach based on the B-spline RRT*, which overcomes the limitations of the RRT* by concentrating on planning optimal paths within maximum curvature limits, thereby significantly enhancing path smoothness. [78] studied motion planning for robots in industrial environments, particularly in obstacle-filled spaces. They proposed the SDA-RRT*Connect method, which dynamically modifies the search direction throughout the planning process, and a path process technique to decrease path length and improve smoothness.

Existing RRT* surveys are primarily narrative, with uneven coverage of informed/batch/learning-guided variants, kinodynamic and safety-aware planning, multi-robot settings, and hardware-aware (parallel/GPU) implementations. Reproducibility and comparability are limited by unclear search/screening/coding protocols and inconsistent metrics, environments, and hardware, in addition to a few standardized benchmarks or stress tests. However, practical guidance is scarce. In particular, prior reviews rarely map constraints and compute budgets to suitable RRT* variants, motivating our structured taxonomy, quantitative trend synthesis, practitioner decision map, and reporting checklist.

7. Results and Discussion

This section includes brief descriptions of the articles used in our systematic literature review. Specifically, a summary of each paper is provided, including RRT* Improvement, advantages, limitations, results, and suggestions for improving path planning for future work, as described in the Table 1.

Table 1. RRT* Improvements: Advantages, Limitations, Results, and Future Directions

Reference No.	Advantage	Limitation	Type of RRT *Improvement	Results	Future improvement
[1]	Optimized for nuclear decommissioning environments	Computational complexity in highly dynamic settings	Bi-RRT* with variable node parameters	More efficient path planning in constrained spaces	Enhance real-time adaptability in dynamic environments.
[2]	Improved for multi-obstacle, narrow environments	Computationally expensive for larger environments	Improved RRT*-Connect	Better performance in tight spaces	Explore hybrid algorithms for faster convergence in dense environments.
[3]	UAV path planning with dynamic windows and informed RRT*	Inefficient for highly dynamic or 3D environments	Informed RRT* + Dynamic Window	More efficient path finding for UAVs	Incorporate adaptive search strategies for real-time path corrections.
[4]	Comprehensive comparison of traditional RRT* methods	Limited practical examples	Overview of RRT* methods	Useful in understanding RRT* strengths and weaknesses	Apply hybrid methods to improve real-world applicability.

Reference No.	Advantage	Limitation	Type of RRT *Improvement	Results	Future improvement
[5]	Review of RRT* for single and multiple robots	Lack of focus on dynamic obstacles	Systematic review of RRT*	Broad overview of RRT* uses in robotics	Focus on adapting RRT* for real-time dynamic obstacle avoidance.
[6]	Improved performance in narrow passages	Limited for highly dynamic or large environments	RJ-RRT for narrow passages	More efficient path planning for narrow spaces	Optimize for real-time obstacle detection and re-planning.
[7]	Kinodynamic motion planning with motion primitives	Complexity in high-dimensional spaces	IDB-RRT with motion primitives	Improved planning for dynamic systems	Extend to larger, more complex scenarios.
[8]	Enhanced for insect-like mobile robots in narrow spaces	Potential inefficiency in very dynamic environments	Insect-like robot path planning with RRT*	Optimized for insect-like robots	Add adaptive re-planning for more dynamic environments.
[9]	Topological guidance for RRT-based motion planning	Complexity in dynamic environments	HAS-RRT with topological guidance	More efficient path finding using topological data	Enhance real-time adjustment in unpredictable environments.
[10]	Asymptotically optimal single-query path planning	May struggle with multi-query or dynamic environments	Informed RRT*-Connect	Optimal solutions in static scenarios	Integrate dynamic obstacle handling for real-time re-planning.
[11]	Path optimization for mobile robots	Computationally expensive in real-time systems	Improved Informed RRT*	Better trajectory optimization for robots	Optimize for real-time feedback and dynamic adjustments.
[12]	Faster, more robust initial path via gridmap skeletonization	Depends on gridmap/skeleton quality; shown mainly for 2D mobile robots in simulation	Informed RRT* with structure-aware seeding	Shorter computation time with smaller standard deviation	Extend to dynamic maps/kinodynamics and 3D, integrate online replanning
[13]	Optimized RRT for mobile robot path planning	May be less efficient in large spaces	Optimized Informed RRT*	Improved efficiency in constrained spaces	Apply dynamic re-planning in larger, unpredictable environments.
[14]	Dual tree search for better heuristic guidance	Complexity and computational overhead	Dynamic Informed Bias RRT*	Better performance in dynamic environments	Use adaptive heuristics for faster convergence.
[15]	Improved autonomous navigation with DWA	May be limited in highly dynamic spaces	Informed RRT* + DWA	Enhanced navigation for robots	Incorporate machine learning for adaptive path adjustments.
[16]	Bridge Test for RRT global path planning	Limited in highly dynamic environments	Bridge Test-based RRT*	Improved pathfinding in constrained environments	Improve scalability for larger environments with frequent changes.
[17]	Efficient dynamic bridging for RRT*	Struggles with highly dynamic or large environments	Dynamic Bridging RRT*	Faster path finding and more efficient planning	Improve adaptability for real-time obstacle avoidance.
[18]	Bidirectional search for improved RRT*	Computationally expensive in higher dimensions	Bi-RRT*	More efficient path finding in 2D spaces	Add real-time adjustments for dynamic obstacles.
[19]	Improved sampling for dynamic environments	Limited to specific environments	Bi-AM-RRT*	Faster motion planning in dynamic settings	Integrate adaptive re-planning for larger-scale scenarios.

