

REVIEW

A Review of Parking Trajectory Planning and Modeling Techniques for Autonomous Vehicles

Xianjian Jin^{1,2,*}, Yinchun Tao¹, Yuhuai Zhang¹, Haoze Wu¹, Jianning Lu¹ and Nonsly Valerieenne Opinat Ikiela¹

¹School of Mechatronic Engineering and Automation, Shanghai University, Shanghai, China

²Shanghai Key Laboratory of Intelligent Manufacturing and Robotics, Shanghai University, Shanghai, China

*Corresponding Author: Xianjian Jin. Email: xianjianjin@shu.edu.cn

Received: 19 March 2026; Accepted: 26 August 2026; Published: 28 September 2026

ABSTRACT: Autonomous parking is a key bottleneck to achieving fully autonomous driving, especially in the final parking problem of automated valet parking (AVP). Unlike traditional structured highway driving, parking scenarios impose stringent requirements on trajectory feasibility, and it requires the simultaneous resolution of nonholonomic motion constraints, narrow passage navigation, and collision avoidance in unstructured environments. This paper provides a comprehensive overview of parking trajectory planning and modeling techniques. Different aspects of parking trajectory planning strategies and modeling methodologies in recent literature are categorized into six major classes: graph-search-based methods, sampling-based methods, artificial potential field (APF)-based methods, numerical optimization-based methods, and geometry-based methods, artificial intelligence (deep learning and reinforcement learning)-based methods. The pros and cons of these methodologies are discussed. Finally, future research directions in this field are also provided.

KEYWORDS: Autonomous vehicles; autonomous parking; trajectory planning; parking planning; planning techniques

1 Introduction

In recent years, autonomous driving technology shown Fig. 1 has been extensively studied by academic institutions worldwide and widely adopted in various vehicle models, positioning itself as one of the most promising advancements in modern transportation [1]. Despite significant progress in highway cruise control and urban autonomous navigation, the “last-mile” challenge, particularly concerning Automated parking systems (APS) and Automated valet parking (AVP), remains a critical bottleneck that requires further breakthroughs. Parking maneuvers are typically conducted in confined and complex environments, where minor collisions with surrounding vehicles or obstacles frequently occur [2–4]. This issue poses a considerable challenge for a large number of drivers; in severe cases, it may lead to traffic congestion. Therefore, the automation of parking operations has gained substantial popularity, with a growing preference for vehicles equipped with autonomous parking capabilities to achieve a safer, more efficient, and more convenient parking experience [5].

Over the past few years, the historical trajectory of automated parking has evolved from passive proximity feedback to active vehicle control [6]. In the late 1990s, initial parking assistance was limited to ultrasonic sensors detecting obstacles and providing auditory or visual warnings to the driver [7]. By the early 21st century, the emergence of semi-automated parking assistance systems enabled automated steering

control while the driver managed longitudinal acceleration and braking [8]. Advances in sensor fusion and enhanced computational capacity later allowed parking systems to integrate throttle and brake control alongside steering [9]. In recent years, the shift toward vehicle electrification and intelligent connectivity has accelerated the integration of automated parking with advanced multi-sensor suites and smart cockpit systems [10]. The industry is now progressing rapidly toward Level 4 AVP, in which vehicles are capable of learning and navigating along pre-memorized routes from designated starting points to parking spaces without driver intervention [2,3]. Further development is focused on full AVP systems, a key research frontier. In such systems, upon reaching a parking facility entrance, the driver may exit the vehicle, which then autonomously proceeds to navigate, avoid obstacles, and park without further human input. Public demonstrations from industry leaders such as Tesla, Bosch, and Daimler, as well as competitions like the DARPA Urban Challenge, have further validated the technical feasibility of these systems under real-world conditions [11].

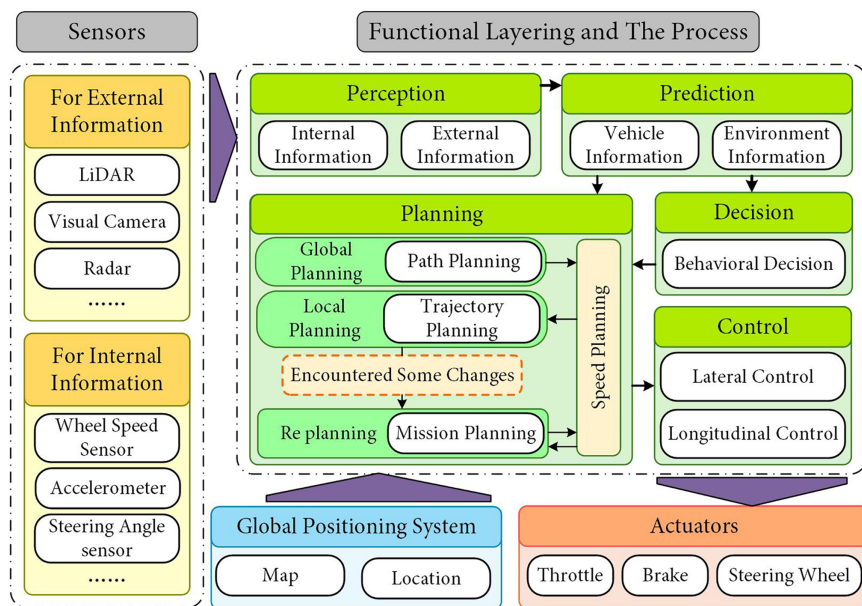


Figure 1: A hierarchical architecture of autonomous vehicle systems.

Although automated parking represents a current and popular research topic, the motion planning involved in parking operations is fundamentally distinct from other autonomous driving technologies, such as highway cruising [12–14]. Highway autonomous driving is generally confined to scenarios involving high speeds and low-curvature roads [15,16]. In contrast, automated parking motion planning must accommodate sharp-angle steering maneuvers and frequent gear shifts (forward/reverse) within highly constrained spaces [17]. Unlike the structured flow of highway environments, parking facilities consist of unstructured spaces filled with both static and dynamic obstacles [18–20]. This requires planning algorithms to solve non-convex optimization problems characterized by high collision risks. In summary, the challenges of automated parking can be classified into three main categories: environmental perception, decision-making and planning, and control execution [21,22]. Given that parking is a low-speed, close-proximity, and highly interactive scenario, it involves significant uncertainty regarding the behavior of pedestrians and other vehicles. Ambiguities in detecting parking space boundaries further complicate the task, presenting substantial challenges at the environmental perception level. At the decision-making and planning level, generating safe, efficient, and comfortable paths within tight spaces constitutes a major

engineering challenge [23,24]. Achieving an optimal balance between real-time performance and global optimality remains a critical concern. Finally, the accurate translation of planned paths into steering, throttle, and braking actions represents the most crucial step toward fully automated parking—a goal that still requires considerable advancement [25–28].

This paper presents a comprehensive review of parking trajectory planning technologies over the past decade, during which significant advancements have been achieved in automated parking systems. The core challenge lies in generating safe, smooth, and executable trajectories in real time, while adhering to vehicle kinematic constraints and relying on high-precision environmental perception and localization. In this context, state-of-the-art methods are classified into six distinct categories: graph search-based approaches (e.g., Hybrid A*), sampling-based methods (e.g., RRT*), artificial potential field techniques, numerical optimization-based strategies (e.g., Model Predictive Control, MPC), geometry-based methods (e.g., Dubins curves, B-spline curve) and emerging artificial intelligence-driven methods (e.g., deep learning and reinforcement learning).

(1) Graph Search-Based Methods

The core of graph-based search lies in transforming the parking problem within a continuous state space into an optimal path search between the initial and terminal nodes on a discrete map, and the search process is guided by heuristic functions to improve efficiency. Graph-based search methods offer resolution completeness; however, it should be noted that due to the discretization of the state space, a globally optimal solution in the continuous state space is not always available; e.g., Hybrid A* only generates near-globally optimal paths for automated parking applications.

(2) Sampling-Based Methods (e.g., RRT*)

Sampling-based methods operate by randomly sampling nodes in the continuous state space of the vehicle and attempting to connect them to an existing tree structure, gradually growing a tree that covers reachable parking configurations. Their main strength lies in strong adaptability to complex environments, and they also achieve probabilistic completeness as the number of samples approaches infinity. e.g., RRT* is among the primary sampling-based methods applied in parking planning.

(3) Artificial Potential Field Methods

The core concept of artificial potential field methods is to simulate virtual force fields: the target position generates an attractive force to guide the vehicle forward, while obstacles produce repulsive forces to avoid collisions. Although computationally efficient, these methods suffer from the well-known “local minima” problem, which must be addressed to ensure feasible parking trajectories can be found.

(4) Numerical Optimization-Based Methods

These methods formulate the trajectory planning problem directly as a constrained mathematical optimization problem. Constraints include vehicle dynamics, collision avoidance, initial and terminal state conditions, and actuator limits. This approach can generate smooth, comfortable, and highly accurate trajectories; however, the optimization process is often computationally intensive and time-consuming.

(5) Geometry-Based Methods

Geometry-based methods utilize parametrized geometric curves or polynomials to interpolate between the given initial and terminal states, as well as a set of intermediate states including position, orientation, discrete waypoints, and curvature conditions driven by task-specific requirements or generated by the algorithm, to produce a continuous path with specified smoothness properties. The main advantage is the

ability to generate curvature-continuous trajectories with very high computational efficiency. However, in complex environments, these methods may fail to produce a feasible path.

(6) Artificial Intelligence-Based Methods

Deep learning approaches learn an end-to-end mapping from perceptual inputs to planned trajectories using large-scale driving datasets. Reinforcement learning, on the other hand, learns planning policies through trial-and-error interactions within simulated environments, aiming to maximize cumulative rewards. These methods hold significant potential in handling high-dimensional sensory inputs, learning complex interactive behaviors, and discovering optimization strategies that are difficult to design manually.

The purpose of this review is to build a taxonomy of motion planning algorithms specifically for autonomous parking. The review is organized as follows: [Sections 2–7](#) critically review the six main families of algorithms. Each of these sections begins by detailing the mathematical formulation and logical architecture of the classical methods within the family, subsequently analyzes the inherent bottlenecks of applying these individual methods to parking scenarios, and finally categorizes recent advanced solutions based on these classical baselines. Subsequently, [Section 8](#) provides a comparative analysis of these algorithmic families based on multidimensional performance metrics, followed by a discussion on the engineering implementation challenges of relevant algorithms and future research direction. Lastly, [Section 9](#) systematically concludes the paper, it highlights the crucial role of hybrid methodologies and cross-layer cooperative architectures in achieving fully autonomous valet parking (AVP).

2 Graph Search-Based Algorithms

Graph search algorithms are a set of algorithms that discretize the search space into nodes and edges of a graph to calculate the shortest path from the start point to the target point. Relying on their reliability, they are widely applied to motion planning for automatic parking problems. Dijkstra's algorithm, A* algorithm, and their variants are two types of classic graph search algorithms.

2.1 Basic Dijkstra's Algorithm and A* Algorithm

Dijkstra's algorithm [29] and A* algorithm [30] the same core search idea: both expand the search scope by traversing child nodes adjacent to the parent node, and both define a cost function to evaluate the merit of child nodes. During the algorithm execution, the child node with the minimum cost among all candidate child nodes is selected as the new parent node, and this process is repeated until the target point is reached. The main difference between the two lies in the cost function: Dijkstra's algorithm with the flowchart shown in [Fig. 2](#) takes the cost function $g(n)$ into account, which is defined as the cumulative cost of all subpaths from the start to the current node n . As shown in [Eq. \(1\)](#), in which $g(\text{parent}(n))$ represents the cumulative cost along the sub-paths from the start node to the parent node of the current node n , while $\Delta g(n)$ stands for the incremental cost of transitioning from the parent node to node n , which is calculated as the Euclidean distance between the two nodes.

$$g(n) = g(\text{parent}(n)) + \Delta g(n) \quad (1)$$

As a latecomer, the A* algorithm not only takes $g(n)$ into account but also considers the estimated cost $h(n)$ from the current node to the target point, which is also called the heuristic function. The function is typically calculated by using the diagonal distance or Manhattan distance between the nodes. A comprehensive cost function $f(n)$ is established to synthetically consider these two cost components, serving as the evaluation criterion for child nodes in the A* algorithm, as shown in [Eq. \(2\)](#). Due to the inclusion

of the heuristic function, the target point exerts a guiding effect on the search process of the A* algorithm, resulting in significantly higher search efficiency than Dijkstra's algorithm, as illustrated in Figs. 3 and 4.

$$f(n) = g(n) + h(n) \quad (2)$$

$$\begin{cases} d_x = |x_2 - x_1| \\ d_y = |y_2 - y_1| \end{cases} \quad (3)$$

$$d_{diagonal} = D \cdot (d_x + d_y) + (\sqrt{2} - 2) D \cdot \min \{d_x, d_y\} \quad (4)$$

$$d_{Manhattan} = D \cdot (d_x + d_y) \quad (5)$$

$$d_{Euclidean} = D \cdot \sqrt{d_x^2 + d_y^2} \quad (6)$$

In automatic parking, Dijkstra's algorithm and A* algorithm are often used to plan paths to target parking spaces in parking lots, as shown in Fig. 5, but are rarely directly applied to plan parking maneuver paths. This is mainly due to the following two limitations: First, both algorithms perform path search based on static maps and lack the ability to handle dynamic obstacles, while parking processes often require interaction with pedestrians and other vehicles. Second, in narrow and complex unstructured parking scenarios, the vehicle's own dynamic constraints are crucial for path planning. However, both algorithms simplify the vehicle into a node on the map, and the generated path is a discrete polyline, which fails to meet the continuity requirements of vehicle dynamics.