Vol. 5, No. 4, 2025, pp. 2118-2139

Reference No.	Advantage	Limitation	Type of RRT *Improvement	Results	Future improvement
[20]	Enhanced path planning for ship routes	May not be effective for general robotics	RRT* with biased sampling	Improved planning in high-risk areas	Extend to more generalized path planning scenarios with real-time updates.
[21]	Improved UAV trajectory planning	May struggle with very dynamic or large environments	Bi-directional APF-RRT* with goal bias	Enhanced UAV planning in obstacle-rich environments	Incorporate real-time re-planning to handle dynamic changes.
[22]	Efficient autonomous vehicle path planning	Limited scalability in high-dimensional dynamic environments	Improved Bi-RRT for dynamic obstacle avoidance	Better performance in dynamic environments	Add adaptive sampling and dynamic re-planning.
[23]	Optimized RRT sampling region	May not work well in environments with high-dimensional spaces	RRT with improved sampling region	Better coverage for path planning	Incorporate dynamic re-planning and real-time obstacle prediction.
[24]	Review of kinodynamic motion planning	Limited to theoretical analysis	Kinodynamic RRT* review	Summary of kinodynamic motion planning improvements	Extend to real-world applications and larger systems.
[25]	Path planning and obstacle avoidance for UAVs	May be inefficient in large-scale environments	Fuzzy Kinodynamic RRT	Better dynamic path planning for UAVs	Improve real-time obstacle avoidance and adaptive behavior.
[26]	Accelerated kinodynamic RRT*	Computational complexity for high-dimensional problems	Dimensionality reduction in kinodynamic RRT*	Faster convergence in high-dimensional spaces	Add hybrid techniques to improve scalability.
[27]	Template-based RRT for efficient motion planning	Limited by environmental complexity	Template-based RRT	More efficient planning for kinodynamic systems	Expand to handle dynamic environments and real-time changes.
[28]	Multi-goal motion planning	Inefficient for very dynamic environments	KRRF: Kinodynamic RRT Forest	Better path planning for multi-goal systems	Enhance adaptability for dynamic environments.
[29]	Efficient trajectory planning for UAVs	May struggle with highly dynamic or 3D environments	Kinodynamic RRT for fixed-wing UAV	More efficient planning for fixed-wing UAVs	Incorporate real-time adjustments for moving obstacles.
[30]	Path planning for quadrotor kinodynamic systems	Computational cost in high-risk environments	Vector Field Guided RRT*	More efficient motion planning for quadrotors	Improve real-time re-planning in dynamic environments.
[31]	Faster path planning with high-quality paths	Limited applicability for dynamic environments	FHQ-RRT*	Faster planning for mobile robots	Integrate adaptive sampling for improved performance in dynamic settings.
[32]	Safe and optimal motion planning	Complexity in dynamic and unpredictable environments	LQR-CBF-RRT*	Safer path planning for robotic systems	Add dynamic obstacle prediction and real-time adjustments.
[33]	Path planning and tracking control for autonomous vehicles	May struggle in highly dynamic environments	Improved RRT* with PSO-LQR	Enhanced tracking control for	Add adaptive re-planning for real-time changes.

Reference No.	Advantage	Limitation	Type of RRT *Improvement	Results	Future improvement
				autonomous vehicles	
[34]	Motion planning for on-orbit assembly	May be limited outside specific applications	LQR-RRT* and MPC	Improved path planning for assembly tasks	Optimize for larger-scale systems and dynamic obstacles.
[35]	Motion planning for flexible spacecraft	Complexity in real-time adaptation	Enhanced LQR-RRT	Better path planning for flexible spacecraft	Add real-time adjustments for unpredictable environmental changes.
[36]	Preoperative path planning for surgical robots	High computational cost for real-time systems	MDP-LQR-RRT*	Enhanced surgical robot path planning	Optimize for real-time re-planning during surgery.
[37]	Spatiotemporal trajectory planning for autonomous vehicles	May not handle highly dynamic environments	Iterative LQR for spatiotemporal planning	Improved trajectory planning for autonomous vehicles	Incorporate dynamic obstacle handling and real-time re-planning.
[38]	RRT-Connect for robotic arm collision avoidance	May not perform well in highly dynamic spaces	RRT-Connect for robotic arm	Improved collision avoidance for robotic arms	Add dynamic re-planning to handle real-time obstacles.
[41]	Optimization for 3D smooth paths in robotic arm planning	May struggle in larger, more dynamic environments	Optimization Algorithm for 3D Paths	Smoother, more optimized paths for robotic arms	Improve real-time adjustments for unpredictable obstacles.
[42]	Collision-free path planning for robotic harvesters	Limited to specific types of robots and environments	Bi-RRT for robotic harvesting	More efficient path planning for robotic arms	Incorporate real-time dynamic obstacle avoidance.
[43]	Efficient collision checking with graph neural networks	May require significant training data and computational power	Collision checking using Graph Neural Networks	Reduced collision detection overhead	Enhance real-time adaptability and robustness.
[44]	Fast collision checking with hybrid safety certificate	May not handle very dynamic environments well	Hybrid safety certificate for fast collision checking	Faster motion planning with collision avoidance	Optimize for larger environments with dynamic obstacles.
[45]	Continuous collision checking for non-holonomic robots	High computational cost for complex environments	Continuous collision checking in narrow environments	More effective path planning for non-holonomic robots	Improve real-time collision detection for moving obstacles.
[46]	Exploiting collisions to enhance motion planning	May be inefficient in larger or highly dynamic environments	Collision-based motion planning for multicopters	More efficient path finding for multicopters	Integrate adaptive sampling for real-time path corrections.
[47]	Predictive path planning for dynamic environments	May struggle with highly dynamic, uncertain environments	Predictive RRT-based path planning	Improved navigation in dynamic environments	Add adaptive re-planning to handle dynamic obstacle movements.
[48]	Efficient robotic arm path planning with DAPF-RRT	May struggle with environments with high uncertainty	DAPF-RRT for robotic arm planning	Improved motion planning for robotic arms	Extend to more dynamic, unpredictable settings.
[49]	Real-time adaptive motion-reactive RRT	Potential complexity and higher computation time in dynamic environments	SMARRT for dynamic environments	Self-repairing and adaptive path finding	Improve real-time adjustments to highly dynamic obstacles.

Reference No.	Advantage	Limitation	Type of RRT *Improvement	Results	Future improvement
[50]	Comprehensive overview of robot path planning	Limited to static and controlled environments	RRT for static and dynamic environments	Broad perspective on RRT applications	Integrate hybrid techniques to handle dynamic obstacles in real-time.
[51]	Faster and feasible path planning for dynamic environments	May struggle with very high-dimensional spaces	Dynamic RRT for fast path planning	Improved path finding in complex, cluttered environments	Enhance re-planning in highly dynamic environments.
[52]	Path planning with RRT* and dynamic window approach	May struggle with real-time adjustments	RRT* with dynamic window approach	Efficient path planning in dynamic environments	Incorporate predictive algorithms for obstacle handling.
[53]	Adaptable manipulator path planning algorithm	May not be optimal for all robotic systems	RRT with environmental adaptation	Better performance in dynamic environments	Enhance real-time adjustments and obstacle avoidance.
[54]	Integrated RRT* and dynamic window approach for UAVs	Potential inefficiency in complex, dynamic environments	RRT* + Dynamic Window Approach for UAVs	More effective UAV motion planning	Implement better real-time adaptation for moving obstacles.
[55]	Real-time path replanning in dynamic environments	Computational cost may increase with environment complexity	Reverse Tree Guided Path Planning	Improved performance in dynamic and unpredictable environments	Integrate faster re-planning techniques for real-time applications.
[56]	Efficient dynamic path planning for high-DOF rover	May be computationally expensive in high-dimensional environments	RRT* with rewiring for rover path planning	More efficient pathfinding in narrow and cluttered spaces	Enhance real-time adaptation and scalability in complex environments.
[57]	Path planning for mobile robots in dense environments	May struggle with large-scale dynamic environments	RRT* with bridge guidance	Better performance in narrow and dense environments	Optimize for dynamic obstacle handling and real-time corrections.
[58]	3D RRT-based planner improvement	High computational cost for large environments	Direct-Steered-DRRT*	More effective 3D planning for robotic systems	Incorporate dynamic feedback for real-time adjustments.
[59]	Energy-efficient green ant colony optimization	May not be optimal for all types of dynamic environments	Green ant colony optimization for path planning	Enhanced energy efficiency in dynamic 3D environments	Improve dynamic re-planning to handle fast-moving obstacles.
[60]	Enhanced dynamic path planning and energy management for robots	Complexity and potential inefficiency in large environments	Enhanced RRT for dynamic path planning	More energy-efficient and faster path finding	Incorporate real-time obstacle prediction and planning.
[61]	Bidirectional-unidirectional RRT for efficient path planning	Potential complexity in high-dimensional or large environments	Bidirectional-unidirectional RRT extension	Faster path planning for robotic systems	Add dynamic feedback for real-time obstacle adjustments.
[62]	Adaptive extension for bidirectional RRT*	Complexity may increase in large-scale environments	AEB-RRT*	Improved efficiency in dynamic environments	Optimize for faster re-planning in large-scale applications.
[63]	Ultra-high speed generation of initial paths	May struggle with large, highly dynamic environments	DBVS-APF-RRT* for fast path generation	Enhanced path generation speed and optimality	Improve adaptability to larger environments with