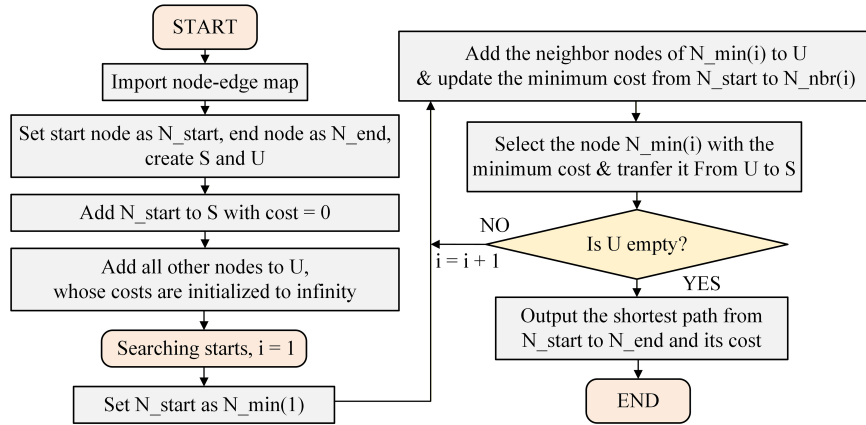


Figure 2: Flowchart of Dijkstra's algorithm for AVP.

In terms of recent research, taking the A* algorithm shown in Fig. 5 as an example: Liu et al. [31] proposed an enhanced A* algorithm for parallel parking scenarios. This algorithm combined the A* algorithm with the dynamic model to generate an initial trajectory with full-dimensional motion information, which served as the initial guess for gradient optimization algorithms. Leu et al. [32] adopted a Bidirectional A* Search Guided Tree (BIAGT) algorithm to generate global reference trajectories for dynamic parking scenarios. In addition, the A* algorithm was used to search for evacuation paths when maintaining the reference trajectory was no longer safe. He and Li [33] addressed the narrow-space parking problem by adding the heading angle dimension to nodes and proposing a fast A* algorithm. The algorithm searched for feasible nodes within the approaching pose range from far to near. Besides, pruning operations were incorporated, which significantly improved search efficiency. Gan et al. [34] focused on dynamic parking scenarios. Heading angle and time dimensions were added to nodes, and a 4D Spatio-temporal map was first

constructed. Then an improved A* algorithm based on RS curves and potential functions was proposed and used to generate global initial paths on the map, as shown in Fig. 6.

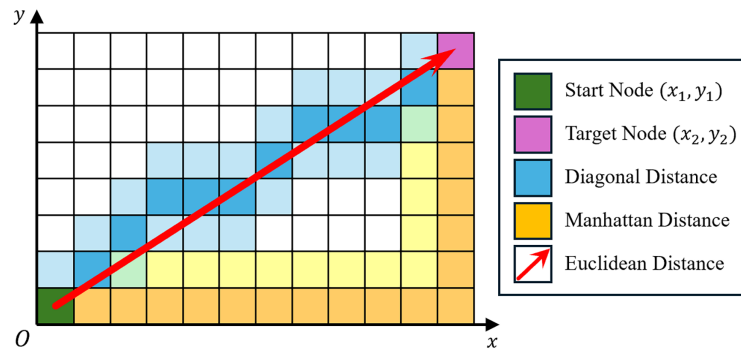


Figure 3: Schematic diagram of various distances between two points. For two points; (x_1, y_1) and (x_2, y_2) , the diagonal distance (4), manhattan distance (5), and Euclidean distance (6) are defined as follows.

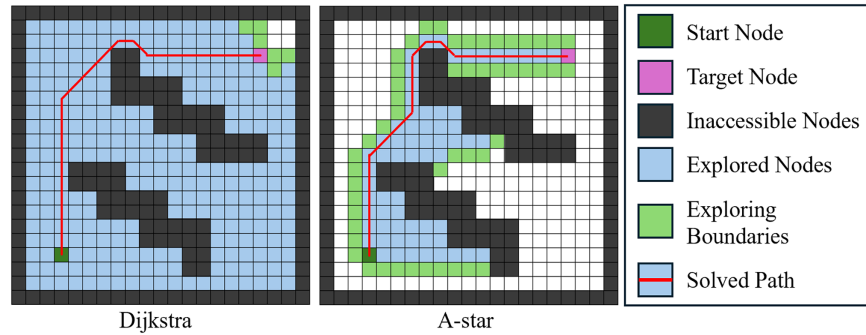


Figure 4: Comparison of the efficiency between Dijkstra's algorithm and A* algorithm.

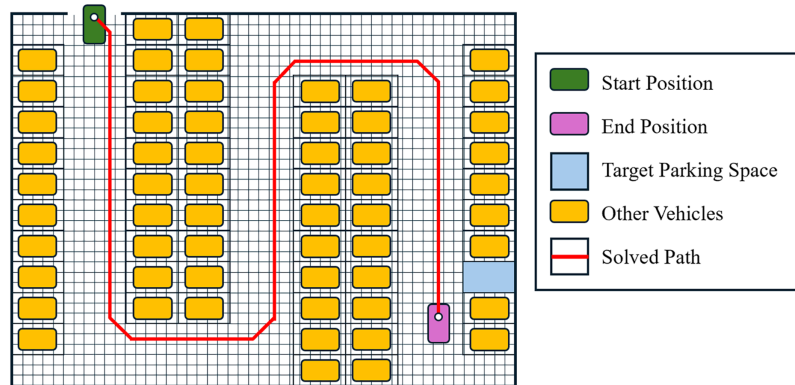


Figure 5: Application of A* algorithm for planning paths to target parking spaces.

2.2 Hybrid A* Algorithm

The Hybrid A* algorithm [35] is a modified variant of the A* algorithm designed for motion planning problems. On the basis of the basic A* search method, it considers vehicle kinodynamic constraints and improves the comprehensive cost function $f(n)$. In this algorithm, a node is defined as a three-dimensional coordinate (x, y, θ) , where θ is the vehicle's heading angle. The path solved by the algorithm consists of

two parts: the path explored through node expansion, and the path connected by Reeds-Shepp (RS) curves. The Hybrid A* algorithm expands to child nodes that satisfy kinodynamic constraints through six types of motion paths defined by different combinations of forward velocity v and front wheel steering angle φ , as shown in Fig. 7a. In addition, the algorithm attempts to directly connect to the target point using an RS curve every time several nodes are expanded. If there is no collision, the path solving is completed, and the connection frequency is related to the distance between the current node and the target point. In the motion planning for automatic parking, due to the small exploration space, the algorithm attempts the connection operation every time a node is expanded during operation.

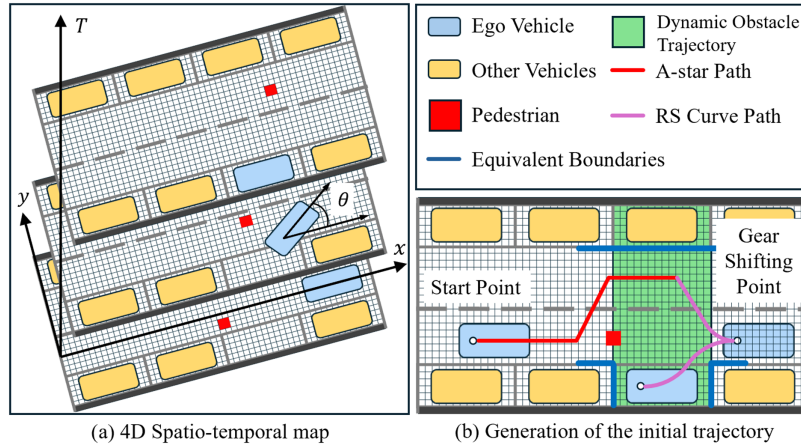


Figure 6: Trajectory planning of the spatio-temporal heuristic A* algorithm.

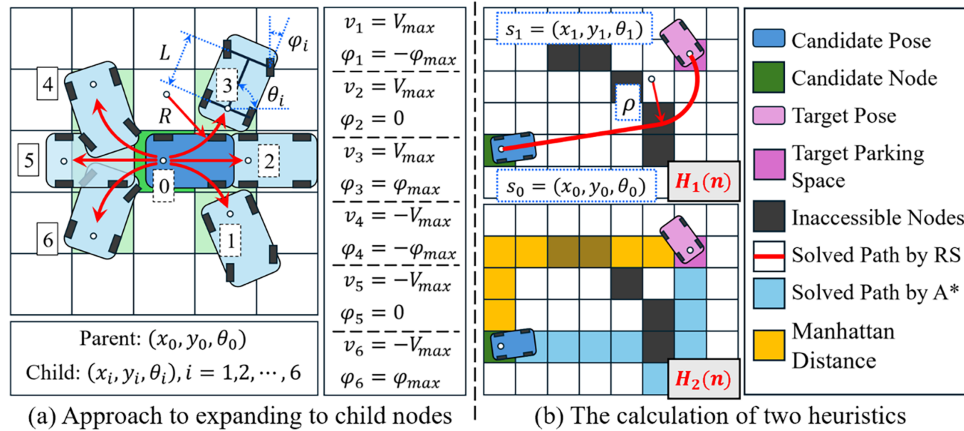


Figure 7: Schematic diagrams related to Hybrid A*.

Regarding the modification of the $f(n)$ function, the content of the actual cost increment function $\Delta g(n)$ of Hybrid A* is first expanded. The new function comprehensively considers the path length cost C_{dist} , gear-shifting cost C_{gear} , and steering angle cost C_{steer} , with corresponding weights assigned to each, as shown in Eq. (7). Among them, the calculation of C_{dist} is related to the front wheel steering angle φ_i of the child node, as shown in Eq. (8); C_{gear} is used to penalize reverse driving operations—if the velocities v of the child node and the parent node are opposite to each other, a penalty coefficient k_{gear} is output; C_{steer} is used

to penalize frequent steering of the front wheels, which is calculated by computing the difference between the front wheel steering angles φ of the parent node and the child node, as shown in Eq. (9).

$$g(n) = \alpha \cdot C_{dist} + \beta \cdot C_{gear} + \gamma \cdot C_{steer} \quad (7)$$

$$C_{dist} = \begin{cases} \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}, & \varphi_i = 0 \\ R \cdot |\theta_i - \theta_0|, & \text{where } R = \frac{L}{|\tan(\varphi_i)|}, \varphi_i \neq 0 \end{cases} \quad (8)$$

$$C_{steer} = |\varphi_i - \varphi_0| \cdot k_{steer} \quad (9)$$

The Hybrid A* heuristic combines two complementary estimates. The constrained heuristic $H_1(n)$, illustrated in the upper panel of Fig. 7b, ignores obstacles but respects vehicle kinematic constraints; it is computed from the length of a Reeds-Shepp curve connecting the current child node to the target, as defined in Eq. (10). The obstacle-aware unconstrained heuristic $H_2(n)$, illustrated in the lower panel of Fig. 7b, is precomputed once by running a reverse Dijkstra search from the target over a two-dimensional occupancy grid while ignoring vehicle orientation and kinematic constraints. During Hybrid A* expansion, the value of $H_2(n)$ for each child node is obtained by a lookup at its grid cell rather than by rerunning Dijkstra from that child node. The final heuristic h is the maximum of $H_1(n)$ and $H_2(n)$, as shown in Eq. (12).

$H_1(n) = \text{Length}(RS(s_0, s_1, \rho))$, In Which :

$$RS \in \{LSL, LSR, RSL, RSR, LRL, LRR\} \quad (10)$$

$$H_2(n) = \max \{d_{Dijkstra}, d_{Manhattan}\} \quad (11)$$

$$h(n) = \max \{H_1(n), H_2(n)\} \quad (12)$$

In motion planning for automatic parking, Hybrid A* has been widely applied. Through its unique path exploration method, the algorithm can generate a path that satisfies vehicle kinodynamic constraints; and via the comprehensive cost function, it is capable of producing near-globally optimal paths in various parking scenarios, as illustrated in Fig. 8. However, the Hybrid A* algorithm still has unavoidable limitations: Firstly, the nodes searched by Hybrid A* must meet vehicle kinodynamic requirements, resulting in a motion planning success rate of less than 100%; Secondly, due to the expansion of the node state dimension to three dimensions, the computational load of Hybrid A* in complex environments increases significantly, which greatly affects search efficiency; Finally, Hybrid A* still struggles to handle dynamic obstacles. To compensate for the three disadvantages mentioned above, recent research can be mainly divided into two directions: improving the comprehensive cost function $f(n)$ and reducing invalid searching nodes. The flowchart of A* and Hybrid A* algorithms for AVP is shown in Fig. 9.

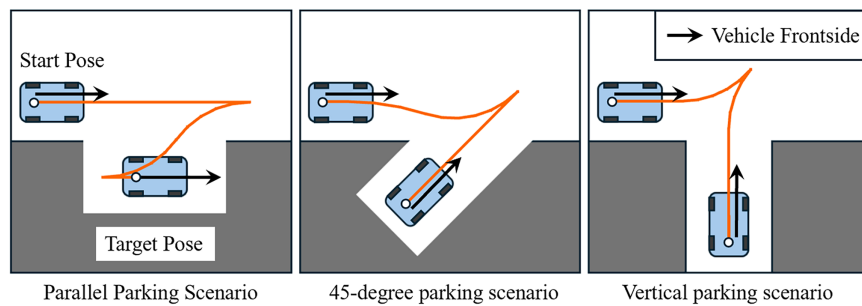


Figure 8: Path planning diagrams of Hybrid A* in multiple parking scenarios.