Reference No.	Advantage	Limitation	Type of RRT *Improvement	Results	Future improvement
			and optimal quality		frequent dynamic obstacles.
[64]	Improved route planning for ships in Arctic waters	Limited scalability in highly dynamic, large-scale environments	D* Lite-based dynamic route planning	Better performance in specific environmental constraints	Add predictive path adjustments for changing environmental conditions.
[65]	Enhanced path planning for intelligent vehicles	May be less efficient in real-time applications	Improved D* Lite for intelligent vehicles	Improved real-time planning for autonomous vehicles	Incorporate adaptive re-planning for real-time dynamic obstacles.
[66]	Enhanced path planning for robots	Complexity for real-time, dynamic obstacle handling	APF and Halton Sequence for improved RRT*	More efficient pathfinding for robots	Add dynamic obstacle prediction and faster re-planning capabilities.
[67]	Hybrid path planning for robotics	Increased complexity in path finding	APF-RRT enhancement with Sobol sequence	More efficient path finding in complex environments	Optimize for real-time adjustments in dynamic environments.
[69]	Improved convergence rate	Limited in highly dynamic or high-dimensional spaces	Hybrid bidirectional search with adaptive adjustments	Improved performance in narrow and cluttered spaces	Improve real-time re-planning for dynamic environments.
[70]	Fast initial solution convergence	Limited to environments that require quick initial paths	Adaptive hybrid sampling for rapid convergence	Faster initial path generation with higher-quality paths	Extend to handle real-time dynamic obstacle avoidance.
[71]	UAV path planning with hybrid motion algorithm	Potential inefficiency in large-scale or 3D spaces	Hybrid motion algorithm for UAVs	Improved obstacle avoidance for UAVs	Incorporate adaptive strategies for highly dynamic and 3D environments.
[72]	Global kinodynamic planning	May not perform well in large, high-dimensional spaces	Kinodynamic motion planning for under actuated systems	Improved planning for timber cranes	Add real-time re-planning for dynamic obstacles in large-scale environments.
[73]	Continuous path planning for non-holonomic robots	High computational cost for real-time adjustments	Continuous RRT* with B-spline curves	More effective for non-holonomic robots	Optimize for more complex environments with dynamic obstacles.
[74]	Path planning for mobile robots in dynamic environments	May struggle with very large or high-risk areas	Analytic geometry and cubic Bézier curve for dynamic environments	More efficient path planning for mobile robots	Improve adaptability for dynamic obstacles in large-scale environments.
[75]	Improved bidirectional RRT with strategies under nonholonomic constraints	Limited scalability in complex environments	S-brrt* with bidirectional RRT for nonholonomic constraints	Improved motion planning for nonholonomic robots	Add real-time adjustments for dynamic obstacles.
[76]	Asymptotically optimal path planning with Bézier splines	May not be applicable to all types of dynamic environments	Bézier spline-based path planning	Optimized for smooth paths	Incorporate real-time path corrections for dynamic environments.
[77]	Smooth path planning under curvature constraints	Limited for highly dynamic environments	B-spline curves for smooth path planning	Improved planning for underwater vehicles	Improve real-time adaptive planning for moving obstacles.

Reference No.	Advantage	Limitation	Type of RRT *Improvement	Results	Future improvement
[78]	Trajectory optimization for robotic manipulators	May require significant computational resources	SDA-RRT* Connect for robotic manipulators	Improved efficiency in industrial environments	Add adaptive re-planning for real-time obstacle avoidance in industrial settings.

8. Conclusion

The RRT*'s elegant theoretical foundations have spurred a rich ecosystem of practical enhancements. Guided sampling accelerates convergence, kinodynamic extensions enforce feasibility, lazy collision checking slashes overhead, dynamic planners enable online adaptation, and post-processing delivers smooth, safe trajectories. A next-generation planner that unifies these strands will bring sampling-based methods ever closer to deployment on real robotic systems operating in dynamic, cluttered environments, and an extensible survey becomes both a compass showing where the field stands and a launchpad highlighting the most fertile directions for tomorrow's innovations. Most kinodynamic and dynamic re-planning methods target low-DOF mobile robots. Scaling LQR-based or hybrid schemes to 7–10 DOF arms remains largely unexplored. Recent research works in learning priors or value-function approximations could inform sampling and steering in complex spaces. In this context, integrating neural samplers with RRT* variants is a promising direction. Composing guided sampling, lazy-collision checks, kinodynamic steering, and incremental re-planning into a single real-time pipeline demands careful orchestration of computation and memory.

In this regard, we envision a modular architecture in which a sampling manager blends informed, learned, and bridge-test samplers, a steer module that switches between straight-line, LQR, or learned controllers based on local dynamics, a collision supervisor that performs lazy and batched checks, a re-planner that invokes DRRT or D* Lite on-the-fly when map updates arrive, and finally a smoother (CHOMP/STOMP) that refines the executed trajectory in the background. Accordingly, our recommendations include a modular real-time RRT*: blend informed/learned/bridge-test sampling; adaptive steering (straight-line/LQR/learned); lazy batched collision checks; incremental re-planning (RRTX/DRRT/D* Lite); and background smoothing (CHOMP/STOMP). In addition, scaling up to 7–10 DOF manipulators with stable kinodynamic steering and warm starts should be considered, as well as using learning-guided priors/value functions with certified fallbacks and distribution-shift tests.

Author Contribution: All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.

Funding: This research received no external funding.

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

References

[1] J. O. Adibeli, Y.-K. Liu, N. Chao, and N. J. Awodi, "Modified bidirectional rapidly exploring random tree star (Bi-RRT*) algorithm with variable node parameter for optimized path planning in nuclear decommissioning environment," *Nuclear Engineering and Design*, vol. 433, p. 113876, 2025, https://doi.org/10.1016/j.nucengdes.2025.113876.

[2] X. He, Y. Zhou, H. Liu, and W. Shang, "Improved RRT*-Connect Manipulator Path Planning in a Multi-Obstacle Narrow Environment," *Sensors*, vol. 25, no. 8, p. 2364, 2025, https://doi.org/10.3390/s25082364.

[3] T. Wu, Z. Zhang, F. Jing, and M. Gao, “A dynamic path planning method for uavs based on improved informed-rrt* fused dynamic windows,” *Drones*, vol. 8, no. 10, p. 539, 2024, https://doi.org/10.3390/drones8100539.

[4] Y. Chu, Q. Chen, and X. Yan, “An Overview and Comparison of Traditional Motion Planning Based on Rapidly Exploring Random Trees,” *Sensors*, vol. 25, no. 7, p. 2067, 2025, https://doi.org/10.3390/s25072067.

[5] D. K. Muhsen, F. A. Raheem, and A. T. Sadiq, “A Systematic Review of Rapidly Exploring Random Tree RRT Algorithm for Single and Multiple Robots,” *Cybernetics and Information Technologies*, vol. 24, no. 3, pp. 78-101, 2024, https://doi.org/10.2478/cait-2024-0026.