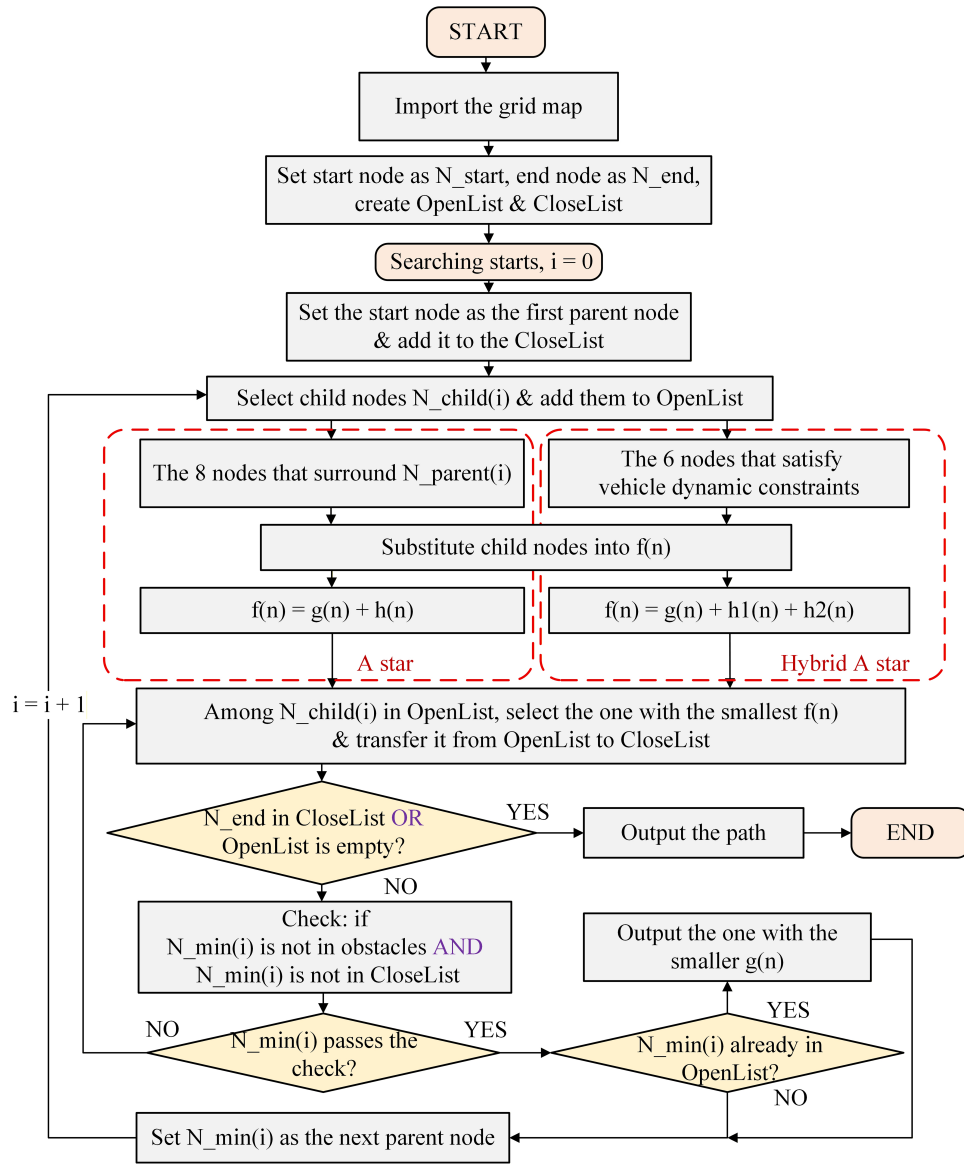


Figure 9: Flowchart of A* and Hybrid A* algorithm for AVP.

2.2.1 Improvements on the Comprehensive Cost Function

The actual cost function of the Hybrid A* algorithm determines the search depth and affects the feasibility and safety of the solved path; while the heuristic function determines the search breadth, and influences the solution efficiency and avoiding the path falling into a local optimum. In the research focused on adapting to multiple parking scenarios, Zeng et al. [36] proposed an improved Hybrid A* algorithm. In this approach, driving direction and steering angle were incorporated into the actual cost. Besides, Reeds-Shepp curve length and eight-neighborhood Euclidean distance were considered in the estimated cost. The results proved that the timeliness and universality of the algorithm are improved. Xiong et al. [37] replaced traditional circular arcs with Clothoid curves to ensure curvature continuity. At the same time, a penalty term was added for obstacle distance to the estimated cost. Multifaceted optimization optimized the trajectory smoothness, obstacle avoidance safety, and universality of the Hybrid A* algorithm. Huang et al. [38]

proposed a Multi-Heuristic Hybrid A* (MHHA*) algorithm by designing a multi-heuristic mechanism, which introduces a primary admissible heuristic for quality bounds and supplements multiple inadmissible ones for strong guidance. By dynamically polling independent queues, it quickly bypasses obstacles and local minima, significantly cutting search time while keeping path sub-optimality bounded.

To make the Hybrid A* algorithm competent for more application scenarios, the node dimension is further expanded in some studies. The original comprehensive cost function can only handle three-dimensional information. Therefore, a novel function that is capable of processing higher-dimensional information is indispensable to the improved algorithm. Addressing the dynamic obstacle avoidance problem, Nawaz et al. [39] proposed a time-indexed Hybrid A* algorithm, where a time dimension is added to each node. The motion prediction of dynamic obstacles is integrated into the cost function, and the function is defined as the sum of obstacle avoidance costs corresponding to the positions of dynamic obstacles at different times. To realize the automatic parking of semi-trailer trains, Tian et al. [40] added the heading angle θ_2 dimension of the semi-trailer to the node. Therefore, the articulation angle cost and lateral movement cost are incorporated into the cost function. Meanwhile, the heuristic function is modified to set different target points for the tractor and the semi-trailer, and ensures motion coordination by establishing a derivation relationship between the attitudes of the two vehicle bodies.

2.2.2 Reduction of Invalid Searching Nodes

The path search method of the traditional Hybrid A* algorithm is effective in a uniform space. However, the goal of automatic parking is to move the vehicle from an open space (the road) to a narrow space (the parking place). Consequently, a large number of invalid nodes are generated during the path search, which greatly affects the search efficiency of the algorithm. To address this issue, Pang et al. [41] proposed an RS-HA* algorithm, in which the start and end nodes were swapped to make the search start from the parking point with dense obstacles. Besides, pruning operations were added to reduce invalid node expansion, significantly shortening the search time.

In addition, using local paths or local target points to guide the global path is also an effective method to reduce invalid nodes in non-uniform spaces [42–46]. Li et al. [42] proposed a scenario-based path planning method. First, appropriate intermediate points were selected according to different parking scenarios; then the Hybrid A* algorithm was used to generate local paths from the intermediate points to the initial state and the end point; Finally, the two paths were spliced to obtain the global path. Sheng et al. [43] focused on parking environments with narrow passages. The A* algorithm was first used to obtain the global path and identify narrow passage segments; then the Hybrid A* algorithm was used to plan sub-paths in these segments; Finally, another Hybrid A* algorithm was used to connect the local paths to generate the trajectory, as shown as in Fig. 10. Su et al. [44] first used the Voronoi algorithm to segment the map. Then the A* algorithm was used to search for the global guide line; After that, a target tree was generated from the parking space to transfer the target pose in the narrow space to the open area; Finally, the Hybrid A* algorithm was used to search for paths that meet vehicle kinodynamics, effectively reducing search time consumption and ensuring safety.

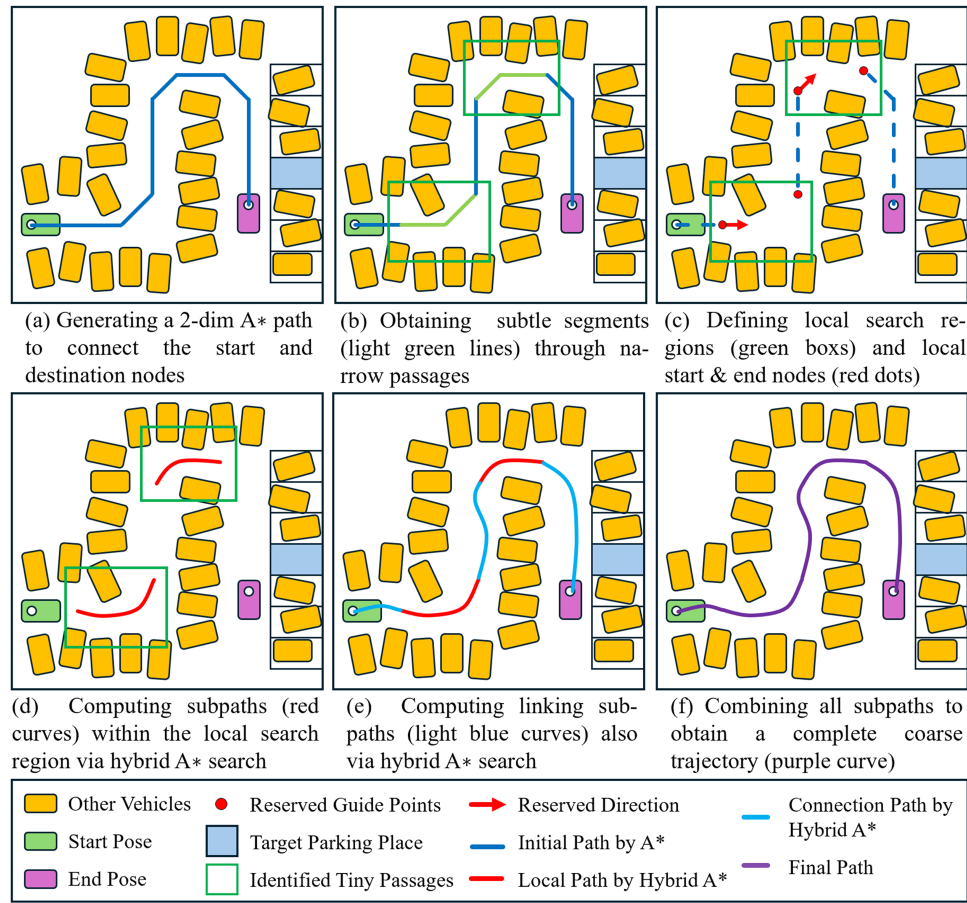


Figure 10: A multi-stage Hybrid A* algorithm for parking scenarios with tiny passages.

3 Sampling-Based Trajectory Planning Methods

Sampling-based path planning methods are important for addressing the high-dimensional, non-convex, and highly constrained planning problems encountered in automated parking [47]. Narrow passages, dense obstacles, and vehicle kinematic constraints make these scenarios challenging. These methods use random or guided samples in the configuration space and avoid explicitly constructing the full configuration-space obstacle region; nevertheless, they still require collision checking against a geometric or occupancy-grid representation of the environment. Under the assumptions of the corresponding planner, probabilistic completeness means that the probability of finding an existing feasible path approaches one as the number of samples tends to infinity. This asymptotic property does not guarantee efficient exploration, and narrow passages remain a well-known source of sampling inefficiency [48].

The development of these methods closely aligns with the specific demands of parking. To address common parking challenges such as narrow passage navigation and precise pose convergence, representative algorithms including the Probabilistic Roadmap (PRM) and the Rapidly-exploring Random Tree (RRT) have played important roles. PRM supports multi-query planning through the construction of an offline roadmap, suitable for repeated planning tasks in structured parking lots. RRT and its variants focus more on single-query real-time path generation, able to integrate vehicle kinematics with biased sampling to directly synthesize feasible trajectories that comply with steering geometry [49]. Although alternative planning methods based on search or optimization also exist, sampling-based planning—with its flexibility to explore

high-dimensional constrained spaces—has become an effective and widely adopted strategy to handle complex geometric and motion constraints in automated parking systems. It is often integrated with other planning modules to form a complete solution.

3.1 Classical Sampling Algorithms

Classical sampling algorithms, namely the Probabilistic Roadmap (PRM) and the Rapidly-exploring Random Tree (RRT), employ distinct strategies highly relevant to parking.

PRM, shown in Fig. 11, operates in two phases: a learning phase, where it randomly generates numerous “milestone” points in the free area of the configuration space and connects neighboring points to form a graph structure called a “roadmap”, and a query phase, where a graph search algorithm (e.g., Dijkstra or A*) quickly finds a path on this precomputed roadmap. This makes PRM suitable for multi-query scenarios, such as planning multiple paths within the same parking lot layout.

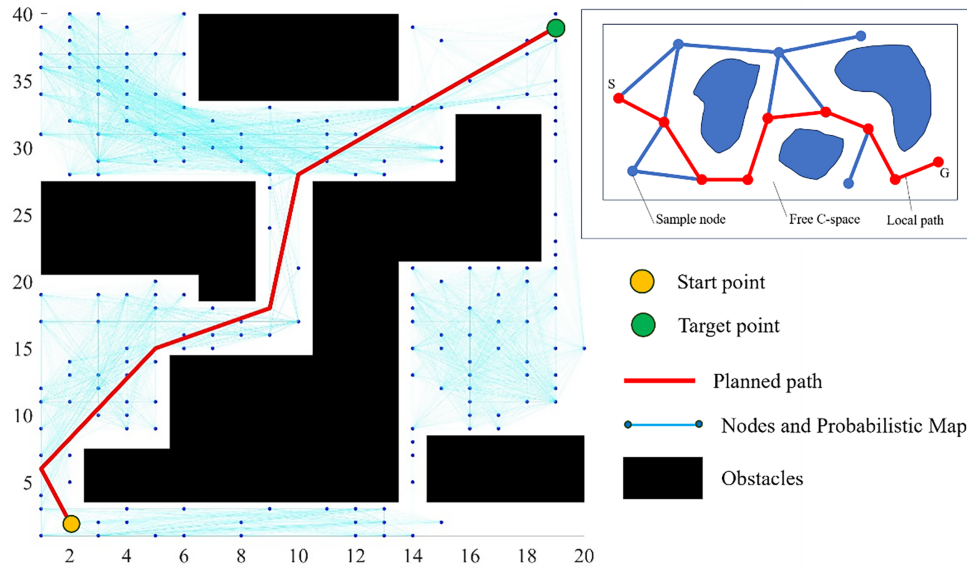


Figure 11: An example of the probabilistic roadmap.