[6] Q. Chai and Y. Wang, “RJ-RRT: improved RRT for path planning in narrow passages,” *Applied Sciences*, vol. 12, no. 23, p. 12033, 2022, https://doi.org/10.3390/app122312033.

[7] J. Ortiz-Haro, W. Hönig, V. N. Hartmann, M. Toussaint and L. Righetti, “iDb-RRT: Sampling-based Kinodynamic Motion Planning with Motion Primitives and Trajectory Optimization,” *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10702-10709, 2024, https://doi.org/10.1109/IROS58592.2024.10802168.

[8] L. Wang, X. Yang, Z. Chen, and B. Wang, “Application of the improved rapidly exploring random tree algorithm to an insect-like mobile robot in a narrow environment,” *Biomimetics*, vol. 8, no. 4, p. 374, 2023, https://doi.org/10.3390/biomimetics8040374.

[9] D. Uwacu, A. Yammanuru, K. Nallamotu, V. Chalasani, M. Morales and N. M. Amato, “HAS-RRT: RRT-Based Motion Planning Using Topological Guidance,” *IEEE Robotics and Automation Letters*, vol. 10, no. 6, pp. 6223-6230, 2025, https://doi.org/10.1109/LRA.2025.3560878.

[10] R. Mashayekhi, M. Y. I. Idris, M. H. Anisi, I. Ahmedy and I. Ali, “Informed RRT*-Connect: An Asymptotically Optimal Single-Query Path Planning Method,” *IEEE Access*, vol. 8, pp. 19842-19852, 2020, https://doi.org/10.1109/ACCESS.2020.2969316.

[11] L. Yuan, J. Zhao, W. Li, and J. Hou, “Improved informed-RRT* based path planning and trajectory optimization for mobile robots,” *International Journal of Precision Engineering and Manufacturing*, vol. 24, no. 3, pp. 435-446, 2023, https://doi.org/10.1007/s12541-022-00756-6.

[12] H. Ryu and Y. Park, “Improved informed RRT* using gridmap skeletonization for mobile robot path planning,” *International Journal of Precision Engineering and Manufacturing*, vol. 20, no. 11, pp. 2033-2039, 2019, https://doi.org/10.1007/s12541-019-00224-8.

[13] B. B. Maseko, C. E. van Daalen, and J. Treurnicht, “Optimised informed RRTs for mobile robot path planning,” *IFAC-PapersOnLine*, vol. 54, no. 21, pp. 157-162, 2021, https://doi.org/10.1016/j.ifacol.2021.12.027.

[14] H. Li, Y. Kang, and H. Han, “Dynamic Informed Bias RRT*-Connect: Improving Heuristic Guidance by Dynamic Informed Bias Using Hybrid Dual Trees Search,” *Journal of Intelligent & Robotic Systems*, vol. 110, no. 3, p. 106, 2024, https://doi.org/10.1007/s10846-024-02144-w.

[15] J. Dai, D. Li, J. Zhao, and Y. Li, “Autonomous Navigation of Robots Based on the Improved Informed-RRT∗ Algorithm and DWA,” *Journal of Robotics*, vol. 2022, no. 1, p. 3477265, 2022, https://doi.org/10.1155/2022/3477265.

[16] H. Tu, Y. Deng, Q. Li, M. Song, and X. Zheng, “Improved RRT global path planning algorithm based on Bridge Test,” *Robotics and Autonomous Systems*, vol. 171, p. 104570, 2024, https://doi.org/10.1016/j.robot.2023.104570.

[17] S. Qiu, B. Li, R. Tong, X. He, and C. Tang, “Efficient Path Planning Based on Dynamic Bridging Rapidly Exploring Random Tree,” *Applied Sciences*, vol. 14, no. 5, p. 2032, 2024, https://doi.org/10.3390/app14052032.

[18] B. Wang, D. Ju, F. Xu, and C. Feng, “Bi-RRT*: an improved bidirectional RRT* path planner for robot in two-dimensional space,” *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 18, no. 10, pp. 1639-1652, 2023, https://doi.org/10.1002/tee.23898.

[19] 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 Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 1282-1293, 2024, https://doi.org/10.1109/TIV.2023.3307283.

[20] C. Liu *et al*., "An enhanced RRT* algorithm with biased sampling and dynamic stepsize strategy for ship route planning in the high-risk areas," *Ocean Engineering*, vol. 332, p. 121466, 2025, https://doi.org/10.1016/j.oceaneng.2025.121466.

[21] J. Fan, X. Chen, and X. Liang, "UAV trajectory planning based on bi-directional APF-RRT* algorithm with goal-biased," *Expert systems with applications*, vol. 213, p. 119137, 2023, https://doi.org/10.1016/j.eswa.2022.119137.

[22] X. Zhang, T. Zhu, L. Du, Y. Hu, and H. Liu, "Local path planning of autonomous vehicle based on an improved heuristic bi-rrt algorithm in dynamic obstacle avoidance environment," *Sensors*, vol. 22, no. 20, p. 7968, 2022, https://doi.org/10.3390/s22207968.

[23] X. Jiang, Z. Wang, and C. Dong, "A Path Planning Algorithm Based on Improved RRT Sampling Region," *Computers, Materials & Continua*, vol. 80, no. 3, pp. 4303-4323, 2024, https://doi.org/10.32604/cmc.2024.054640.

[24] J. Zhang, "Kinodynamic Motion Planning for Robotics: A Review," *2021 5th International Conference on Robotics and Automation Sciences (ICRAS)*, pp. 75-83, 2021, https://doi.org/10.1109/ICRAS52289.2021.9476660.

[25] L. Chen, I. Mantegh, T. He and W. Xie, "Fuzzy Kinodynamic RRT: a Dynamic Path Planning and Obstacle Avoidance Method," *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 188-195, 2020, https://doi.org/10.1109/ICUAS48674.2020.9213964.

[26] D. Zheng and P. Tsiotras, "Accelerating Kinodynamic RRT* Through Dimensionality Reduction," *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3674-3680, 2021, https://doi.org/10.1109/IROS51168.2021.9636754.

[27] S. Yang, M. Zhao, and M. Shimosaka, "Improved Template-Based RRT for Efficient Kinodynamic Motion Planning," *The Proceedings of JSME annual Conference on Robotics and Mechatronics (Robomec) 2023*, 2023, https://doi.org/10.1299/jsmermd.2023.2A1-D18.

[28] P. Ježek, M. Minařík, V. Vonásek and R. Pěnička, "KRRF: Kinodynamic Rapidly-Exploring Random Forest Algorithm for Multi-Goal Motion Planning," *IEEE Robotics and Automation Letters*, vol. 9, no. 12, pp. 10724-10731, 2024, https://doi.org/10.1109/LRA.2024.3478570.

[29] J. Ge, L. Liu, X. Dong and W. Tian, "Trajectory Planning of Fixed-wing UAV Using Kinodynamic RRT Algorithm," *2020 10th International Conference on Information Science and Technology (ICIST)*, pp. 44-49, 2020, https://doi.org/10.1109/ICIST49303.2020.9202213.

[30] Z. Tang, B. Chen, R. Lan, and S. Li, "Vector field guided rrt* based on motion primitives for quadrotor kinodynamic planning," *Journal of Intelligent & Robotic Systems*, vol. 100, pp. 1325-1339, 2020, https://doi.org/10.1007/s10846-020-01231-y.

[31] X. Dong, Y. Wang, C. Fang, K. Ran, and G. Liu, "FHQ-RRT*: An Improved Path Planning Algorithm for Mobile Robots to Acquire High-Quality Paths Faster," *Sensors*, vol. 25, no. 7, p. 2189, 2025, https://doi.org/10.3390/s25072189.

[32] G. Yang, M. Cai, A. Ahmad, A. Prorok, R. Tron, and C. Belta, "LQR-CBF-RRT*: Safe and optimal motion planning," *arXiv*, 2023, https://doi.org/10.48550/arXiv.2304.00790.

[33] Y. Zhang, F. Gao, and F. Zhao, "Research on path planning and tracking control of autonomous vehicles based on improved RRT* and PSO-LQR," *Processes*, vol. 11, no. 6, p. 1841, 2023, https://doi.org/10.3390/pr11061841.

[34] B. Doerr and R. Linares, "Motion planning and control for on-orbit assembly using LQR-RRT* and nonlinear MPC," *arXiv*, 2020, https://doi.org/10.48550/arXiv.2008.02846.

[35] X. Zhong, Z. Wei and T. Chen, "Motion Planning and Pose Control for Flexible Spacecraft Using Enhanced LQR-RRT*," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 59, no. 6, pp. 8743-8751, 2023, https://doi.org/10.1109/TAES.2023.3309612.

[36] Z. Liu, M. Li, R. Zhang, G. Zhang, S. Han, and K. Chen, "Preoperative path planning of craniotomy surgical robot based on improved MDP-LQR-RRT* algorithm," *Biomedical Signal Processing and Control*, vol. 105, p. 107647, 2025, https://doi.org/10.1016/j.bspc.2025.107647.

[37] Y. Liu, X. Pei, H. Zhou and X. Guo, “Spatiotemporal Trajectory Planning for Autonomous Vehicle Based on Reachable Set and Iterative LQR,” *IEEE Transactions on Vehicular Technology*, vol. 73, no. 8, pp. 10932-10947, 2024, https://doi.org/10.1109/TVT.2024.3371184.

[38] M. Cao, H. Mao, X. Tang, Y. Sun, and T. Chen, “A novel RRT*-Connect algorithm for path planning on robotic arm collision avoidance,” *Scientific Reports*, vol. 15, no. 1, p. 2836, 2025, https://doi.org/10.1038/s41598-025-87113-5.

[39] M. L. Muhammed, E. H. Flaieh, and A. J. Humaidi, “Embedded system design of path planning for planar manipulator based on chaos A* algorithm with known-obstacle environment,” *Journal of Engineering Science and Technology*, vol. 17, no. 6, pp. 4047-4064, 2022, https://jestec.taylors.edu.my/Vol%2017%20Issue%206%20December%202022/17_6_22.pdf.

[40] 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 Transactions on Mechatronics*, vol. 27, no. 6, pp. 4774-4785, 2022, https://doi.org/10.1109/TMECH.2022.3165845.

[41] L. Ren, Y. Kang, L. Yang, H. Jia, and S. Wang, “Optimization Algorithm for 3D Smooth Path of Robotic Arm in Dynamic Obstacle Environments,” *Applied Sciences*, vol. 15, no. 4, p. 2116, 2025, https://doi.org/10.3390/app15042116.

[42] Z. Liu, R. M. Sampurno, R. R. D. Abeyrathna, V. M. Nakaguchi, and T. Ahamed, “Development of a Collision-Free Path Planning Method for a 6-DoF Orchard Harvesting Manipulator Using RGB-D Camera and Bi-RRT Algorithm,” *Sensors*, vol. 24, no. 24, p. 8113, 2024, https://doi.org/10.3390/s24248113.