Phase 1: Roadmap Construction (Learning Phase) $G = (V, E)$

- (1) Node Sampling: Randomly sample N configurations from the free space.
- (2) Neighborhood Definition: For a node q_i , candidate neighbors can be selected by a fixed-radius rule, a k -nearest-neighbor rule, or an adaptive neighborhood criterion. The distance metric $dist(q_a, q_b)$ is often the Euclidean norm in configuration space.
- (3) Local Connection & Edge Creation: For each neighbor q_j in the neighborhood, a local planner (e.g., straight-line interpolation) generates a path $\sigma: [0, 1] \rightarrow C$ with $\sigma(0) = q_i$, $\sigma(1) = q_j$. An edge e_{ij} is added to the graph if this path is collision-free. The edge weight is typically the distance: $w_{ij} = dist(q_i, q_j)$.

$$V = \{q_i | q_i \in C_{free}, i = 1, \dots, N\}, q_i \sim Uniform(C_{free}) \quad (13)$$

$$N_K(q_i) = KNN(q_i, V, K), N_R(q_i) = \{q \in V \setminus \{q_i\} | dist(q_i, q) < r\} \quad (14)$$

$$\forall s \in [0, 1], \sigma(s) \in C_{free}, E = \{e_{ij} | q_j \in N(q_i) \text{ and } \sigma \subset C_{free}\} \quad (15)$$

Phase 2: Path Query

- (1) Attempt to connect these query nodes to the existing roadmap G using the same local planner. Find sets of roadmap nodes reachable from each. Temporarily add q_s and q_g to G with edges to V_s and V_g .
- (2) Graph Search: Perform a shortest-path search (e.g., Dijkstra's algorithm) on the augmented graph. If a path P^* exists, it is returned as a sequence of nodes/edges; otherwise, the query fails.

$$V_s, g = \{v \in V \mid \text{LocalPlanner}(q_s, g, v) \text{ is collision-free}\} \quad (16)$$

$$P^* = \arg \min_{P \in \text{Paths}(q_s, q_g)} \sum_{e \in P} w_{ij} \quad (17)$$

In contrast, RRT shown in Fig. 12 is a single-query algorithm ideal for one-time parking maneuvers. It grows a tree iteratively from the start pose. In each iteration, it samples a random point in the configuration space, finds the nearest node in the existing tree, and then extends from this node towards the sample by a fixed step. If this extension is collision-free, the new node is added. This biased, greedy exploration allows the tree to rapidly expand towards unexplored regions until it reaches the goal region.

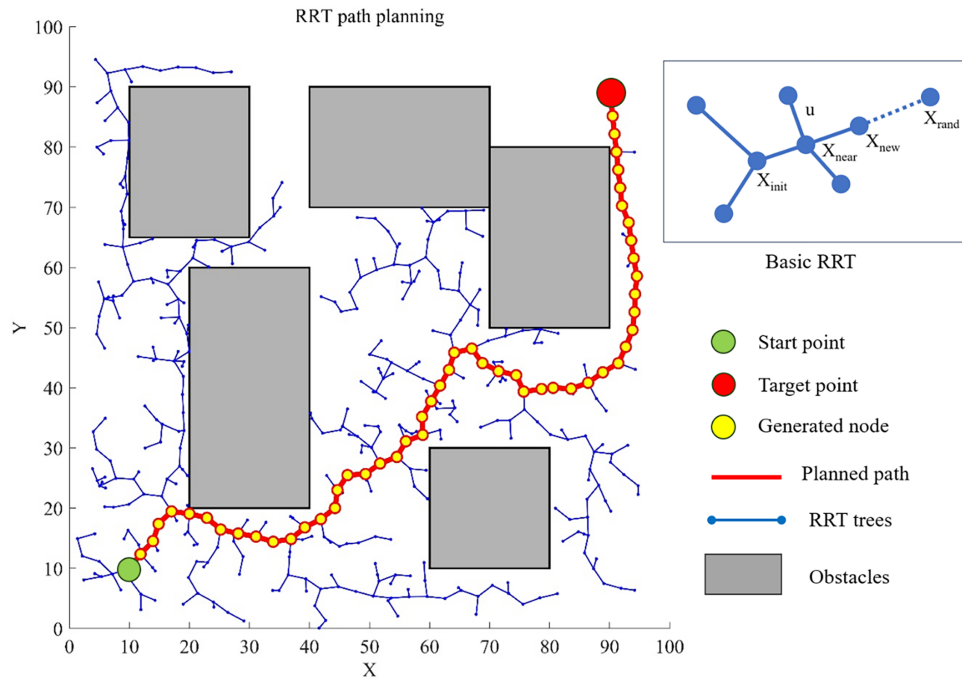


Figure 12: An example of RRT path planning.

The RRT algorithm incrementally builds a tree $T = (V, E)$ rooted at the start configuration q_{start} to explore the configuration space C , aiming to reach a goal region.

The core of the RRT algorithm is a single iteration that expands the tree. For iteration k :

$$q_{\text{rand}} \sim \text{Uniform}(C_{\text{free}}) \quad (18)$$

$$q_{\text{near}} = \arg \min_{q \in V} \text{dist}(q, q_{\text{rand}}) \quad (19)$$

$$q_{\text{new}} = q_{\text{near}} + \min\left(1, \frac{\delta}{\|q_{\text{rand}} - q_{\text{near}}\|}\right)(q_{\text{rand}} - q_{\text{near}}) \quad (20)$$

Validate the path segment $\sigma(s)$ from q_{near} to q_{new} for collisions.

$$\forall s \in [0, 1], \sigma(s) \in C_{free} \quad (21)$$

$$\sigma(s) = q_{near} + s(q_{new} - q_{near}) \quad (22)$$

If the path is collision-free, add q_{new} to the vertex set V and add the edge (q_{near}, q_{new}) to the edge set E .

$$V \leftarrow V \cup \{q_{new}\}, E \leftarrow E \cup \{(q_{near}, q_{new})\} \quad (23)$$

Sampling-based methods offer important advantages for automated parking. They explore the three-dimensional vehicle configuration space (X, Y, θ) without exhaustively discretizing it into a high-resolution grid, thereby reducing the cost of constructing and searching a dense graph. They also adapt readily to perpendicular, parallel, and angled parking layouts and can discover collision-free routes through irregular obstacle arrangements and narrow passages. Basic geometric sampling, however, does not automatically satisfy Ackermann steering or other nonholonomic constraints. Kinematically feasible trajectories are obtained when the planner incorporates motion primitives, kinodynamic state propagation, or a steering function that respects the vehicle model. Under the assumptions of the corresponding algorithm, sampling-based planners are probabilistically complete: the probability of finding an existing feasible path approaches one as the number of samples tends to infinity. Their modular structure also facilitates implementation and extension for parking applications.

Despite these strengths, applying classical sampling-based algorithms directly to automated parking exposes several critical challenges, primarily revolving around the trade-off between planning efficiency and path quality [50]. The highly random search strategy of RRT often wastes samples in open areas while struggling in the tight confines of a parking slot entrance, where valid configurations are scarce. This can lead to unacceptably long search times, failing to meet the real-time requirements (typically tens to hundreds of milliseconds) of a parking system. Moreover, the resulting paths are often erratic, containing unnecessary turns and zigzags. These paths are not only uncomfortable but also challenging, if not impossible, for the vehicle's path tracking controller to follow due to discontinuous curvature. This issue stems from the algorithms operating in a geometric configuration space, ignoring the nonholonomic constraints (Ackermann steering) of real vehicles. Paths generated by straight-line connections between samples are often kinematically infeasible. The "narrow passage" problem is particularly acute in parking. In very tight spots, the probability of a random sample falling into the feasible channel is extremely low, causing planning failure or severe efficiency drops for both PRM (failing to connect across the passage) and RRT (struggling to grow through it). Lastly, parameters like step size significantly impact performance. A large step size increases collision risk and may overshoot narrow passages, while a small one slows convergence, requiring careful tuning for different parking scenarios [51–54].

3.2 Improvements in PRM-Based Methods for Automated Parking

For multi-query path planning in a fixed parking-lot layout, the advanced PRM scheme shown in Fig. 13 improves connectivity in constrained regions, while the overall PRM workflow used for automated valet parking is summarized in Fig. 14. To address the narrow passages and vehicle constraints inherent in parking environments, PRM improvements mainly focus on increasing roadmap connectivity and construction efficiency. (1) Guided sampling strategies: uniform random sampling often produces too few valid nodes in tight parking slots or narrow passages. Bridge-test sampling identifies transition zones through collision-free and collision configurations and increases the probability of sampling critical bottlenecks. (2) Adaptive and multi-resolution roadmaps: the structured features of parking environments, including parking-bay boundaries and lane directions, are used to vary the sampling density or construct hierarchical roadmaps.

Denser samples are placed near parking-space entrances and gaps between vehicles, whereas sparse global sampling limits computational cost while retaining roadmap coverage.

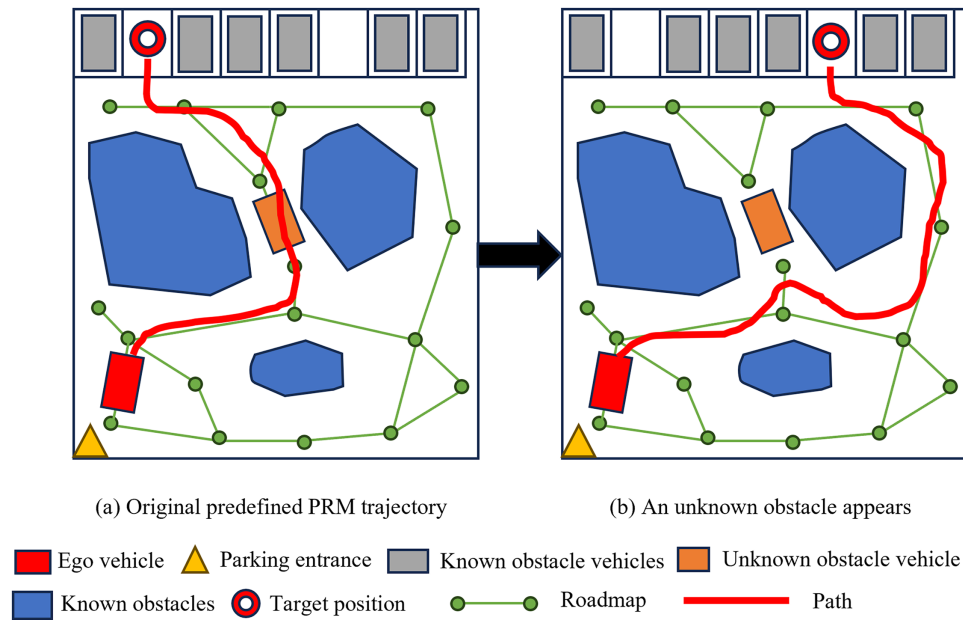


Figure 13: An example of an advanced PRM method.

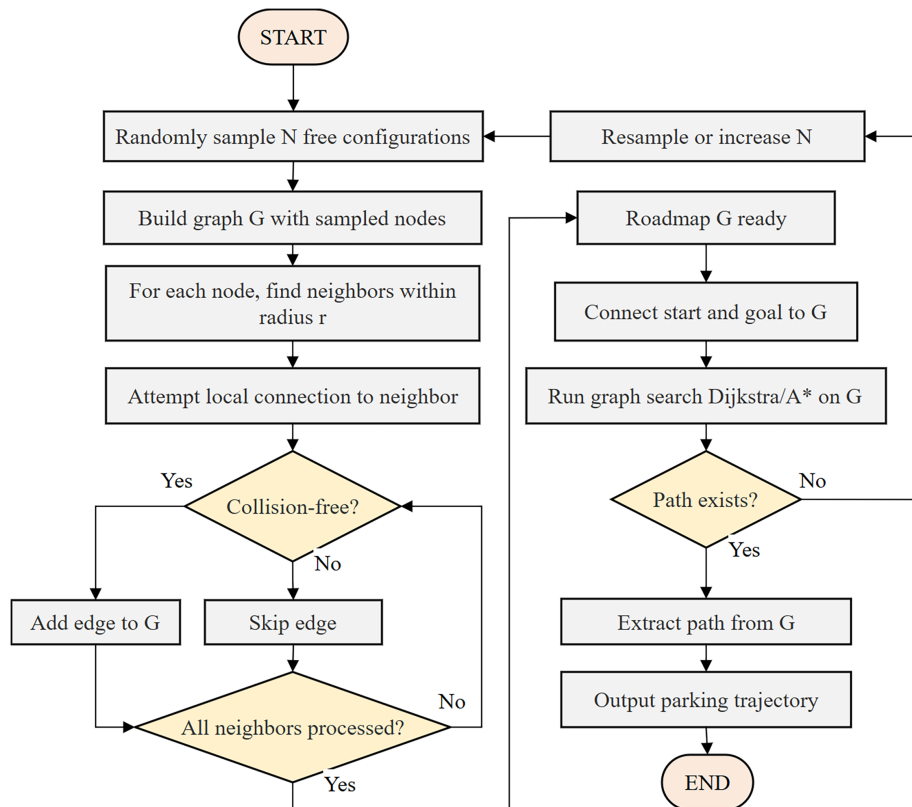


Figure 14: Flowchart of probabilistic roadmap (PRM) algorithm for AVP.

Relevant studies illustrate the development of PRM-based parking planners. Fuji et al. [55] incorporated vehicle orientation and nonholonomic constraints into a multi-resolution state roadmap for automated parking; simulations and real-vehicle tests demonstrated efficient planning of complex maneuvers. Tazaki et al. [56] subsequently developed a hierarchical multi-resolution state roadmap based on parking-lot guidelines, which reduces online computation and maintains kinematic feasibility and planning quality. Together, these studies show that environmental structure and vehicle kinematics can be integrated into PRM without losing its multi-query advantage. Alpkiray et al. [57] combined PRM with an Artificial Bee Colony algorithm to refine an initial route under safety, smoothness, energy, and time constraints. Han et al. [58] proposed a dual-layer PRM-RL framework in which PRM-generated subgoals guide Proximal Policy Optimization exploration and the learned policy refines kinematically feasible paths with smoothness constraints. Padmaraja and Ranjith [59] applied PRM to coordinated multi-AGV warehouse navigation by dynamic scheduling and collision avoidance; although this setting differs from automated parking, it illustrates the scalability of roadmap-based coordination.

3.3 Improvements in RRT-Based Methods for Automated Parking

In contrast to PRM, the Rapidly-exploring Random Tree (RRT) shown in Fig. 15 and its variants are primarily designed for single-query, real-time parking path planning. To enhance their performance in narrow and potentially dynamic parking environments, improvements have progressed along two main avenues: first, adaptations for system integration and specific application scenarios; second, core algorithmic optimizations for search efficiency and path quality. The majority of innovations target the RRT family, optimizing it for the single, real-time planning requests typical of a one-time parking maneuver. Key enhancements include guided sampling strategies such as bidirectional growth for faster convergence (e.g., RRT-Connect), asymptotically optimal variants (e.g., RRT*) that continuously refine an initial feasible solution through rewiring and cost reduction, kinematic integration through motion primitives to ensure feasibility, and hybrid frameworks that combine sampling with optimization to balance efficiency with trajectory quality. In particular, RRT* progressively converges toward the global optimum as the number of samples tends to infinity, yielding smoother and more energy-efficient trajectories.