[43] C. Yu and S. Gao, “Reducing collision checking for sampling-based motion planning using graph neural networks,” *Advances in Neural Information Processing Systems*, 2021, https://proceedings.neurips.cc/paper/2021/file/224e5e49814ca908e58c02e28a0462c1-Paper.pdf.

[44] S. Shi, J. Chen and Y. Li, “Hybrid Safety Certificate for Fast Collision Checking in Sampling-Based Motion Planning,” *IEEE Robotics and Automation Letters*, vol. 8, no. 1, pp. 113-120, 2023, https://doi.org/10.1109/LRA.2022.3223021.

[45] D. Henning and G. Schildbach, “A Continuous Collision Checking Method for Non-holonomic Path Planning in Narrow Environments,” *2022 26th International Conference on System Theory, Control and Computing (ICSTCC)*, pp. 594-600, 2022, https://doi.org/10.1109/ICSTCC55426.2022.9931874.

[46] J. Zha and M. W. Mueller, “Exploiting collisions for sampling-based multicopter motion planning,” *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7943-7949, 2021, https://doi.org/10.1109/ICRA48506.2021.9561166.

[47] Z. Zhang, B. Qiao, W. Zhao, and X. Chen, “A predictive path planning algorithm for mobile robot in dynamic environments based on rapidly exploring random tree,” *Arabian Journal for Science and Engineering*, vol. 46, pp. 8223-8232, 2021, https://doi.org/10.1007/s13369-021-05443-8.

[48] Z. Wang, J. Tang, F. Yi, X. Ren, and K. Wang, “Research on path planning of robotic arms based on DAPF-RRT algorithm,” *PloS one*, vol. 20, no. 5, p. e0323734, 2025, https://doi.org/10.1371/journal.pone.0323734.

[49] Z. Shen, J. Wilson, R. Harvey, and S. Gupta, “Smarrt: Self-repairing motion-reactive anytime rrt for dynamic environments,” *arXiv*, 2021, https://doi.org/10.48550/arXiv.2109.05043.

[50] F. A. Raheem, S. M. Raafat, and S. M. Mahdi, “Robot path-planning research applications in static and dynamic environments,” *Earth Systems Protection and Sustainability*, vol. 1, pp. 291-325, 2022, https://doi.org/10.1007/978-3-030-85829-2_12.

[51] P. Zhao *et al.*, “Dynamic RRT: fast feasible path planning in randomly distributed obstacle environments,” *Journal of Intelligent & Robotic Systems*, vol. 107, no. 4, p. 48, 2023, https://doi.org/10.1007/s10846-023-01823-4.

[52] R. Zhang, L. Zhou, and Z. Liu, “Dynamic path planning for mobile robot based on RRT* and dynamic window approach,” *Journal of System Simulation*, vol. 36, no. 4, pp. 957-968, 2024, https://dc-china-simulation.researchcommons.org/journal/vol36/iss4/14/.

[53] Z. Tian, L. Li, Y. Wang, X. Wang and P. Zan, “An Environment Adaptation Algorithm Based on RRT for Manipulator Path Planning,” *2023 42nd Chinese Control Conference (CCC)*, pp. 2821-2825, 2023, https://doi.org/10.23919/CCC58697.2023.10240745.

[54] K. Jain and A. Kashyap, “Integrated RRT* and Dynamic Window Approach for UAV Obstacle Control,” *Aerotecnica Missili & Spazio*, pp. 1-14, 2025, https://doi.org/10.1007/s42496-025-00273-5.

[55] B. Cui, R. Cui, W. Yan, Y. Wang and S. Zhang, “RT-RRT: Reverse Tree Guided Real-Time Path Planning/Replanning in Unpredictable Dynamic Environments,” *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5380-5387, 2024, https://doi.org/10.1109/IROS58592.2024.10802722.

[56] S. Katiyar and A. Dutta, “Dynamic path planning over CG-Space of 10DOF Rover with static and randomly moving obstacles using RRT* rewiring,” *Robotica*, vol. 40, no. 8, pp. 2610-2629, 2022, https://doi.org/10.1017/S0263574721001843.

[57] J. Li, X. Zhou, J. Mao, and C. Zhang, “Dynamic RRT* with bridge guidance for robot path planing in dense and narrow passages environment,” *Journal of Control and Decision*, pp. 1-15, 2025, https://doi.org/10.1080/23307706.2025.2482816.

[58] M. S. Lopes, M. F. Silva, J. P. C. de Souza and P. Costa, “Direct-Steered-DRRT*: A 3D RRT-based planner improvement,” *2024 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pp. 100-105, 2024, https://doi.org/10.1109/ICARSC61747.2024.10535937.

[59] V. Sangeetha, R. Krishankumar, K. Ravichandran, and S. Kar, “Energy-efficient green ant colony optimization for path planning in dynamic 3D environments,” *Soft Computing*, vol. 25, no. 6, pp. 4749-4769, 2021, https://doi.org/10.1007/s00500-020-05483-6.

[60] R. Chitre and A. Sinha, “An Enhanced RRT based Algorithm for Dynamic Path Planning and Energy Management of a Mobile Robot,” *arXiv*, 2023, https://doi.org/10.48550/arXiv.2310.20590.

[61] J. Wang, W. Chi, C. Li and M. Q. -H. Meng, “Efficient Robot Motion Planning Using Bidirectional-Unidirectional RRT Extend Function,” *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 3, pp. 1859-1868, 2022, https://doi.org/10.1109/TASE.2021.3130372.

[62] X. Wang, J. Wei, X. Zhou, Z. Xia, and X. Gu, “AEB-RRT*: an adaptive extension bidirectional RRT* algorithm,” *Autonomous Robots*, vol. 46, no. 6, pp. 685-704, 2022, https://doi.org/10.1007/s10514-022-10044-x.

[63] Z. Feng, L. Zhou, J. Qi, and S. Hong, “DBVS-APF-RRT*: A global path planning algorithm with ultra-high speed generation of initial paths and high optimal path quality,” *Expert Systems with Applications*, vol. 249, p. 123571, 2024, https://doi.org/10.1016/j.eswa.2024.123571.

[64] T. Xu, H. Yang, J. Ma, K. Xiong, and Q. Hu, “An Improved D* Lite-Based Dynamic Route Planning Algorithm for Ships in Arctic Waters,” *Journal of Marine Science and Engineering*, vol. 12, no. 12, p. 2323, 2024, https://doi.org/10.3390/jmse12122323.

[65] X. Li, Y. Lu, X. Zhao, X. Deng, and Z. Xie, “Path planning for intelligent vehicles based on improved D* Lite,” *The Journal of Supercomputing*, vol. 80, no. 1, pp. 1294-1330, 2024, https://doi.org/10.1007/s11227-023-05528-1.

[66] M. T. Hameed, F. A. Raheem, and A. R. Nasser, “Enhanced RRT* with APF and Halton Sequence for Robot Path Planning,” *Journal of Robotics and Control (JRC)*, vol. 6, no. 2, pp. 493-513, 2025, https://doi.org/10.18196/jrc.v6i2.24921.

[67] M. T. Hameed, F. A. Raheem, and A. R. Nasser, “Hybrid Path Planning for Robotics: APF-RRT Enhancement with Sobol Sequence Integration,” *International Journal of Intelligent Engineering & Systems*, vol. 18, no. 3, pp. 556-568, 2025, https://doi.org/10.22266/ijies2025.0430.38.

[68] S. Azubairi, A. Petunin, H. L. Alwan, M. M. Msallam, and A. Humaidi, “Dynamic Processing 2D Maps Method for Robot’s Trajectory Planning,” *Proceedings of Engineering and Technology Innovation*, vol. 30, pp. 79-89, 2025, https://doi.org/10.46604/peti.2024.14508.

[69] Y. Lin and L. Zhang, “An improved Quick Informed-RRT* algorithm based on hybrid bidirectional search and adaptive adjustment strategies,” *Intelligent Service Robotics*, vol. 17, no. 4, pp. 847-870, 2024, https://doi.org/10.1007/s11370-024-00541-6.

[70] Z. Su, Q. Lv, Z. Huang, T. Yu, and X. Wang, "Ffc-rrt*: An adaptive hybrid sampling path planning algorithm with rapid initial solution convergence," *Intelligent Service Robotics*, vol. 18, pp. 499-512, 2025, https://doi.org/10.1007/s11370-025-00599-w.

[71] V. Pamarthi and R. Agrawal, "Unmanned aerial vehicle path planning with hybrid motion algorithm for obstacle avoidance," *Measurement: Sensors*, vol. 36, p. 101195, 2024, https://doi.org/10.1016/j.measen.2024.101195.

[72] M.-P. Ecker, B. Bischof, M. N. Vu, C. Fröhlich, T. Glück, and W. Kemmetmüller, "Global kinodynamic motion planning for an underactuated timber crane with stochastic trajectory optimization," *Mechatronics*, vol. 108, p. 103319, 2025, https://doi.org/10.1016/j.mechatronics.2025.103319.

[73] S. Eshtehardian and S. Khodaygan, "A continuous RRT*-based path planning method for non-holonomic mobile robots using B-spline curves," *Journal of ambient intelligence and humanized computing*, vol. 14, no. 7, pp. 8693-8702, 2023, https://doi.org/10.1007/s12652-021-03625-8.

[74] V. Bulut, "Path planning of mobile robots in dynamic environment based on analytic geometry and cubic Bézier curve with three shape parameters," *Expert Systems with Applications*, vol. 233, p. 120942, 2023, https://doi.org/10.1016/j.eswa.2023.120942.

[75] Y. Zhang and D. Gong, "S-BRRT*: A Spline-based Bidirectional RRT with Strategies under Nonholonomic Constraint," *2021 33rd Chinese Control and Decision Conference (CCDC)*, pp. 1753-1758, 2021, https://doi.org/10.1109/CCDC52312.2021.9602738.

[76] G. Vailland, V. Gouranton and M. Babel, "Cubic Bézier Local Path Planner for Non-holonomic Feasible and Comfortable Path Generation," *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7894-7900, 2021, https://doi.org/10.1109/ICRA48506.2021.9560854.

[77] H. Feng, Q. Hu, Z. Zhao, and X. Feng, "Smooth path planning under maximum curvature constraints for autonomous underwater vehicles based on rapidly-exploring random tree star with b-spline curves," *Engineering Applications of Artificial Intelligence*, vol. 133, p. 108583, 2024, https://doi.org/10.1016/j.engappai.2024.108583.

[78] G. Wu, P. Wang, B. Qiu, and Y. Han, "SDA-RRT* Connect: A Path Planning and Trajectory Optimization Method for Robotic Manipulators in Industrial Scenes with Frame Obstacles," *Symmetry*, vol. 17, no. 1, p. 1, 2024, https://doi.org/10.3390/sym17010001.