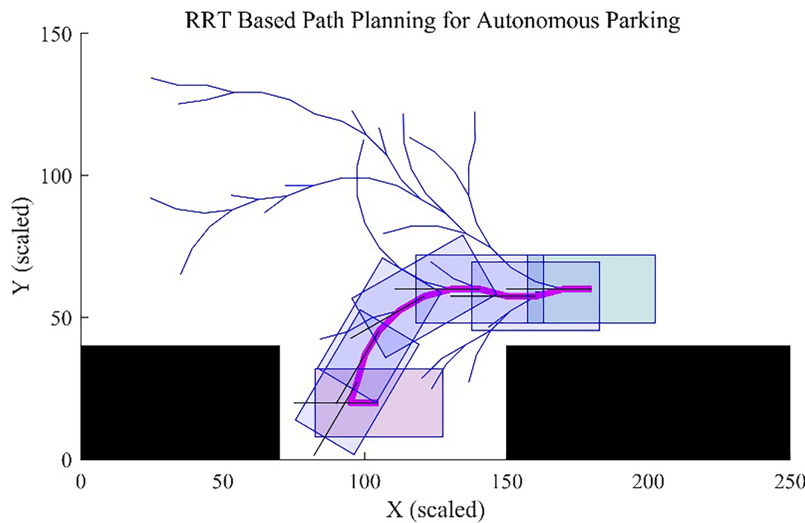


Figure 15: RRT for autonomous parking.

Parking-oriented RRT improvements reconcile random exploration with nonholonomic vehicle kinematics. A basic RRT connects samples by straight segments and can generate jagged paths that violate steering and minimum-turning-radius limits. Motion-primitive expansion instead propagates the vehicle state along feasible arcs and enforces steering bounds during tree growth. Arc-based primitives alone may still create curvature discontinuities at junctions, resulting in abrupt steering commands. Continuous-curvature steering functions, such as the clothoid-based target-tree construction in [60], smooth these transitions and produce progressively varying steering inputs. Fig. 16 summarizes the flow of representative multi-variant RRT architectures.

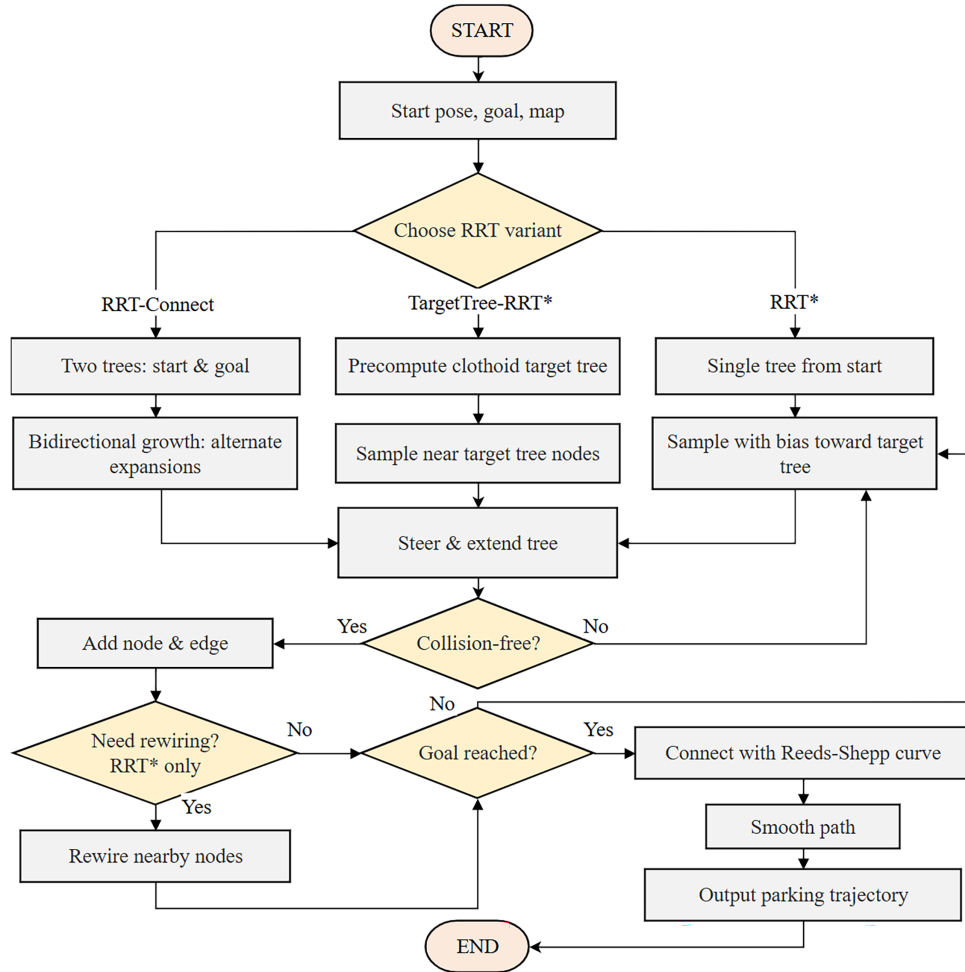


Figure 16: Flowchart of multi-variant RRT architectures for AVP.

3.3.1 Enhancements through RRT*-Based Methods

RRT* methods refine feasible paths through rewiring and asymptotically approach the optimum under their standard assumptions. Parking-oriented variants improve convergence and path executability through specialized target representations, sampling strategies, and local connections, as summarized in Fig. 17. Kim et al. [60] proposed TargetTree-RRT*, which replaces a single goal state with a precomputed set of clothoid-based backward paths, and the target tree improves goal connection in narrow parking spaces and yields continuous-curvature trajectories. Yang et al. [61] combined Sobol-sequence RRT* with numerical optimal control for four-wheel-steering vehicles; the low-discrepancy samples improve global coverage, while

numerical optimization refines the trajectory and handles dynamic-obstacle avoidance. Dong et al. [62] used reverse tree growth, Reeds–Shepp connections, and knowledge-biased sampling to accelerate automated-parking searches and improve kinematic feasibility. Reliability-oriented extensions developed for off-road planning provide complementary robustness mechanisms: R2-RRT* incorporates terrain uncertainty and path-reliability constraints [63], whereas ER-RRT* uses Gaussian-process surrogate modeling and active learning to reduce reliability-evaluation cost [64]. These latter methods are not parking-specific, but their uncertainty-aware formulations indicate how sensing and localization uncertainty could be incorporated into future automated-parking planners.

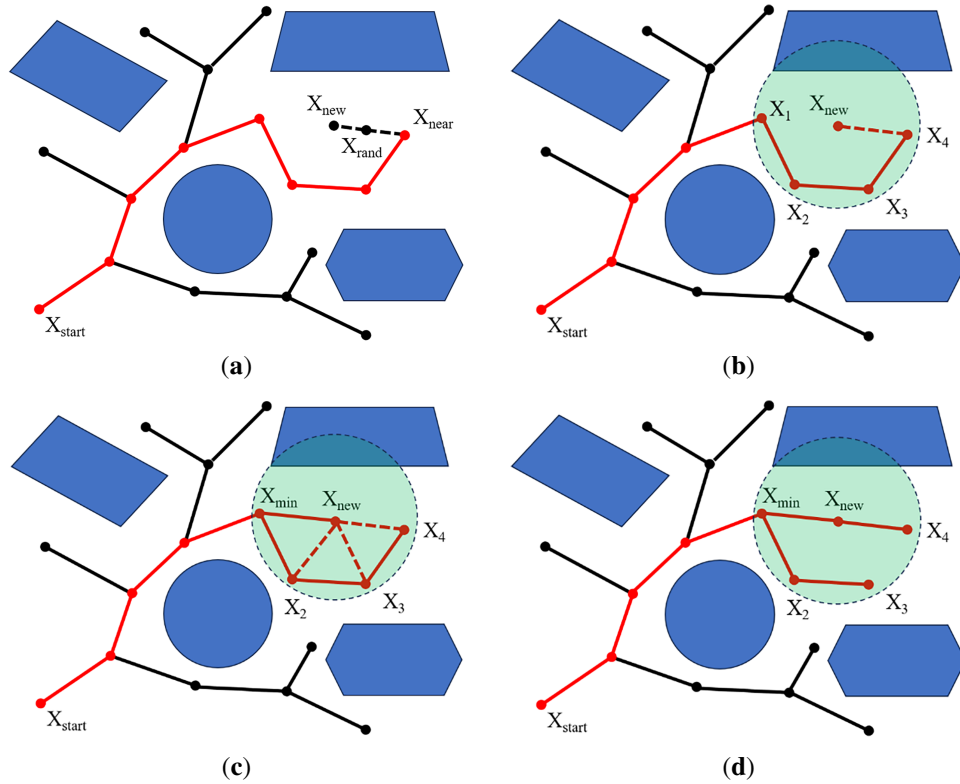


Figure 17: An example of the RRT* improvements—sobel-RRT* algorithm: (a) new nodes generated; (b) neighborhood node search; (c) reselect the parent node; (d) rewiring.

3.3.2 Enhancements through RRT-Connect Methods

The RRT-Connect line of research exploits bidirectional search to drastically reduce the time to find an initial feasible path, as shown in Fig. 18, a critical requirement for real-time parking maneuvers.

$$q_{near}^t = \arg \min_{q \in V_i} dist(q, q_{new}^c) \quad (24)$$

$$q_{connect}^i = q_{near}^t + \frac{i \cdot \delta_{step}}{\|q_{new}^c - q_{near}^t\|} (q_{new}^c - q_{near}^t), i = 1, 2, \dots \quad (25)$$

Improvements in this area often integrate RRT planners with higher-level systems or adapt them to complex vehicle models. For instance, Schörner et al. [65] developed an automated valet parking system that implements RRT*-Connect for parking-maneuver planning within an integrated parking-management

framework. Lattarulo et al. [66] developed an RRT-based trajectory planner for semi-trailer trucks performing constrained logistics maneuvers. Solmaz et al. [67] evaluated RRT-based algorithms for automated valet parking applications in simulation and real-world tests. Manav and Lazoglu [68] presented a cascade path-planning approach for truck-trailer parking that combines Closed-Loop RRT, which expands the tree through closed-loop prediction to respect vehicle dynamics, with an Iterative Analytical Method. Ma et al. [69] proposed Bi-Risk-RRT, a bidirectional motion-planning algorithm that uses the reverse tree as a heuristic and incorporates risk prediction for kinodynamic planning. Together, these studies emphasize reliable and efficient generation of an initial collision-free trajectory for constrained parking scenarios.

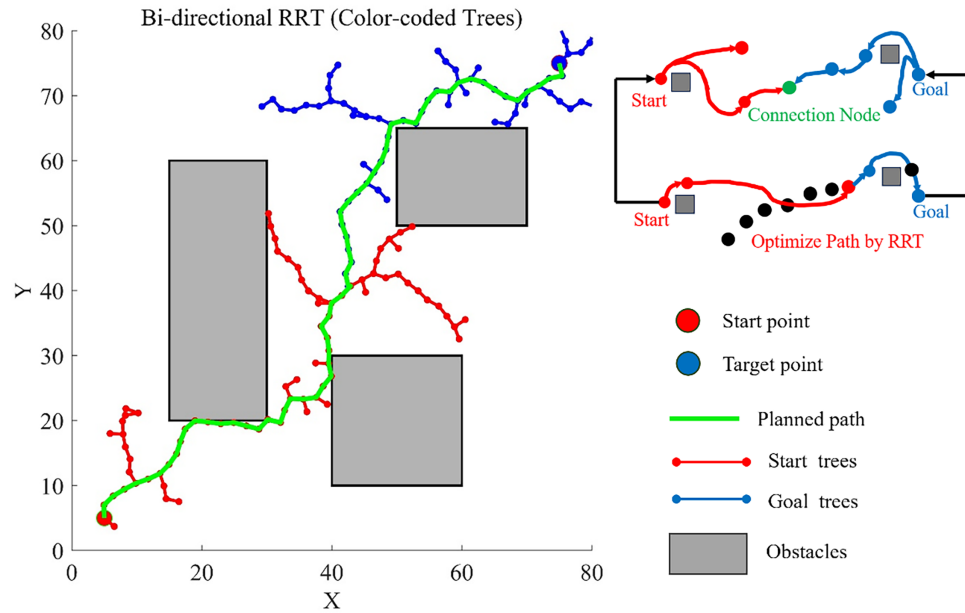


Figure 18: An example of the RRT-connect improvements.

Sampling-based path planning algorithms, starting from the foundational work of PRM and RRT, provide a powerful paradigm for solving the high-dimensional, nonlinearly constrained search problem inherent in automated parking. Their search capability in high-dimensional spaces, adaptability to complex environments, and theoretical probabilistic completeness make them indispensable core technologies in this field. Future developments are likely to focus on deeper integration, such as combining with deep learning to use perceptual data for direct sampling guidance, and with reinforcement learning for adaptive parameter tuning. Furthermore, increased emphasis on engineering and systematization will see these algorithms serving as robust components within large-scale, high-concurrency automated parking systems, collaborating more closely with prediction and control modules [70–73]. Ultimately, the continued evolution of sampling-based algorithms will remain a key enabling technology, driving automated parking functions towards greater intelligence, reliability, and comfort.

4 Artificial Potential Field (APF) Based Planning Methods

Among various path planning algorithms, the Artificial Potential Field (APF) method shown in Figs. 19 and 20 has garnered significant attention since its introduction due to its intuitive concept, high computational efficiency, and ease of implementation. It shows great potential in vehicle path planning, particularly in automated parking scenarios.

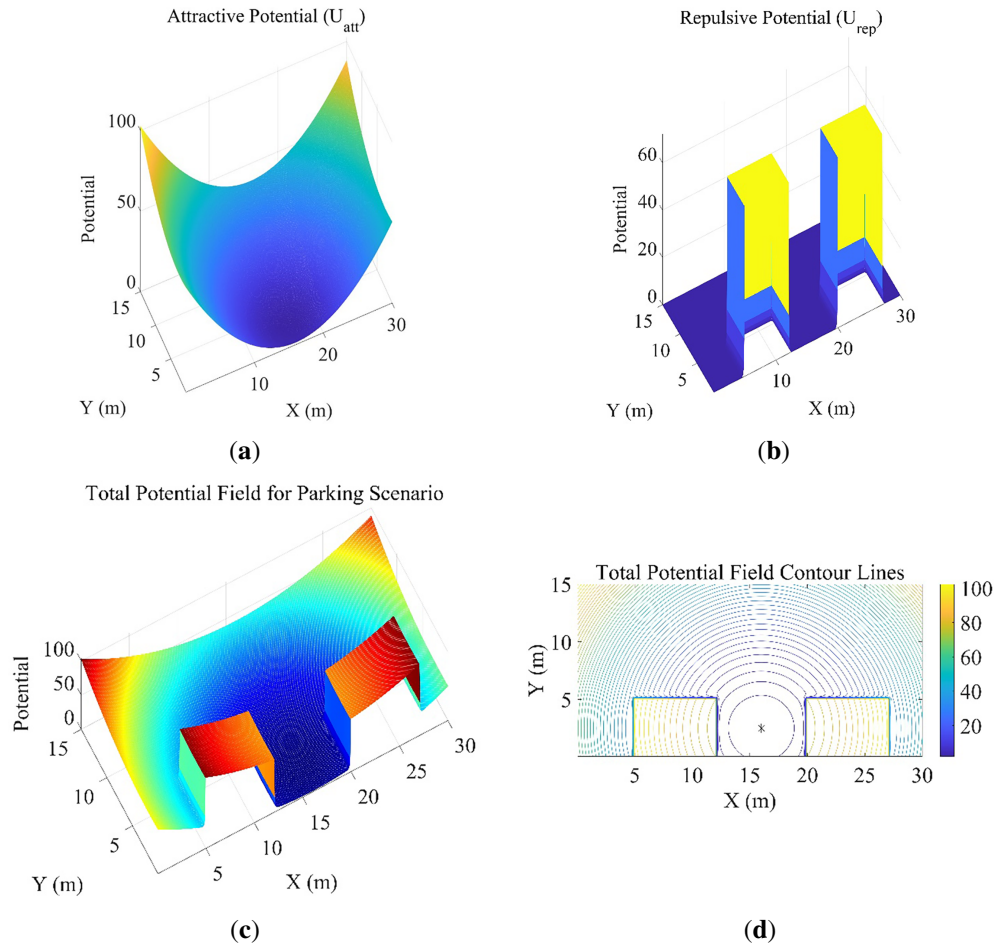


Figure 19: Artificial potential field in parking scenario: (a) attractive potential; (b) repulsive potential; (c) total potential; (d) contour lines.

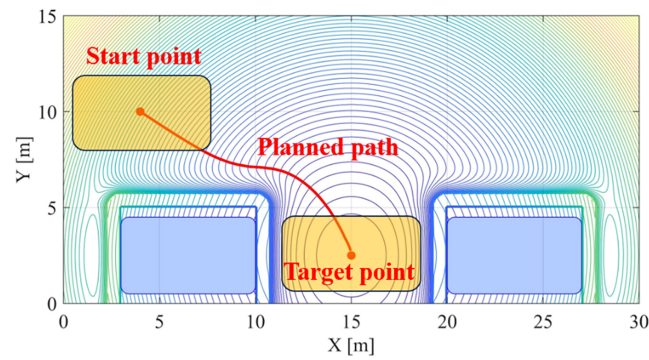


Figure 20: Artificial potential field-based path planning in parking scenario.

The basic idea of APF originates from the concept of potential fields in physics and was first proposed by Khatib in 1986 for robot obstacle avoidance. This method abstracts the movement of a mobile robot in the environment as motion under the influence of a virtual force field: the target point generates an “attractive force” pulling the robot towards it, while obstacles generate “repulsive forces” pushing the robot away from

dangers. The resultant force, calculated from these forces, determines the robot's direction of movement, enabling real-time obstacle avoidance and path planning.

Applying this method to vehicles, especially in automated parking scenarios, is inherently suitable. Parking environments typically have clear targets (parking spots) and well-defined obstacles (surrounding vehicles, pillars, walls, etc.), and demand extremely high real-time planning performance. Traditional graph-search or sampling-based methods (e.g., A*, RRT) can find paths but might suffer from limitations in computational efficiency or path smoothness. In contrast, APF can directly generate smooth, continuous trajectories, aligning well with the input requirements of vehicle controllers and providing an ideal foundation for achieving fluid and natural parking maneuvers. In recent years, with increasing demands for parking experience, researchers have conducted numerous innovations and improvements based on the classical APF, enhancing its ability to adapt to the complex requirements of automated parking [73–76].

4.1 Classical Artificial Potential Field Method

The classical APF model primarily consists of two components: Attractive Field:

Generated by the target point (i.e., the target pose of the parking space, including position and heading angle). Its magnitude is usually proportional to the distance between the vehicle and the target point, directed towards the target. A commonly used attractive potential function is the quadratic form:

$$U_{att}(q) = 0.5k_{att}\rho(q, q_{goal})^2 \quad (26)$$

where k_{att} is the attractive gain coefficient, and ρ is the distance from the vehicle's current position q to the target position q_{goal} . The corresponding attractive force F_{att} is the negative gradient of this potential function.

Repulsive Field: Generated by obstacles in the environment, its effect is typically confined to a region around each obstacle; the vehicle experiences a repulsive force only when it enters this region. For the classical repulsive potential in Eq. (27), the repulsive-force magnitude increases nonlinearly as the vehicle-obstacle distance decreases and becomes zero outside the influence distance. A classical repulsive potential function is given in Eq. (27).

$$U_{obs}(q) = \begin{cases} 0.5k_{rep} \left(\frac{1}{\rho(q, q_{obs})} - \frac{1}{\rho_0} \right)^2, & \rho(q, q_{obs}) < \rho_0 \\ 0, & \rho(q, q_{obs}) \geq \rho_0 \end{cases} \quad (27)$$

where k_{rep} is the repulsive gain coefficient, and ρ_0 is the influence radius of the obstacle. The corresponding repulsive force F_{rep} is the negative gradient of this potential function.

The resultant force acting on the vehicle is the vector sum of all attractive and repulsive forces:

$$F_{total} = F_{att} + \sum F_{rep} \quad (28)$$

The vehicle's direction of motion is determined by this resultant force.

In automated parking, the Artificial Potential Field (APF) method shown in Fig. 21 is attractive because its local gradient calculations support rapid replanning and smooth motion generation. Khatib [77] established the classical potential-field formulation for real-time obstacle avoidance. Chiang et al. [78] combined path guidance with stochastic reachable sets to improve planning in highly dynamic environments. For vehicle applications, Dolgov et al. [79] incorporated potential functions into planning in unknown semi-structured environments, while Martin et al. [80] studied trajectory planning for automated buses in parking areas. Azevedo et al. [81] examined coordination mechanisms for high-density automated parking;

their study emphasized that local motion planning should operate consistently with system-level vehicle coordination. Potential-field concepts have also been adapted to multi-agent formation planning [82] and autonomous grain-cart motion planning [83], which illustrates their broader flexibility while also indicating the need for vehicle- and scenario-specific validation. When heading error is included in the vehicle state, the attractive field can generate an orientation-correction component in addition to positional attraction, enabling simultaneous convergence toward the target position and heading.

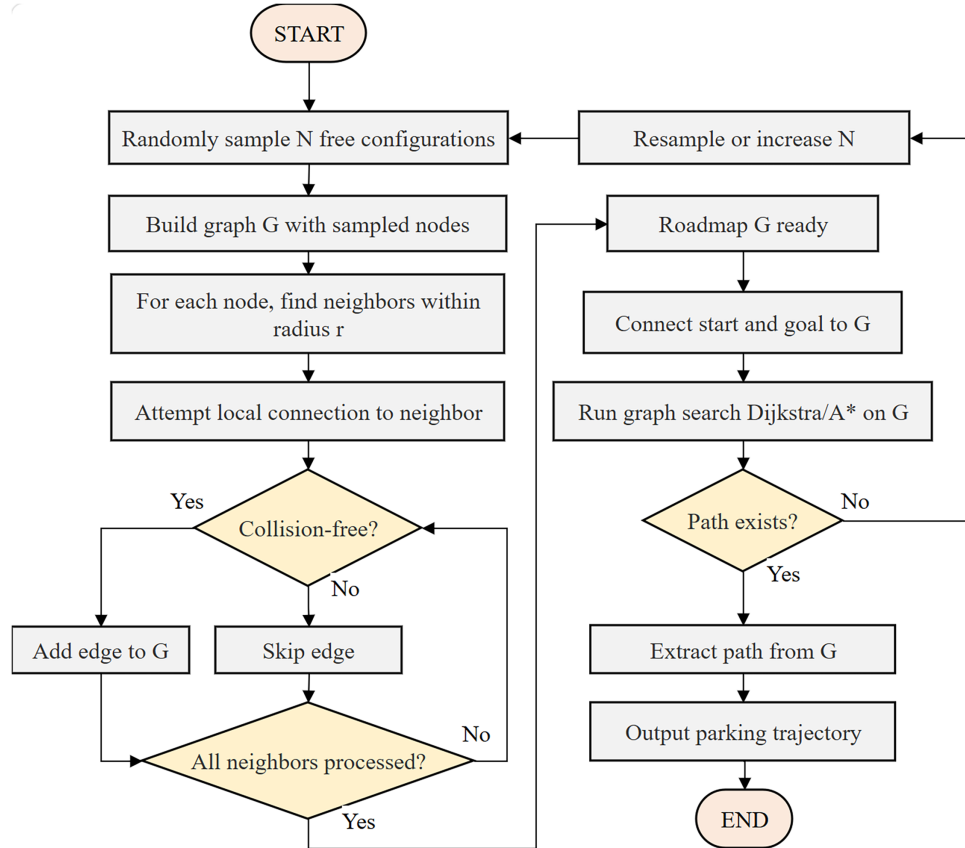


Figure 21: Flowchart of hybrid path planning for AVP based on APF.

4.2 Challenges of Applying APF to Parking Scenarios

Classical APF has several limitations in practical parking environments. The first is the local-minimum problem: attractive and repulsive forces can balance under complex obstacle arrangements, trapping the vehicle before it reaches the parking pose. U-shaped spaces and narrow dead ends are particularly susceptible to this behavior. Hybrid global-local planning and online replanning are commonly used to recover from such traps [84,85]. A second limitation is goal non-reachability. Near the target, the attractive force approaches zero while obstacle-induced repulsion may remain strong; this causes oscillation or prevents precise convergence. Narrow passages create a related potential-barrier effect because repulsion from vehicles on both sides can block entry or induce lateral oscillation. Vehicle kinematics must also be represented explicitly. Classical point-mass APF can generate collision-free paths that violate nonholonomic steering and minimum-turning-radius constraints; vehicle-oriented potential-field controllers address these constraints at the planning-control level [86,87], and Dong et al. [88] converted a discretized APF path into a Reeds–Shepp trajectory for executable perpendicular parking. Finally, APF performance is sensitive to attractive

and repulsive gains. Excessive repulsion can prevent entry into a narrow space, whereas excessive attraction can reduce obstacle clearance. Learning-guided planning [89] and optimal-control formulations [90] provide alternative mechanisms for improving adaptability and enforcing feasibility when fixed potential-field parameters are insufficient.

4.3 Innovative Improvements of APF in Parking Applications

To overcome the aforementioned challenges, researchers have proposed various innovative solutions in recent years, greatly enhancing the performance and robustness of APF in automated parking.

4.3.1 Hybrid Algorithms to Escape Local Minima

Combining APF with other global planning algorithms is an effective strategy to solve the local minima problem. When the vehicle is detected to be stuck in a local minimum, the system can switch to a sampling-based (e.g., RRT) or graph-search (e.g., A*) algorithm to generate a guiding path, helping the vehicle “jump out” of the trap. Afterwards, it switches back to the efficient APF for local planning and obstacle avoidance. This “global-local” combined framework balances global optimality and local real-time performance.

By combining APF with global planners like RRT* to escape local minima, researchers have further developed this hybrid approach through APF-guided sampling, where potential fields directly steer tree growth to efficiently generate high-quality paths for parking-like constrained environments. Tao et al.’s paper [84] improves RRT* by integrating an enhanced APF that guides tree growth with attractive and repulsive forces, reducing randomness and avoiding local minima. Although not exclusively for parking, the method addresses key parking-like challenges—tight spaces and obstacle avoidance—showing that APF-guided sampling can produce faster and higher-quality paths suitable for autonomous parking scenarios. Li et al.’s paper [85] presents a motion-planning framework for autonomous valet parking in dynamic, unstructured environments with moving obstacles. After predicting surrounding vehicles’ motions using an IMM-based intention estimator, the system determines the drivable area by evaluating collision risk through an Artificial Potential Field (APF). The APF is used to assess safety margins around static and moving obstacles, guiding the planner away from high-risk regions. Within this feasible area, an RRT-based local planner generates real-time collision-free paths suitable for parking scenarios. Simulations verify that combining APF-based risk assessment with sampling-based planning enables safe and effective AVP motion planning.

4.3.2 Improvement and Optimization of Potential Field Functions

This is one of the core directions of innovation. Researchers have designed various new potential field functions [86]: (1) Improved Repulsive Field Functions: To address the goal non-reachability problem, a common improvement is to introduce a factor related to the distance to the goal into the repulsive function. This causes the repulsive force to gradually decrease to zero as the vehicle approaches the target, ensuring the goal point becomes the global minimum energy point. (2) Introduction of Guiding Potential Fields/Vector Fields: Instead of using the target point directly as the attraction source, a vector field conforming to the road or lane geometry is constructed. For example, in parking scenarios, a tangential attractive field along the direction of a predefined reference path can be designed, supplemented by a repulsive field perpendicular to the path for obstacle avoidance. This effectively guides the vehicle smoothly into the parking space, preventing it from getting stuck at narrow channel entrances. (3) Asymmetric Potential Fields: Considering the different distances to obstacles on either side of the vehicle during parking, asymmetric repulsive fields can be designed, generating stronger repulsion on the side closer to obstacles. This guides the vehicle to perform steering and adjustments in a safer and more reasonable posture.

Dong et al. [88] applies the Artificial Potential Field (APF) method to generate automated perpendicular-parking trajectories. A discretized APF is used to produce an initial collision-free holonomic path, which is then converted into a kinematically feasible trajectory using Reeds–Shepp curves to satisfy the vehicle’s minimum turning-radius constraints. The final optimized path is translated into waypoints for tracking. Experiments with multiple starting configurations show that the APF-based framework can reliably produce effective and executable perpendicular-parking maneuvers. Kim and Huh [89] proposes a hybrid autonomous parking planner combining neural networks with traditional methods like Hybrid A*. The approach addresses challenges similar to APF-based parking, such as generating feasible, collision-free trajectories in constrained environments. A conditional variational autoencoder (CVAE) guides the search by learning environment-aware feasible paths, improving efficiency and reducing computational cost. This method demonstrates that learning-based guidance can enhance traditional planning techniques for autonomous parking and address the difficulty of parameter tuning, offering a complementary alternative to APF for trajectory generation and path feasibility in complex parking scenarios.

The Artificial Potential Field method, as a historically significant and intuitively conceptual path planning approach, maintains vigorous vitality in the highly practical field of automated parking, leveraging its real-time performance and smooth path generation. Although the classical method has inherent limitations such as local minima, goal non-reachability, and neglect of vehicle dynamics, these have spurred numerous fruitful research innovations.

Future development trends will focus on integration and adaptation. By deeply integrating APF with global planners, advanced control theory (e.g., MPC), and data-driven intelligent methods (e.g., reinforcement learning, deep learning), a next-generation automated parking path planning system can be constructed that possesses a global perspective, local agility, reliable models, and intelligent behavior [91–93]. We have reason to believe that the continuously improved and innovated “intelligent potential field” will continue to play an indispensable key role in the technological journey of making parking safer, more efficient, and more comfortable.

5 Trajectory Planning Method Based on Numerical Optimization

5.1 Establishing Constraints and Constructing Objective Functions

The numerical optimization planning method describes automatic parking path planning as a constrained optimal control problem from the perspective of optimal control theory. Its core elements include system states and control inputs. System states describe the vehicle pose and motion state, usually including the coordinates of the rear axle center, heading angle, speed, and front wheel steering angle φ . Control inputs represent the quantities manipulated by the driver or controller, typically acceleration a and the rate of change of steering angle ω . The two core steps of numerical optimization are establishing constraints and constructing objective functions.

5.1.1 Constraints in Parking Scenarios

In parking scenarios, constraints usually include vehicle kinematic constraints, obstacle avoidance constraints, state variable constraints, control variable constraints, boundary condition constraints, etc.

Vehicle kinematic constraints ensure that the trajectory conforms to the vehicle’s kinematic model shown in Fig. 22, that is, the vehicle must move according to its kinematic laws. It mainly includes the following parameters: the position x and y of the midpoint of the vehicle’s rear axle in the ground coordinate system, the vehicle’s azimuth angle θ , the longitudinal speed v in the vehicle coordinate system, and the front wheel steering angle φ .

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \\ \dot{v}(t) \\ \dot{\phi}(t) \end{bmatrix} = \begin{bmatrix} v(t) \cdot \cos \theta(t) \\ v(t) \cdot \sin \theta(t) \\ v(t) \cdot \tan \varphi(t) / L_w \\ a(t) \\ \omega(t) \end{bmatrix}, t \in [t_0, t_f] \quad (29)$$

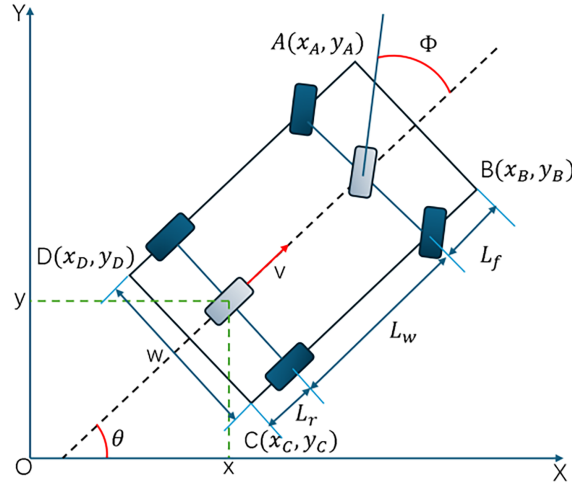


Figure 22: Vehicle kinematic model.

The essence of obstacle avoidance constraints is to convert the physical isolation requirements between the vehicle and surrounding obstacles (such as other vehicles, walls) into a series of mathematical inequalities. Generally, it requires that the minimum distance between the vehicle contour (often approximated as a rectangle or circle) and the obstacle contour is always greater than the preset safety threshold. Common methods include: the polygonal region method shown in Fig. 23, the circular envelope method, the polygon separating axis theorem, etc.

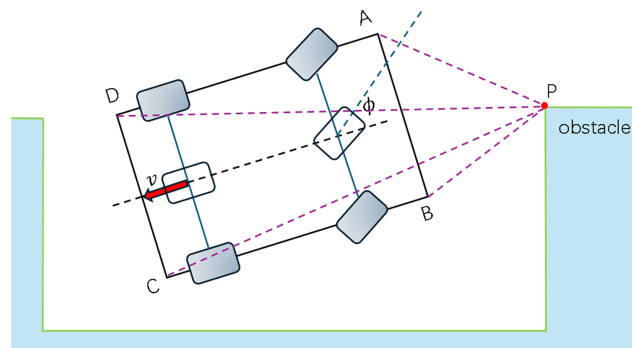


Figure 23: Polygon area method.

For example, the polygonal-region method can be used for parking obstacle avoidance by testing whether an obstacle point lies inside the vehicle polygon. For the four boundary points A, B, C, and D of the vehicle and an obstacle point P, the condition in Eq. (30) detects potential overlap between the obstacle and the vehicle. Collision avoidance requires that no obstacle point lie inside the vehicle polygon during the maneuver.

$$S_{\Delta ABCD} < S_{\Delta PAB} + S_{\Delta PBC} + S_{\Delta PCD} + S_{\Delta PDA} \quad (30)$$

When applying kinematic constraints shown in Fig. 24 to state and control variables, actuator limitations must be considered to prevent sudden changes in vehicle speed and front steering angle, as this may lead to excessive mechanical loads and even make the operation infeasible. Specifically, two state variables and two control inputs are subject to the following limitations [94,95]:

$$\begin{bmatrix} -v_{\max} \\ -a_{\max} \\ -\varphi_{\max} \\ -\omega_{\max} \end{bmatrix} \leq \begin{bmatrix} v(t) \\ a(t) \\ \varphi(t) \\ \omega(t) \end{bmatrix} \leq \begin{bmatrix} v_{\max} \\ a_{\max} \\ \varphi_{\max} \\ \omega_{\max} \end{bmatrix}, t \in [t_0, t_f] \quad (31)$$

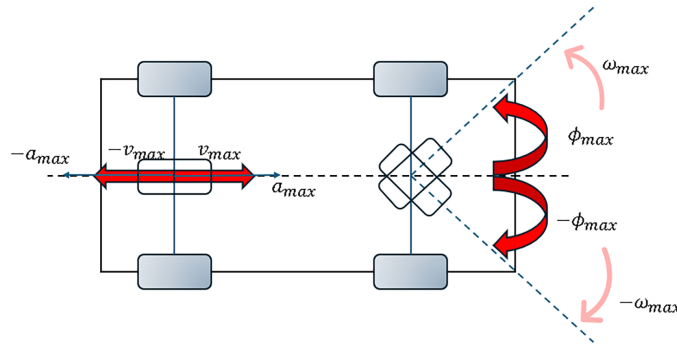


Figure 24: Vehicle actuator constraints.

In parallel parking, boundary condition constraints refer to the requirements that the initial state and target state of the vehicle must meet. The specific target state of the parallel parking task is usually determined by specific requirements, such as facilitating passengers to get off and requiring the vehicle to park at a designated position. Therefore, the initial posture of the vehicle path planning and the posture at the end of the parking planning must comply with certain restrictions. For the boundary constraints of the intermediate stage in segmented parallel parking, it is necessary to ensure that the end state of the previous stage matches the initial state of the next stage. The boundary constraints shown in Fig. 25 must satisfy the following formula.

$$\begin{bmatrix} x(t_0) \\ y(t_0) \\ \theta(t_0) \\ v(t_0) \\ \varphi(t_0) \end{bmatrix} = \begin{bmatrix} x_{start} \\ y_{start} \\ \theta_{start} \\ v_{start} \\ \varphi_{start} \end{bmatrix}, \begin{bmatrix} x(t_f) \\ y(t_f) \\ \theta(t_f) \\ v(t_f) \\ \varphi(t_f) \end{bmatrix} = \begin{bmatrix} x_{final} \\ y_{final} \\ \theta_{final} \\ v_{final} \\ \varphi_{final} \end{bmatrix} \quad (32)$$

5.1.2 Objective Function in Parking Scenarios

Numerical optimization methods with a flowchart shown in Fig. 26 require an optimization objective function to evaluate the quality of the optimization. Common objectives in parking scenarios include: shortest time, minimum control energy, riding comfort, etc. The objective function (J) is a scalar function used to evaluate the quality of the path, usually in the Bolza form, where Φ is the terminal cost, and L is the process cost.

$$J = \phi(x(t_0), t_0, x(t_f), t_f) + \int_{t_0}^{t_f} L(x(t), u(t), t) dt \quad (33)$$

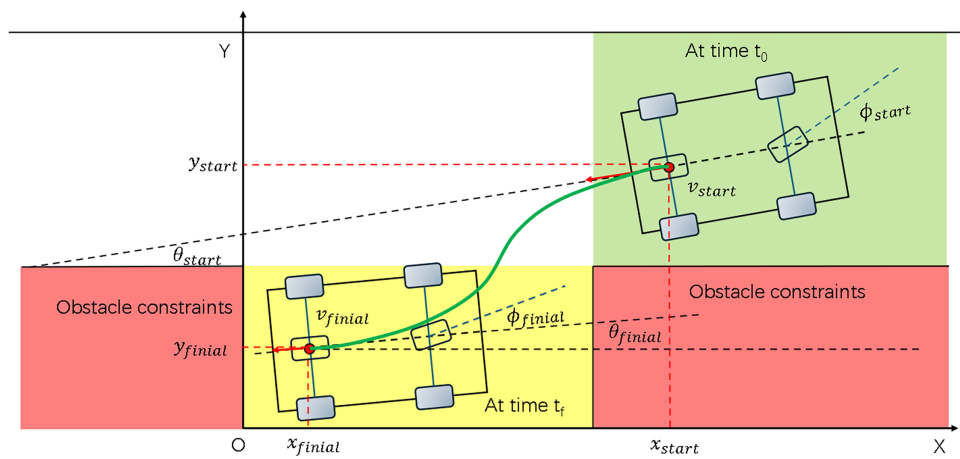


Figure 25: Boundary condition constraints.

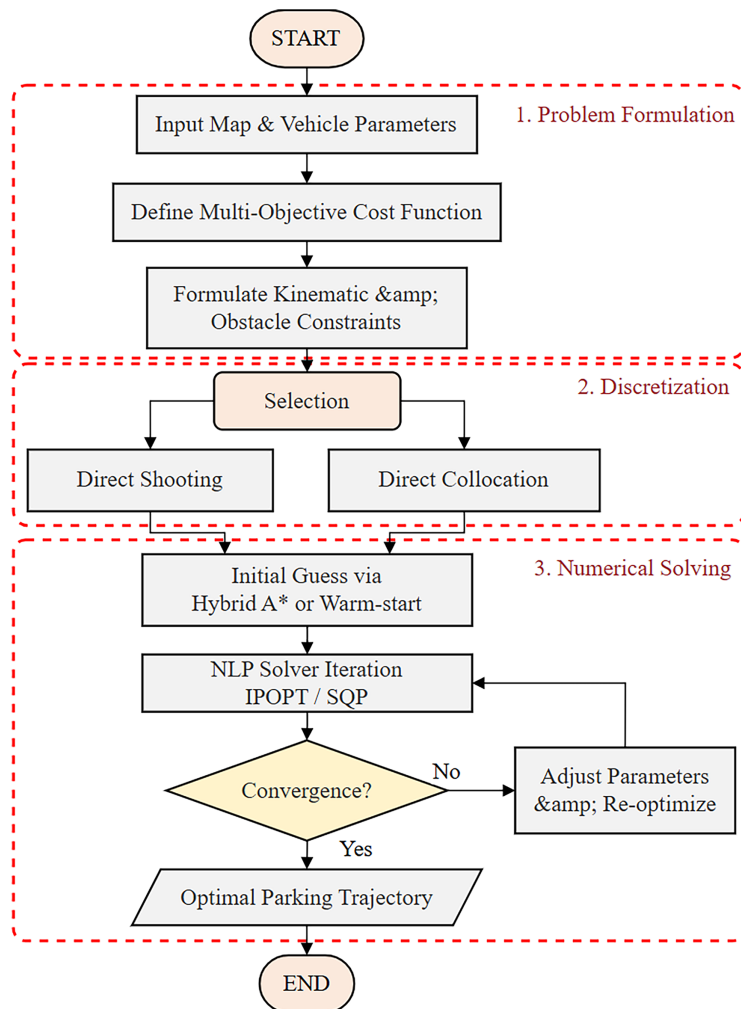


Figure 26: Flowchart of numerical optimization methods for AVP.

5.1.3 Final Representation of the Parking Problem

After establishing constraints and constructing the objective function, for example, with the shortest parking time as the optimization objective, the entire optimal parking control problem can be described in the following form:

$$\begin{aligned}
 J &= t_f - t_0 \\
 \forall t &\in [0, t_f] \\
 \text{constraint equation}
 \end{aligned}
 \tag{34}$$

5.2 Solution of Numerical Optimization Methods

5.2.1 Direct Shooting Method

The direct shooting method shown in Fig. 27 is a numerical approach that transforms a continuous optimal-control problem into a nonlinear programming problem. Its core idea is to parameterize the control variables, perform forward simulation, and construct an optimization problem. First, the continuous control input is discretized into a finite number of parameters, such as piecewise-constant controls. Starting from the given initial state, the vehicle dynamics are integrated numerically, for example with a Runge-Kutta method, to obtain the state trajectory and terminal state. The terminal-state deviation and process cost form the objective, together with any path and actuator constraints. An optimization algorithm, such as an interior-point method or sequential quadratic programming, then iteratively adjusts the control sequence until a feasible trajectory is obtained.

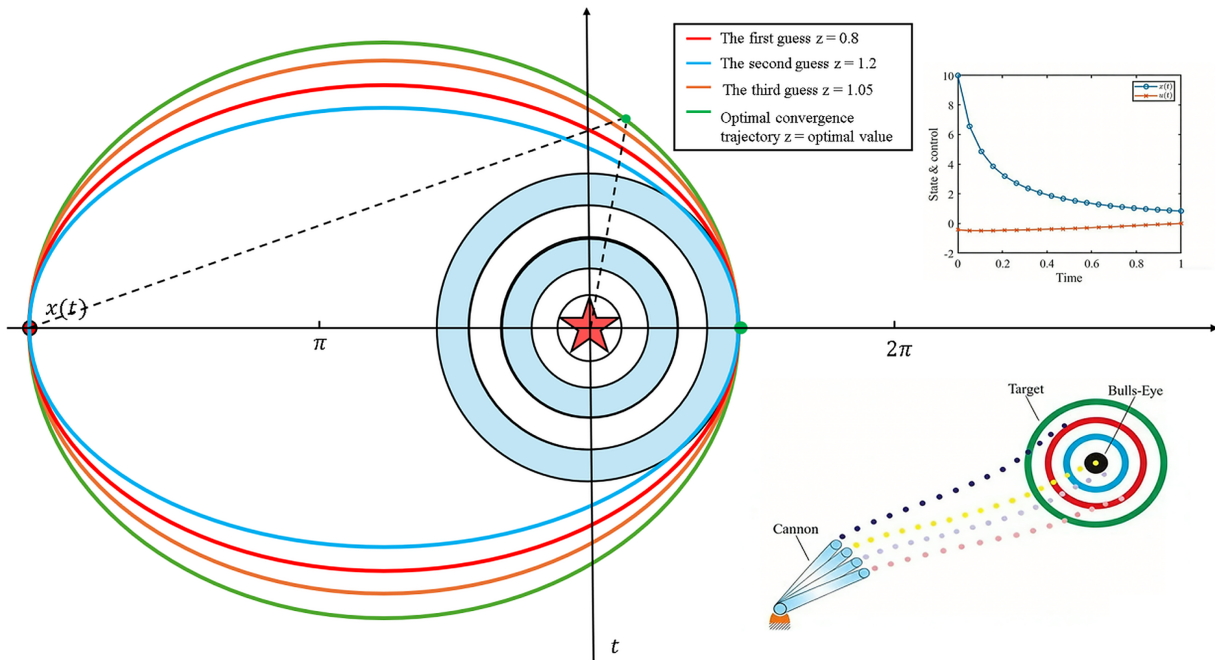


Figure 27: Diagram of the direct shooting method.

5.2.2 Direct Collocation Method

The direct collocation method is a mainstream numerical method for trajectory optimization. Its core idea is to discretize both state variables and control variables simultaneously, and by enforcing the validity

of the dynamic equation at the selected collocation points, the continuous optimal control problem is transformed into a nonlinear programming problem. This method divides the time interval into multiple segments, selects collocation points (such as Gaussian points or equidistant points) in each segment, and approximates the state trajectory using polynomials (such as Lagrange interpolation); the key step is to require that the dynamic differential equation of the system is exactly satisfied at these collocation points, thereby converting it into a series of algebraic equality constraints. Thus, the original problem is transformed into a nonlinear optimization problem with all state and control variables at the collocation points as decision variables, and with path inequality constraints as conditions. It can be efficiently solved by solvers such as interior point methods shown in Fig. 28, and has the advantages of numerical stability and controllable accuracy, making it a standard method for generating high-precision trajectories in applications such as automatic parking.

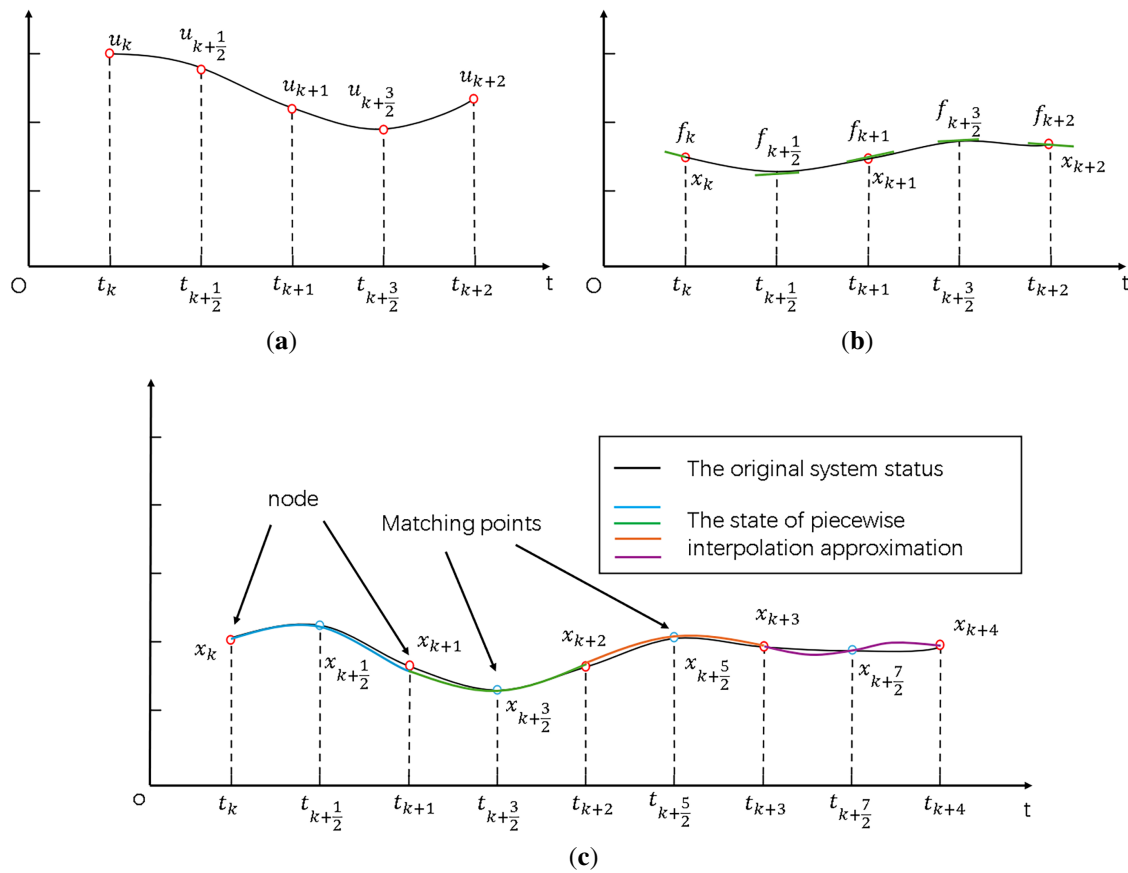


Figure 28: Hermite-simpson direct interpolation point method diagram. (a) State interpolation over one interval; (b) collocation-defect evaluation; (c) piecewise hermite-simpson trajectory approximation.

5.2.3 Sequential Quadratic Programming

The core idea of SQP is to approximate the original problem as a quadratic programming sub-problem at the current iteration point—the objective function is approximated by the second order, and the constraints are approximated by the first order. Solving this QP sub-problem yields a search direction, and then a line search is performed along this direction to update the iteration point until convergence. Its advantage is that it has a local superlinear convergence rate. However, it is necessary to calculate the gradients and Hessian

matrices (or approximations) of the objective function and constraints. For large-scale problems, it is crucial to utilize their sparse structure [96].

5.2.4 Interior Point Method

Interior Point Methods are powerful NLP solvers for large-scale and highly constrained problems. The classical barrier method incorporates inequality constraints into the objective through barrier functions and solves a sequence of barrier subproblems by Newton-type iterations. The primal-dual interior point method directly applies Newton's method to the perturbed Karush-Kuhn-Tucker conditions and simultaneously solves for the primal variables, dual variables, and slack variables. During the solution process, the iteration points remain inside the feasible region.

5.3 Common Cases of Numerical Optimization Methods

5.3.1 One-Stage Solution Method

Liu et al. proposed a chance-constrained trajectory optimization method based on conservative approximation. In terms of problem modeling, its core innovation is to first extend the deterministic parking trajectory planning problem to a chance-constrained optimization problem considering model and external uncertainties, allowing constraints to be satisfied with a certain probability, which significantly reduces conservatism compared with robust optimization [97].

$$\psi(k, s) = \begin{cases} 2 \frac{k + \beta_1}{k + \beta_2 e^{-ks}} - 1 & \text{if } s \geq s_0 \\ 0 & \text{if } s < s_0 \end{cases} \quad (35)$$

The convergence of this approximation is proved theoretically, and an efficient solution is achieved using Monte Carlo sampling and a gradient descent algorithm. This method achieves a balance between safety and optimality (such as parking time) in uncertain environments by adjusting parameters.

Kim et al. proposed a vertical parking trajectory planning method based on Model Predictive Control (MPC). In problem modeling, its innovation is to integrate the vehicle kinematics and approximate clothoid model through the concept of "virtual traction distance" shown in Fig. 29 to construct a linear time-varying system model with clothoid curve parameters as control inputs. This allows the trajectory planning task to be directly described as a constrained MPC problem [98]. In terms of solution method, this study uses the MPC framework to convert the planning problem into a Quadratic Programming (QP) problem for online solution in each control cycle. The constraint set encodes both actuator physical limitations and geometric path requirements simultaneously. This method realizes the integration of "planning-control", can handle noise and constraints in real time, generate smooth and feasible trajectories, and is computationally efficient due to the convex optimization nature of the problem.

Qiu et al. proposed a hierarchical coupled automatic parallel parking method combining Gaussian Pseudospectral Method (GPM) and Model Predictive Control (MPC). In problem modeling, a clear hierarchical architecture is adopted: the upper trajectory planning layer models the parking problem as a continuous-time optimal control problem with the shortest time as the objective, including complete kinematic, physical, and polygonal collision avoidance constraints; the lower tracking layer is responsible for trajectory tracking [99]. The innovation in the solution method lies in adopting different optimization strategies for the upper and lower layers and effectively integrating them: the planning layer adopts the Gaussian pseudospectral method, converts the problem into a nonlinear programming problem after Legendre-Gauss collocation discretization, and solves it using the interior point method, thus efficiently generating a globally optimized

time-state trajectory; the tracking layer adopts MPC to achieve precise tracking through online rolling optimization based on the linear error model.

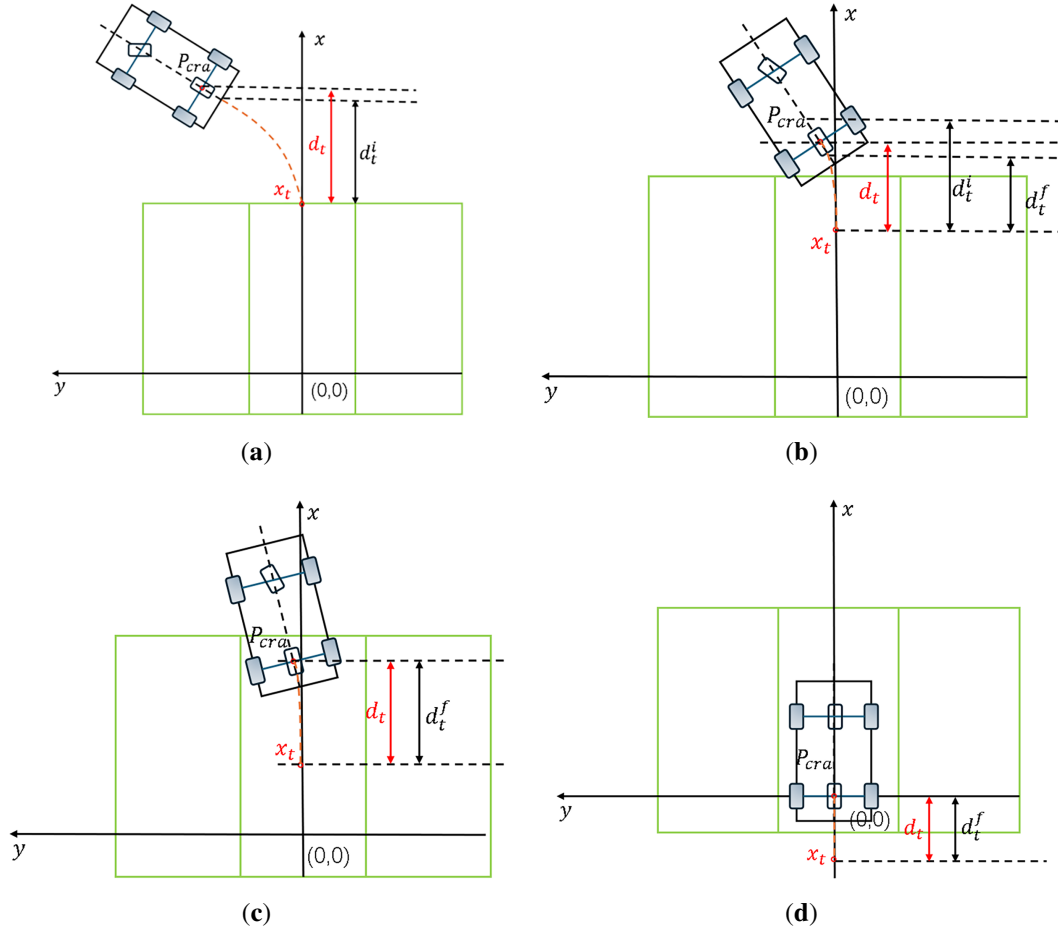


Figure 29: Illustration of the virtual towing method in reverse parking. (a) Initial approach configuration; (b) transition toward the parking-space centerline; (c) heading-alignment configuration; (d) terminal parking pose.

This framework takes into account both the quality of global optimization and the real-time robustness of local tracking.

Song et al. proposed a hybrid planning method combining offline nonlinear programming and online Monte Carlo Tree Search (MCTS). In problem modeling, it innovatively adopts a two-stage decoupling strategy: in the offline stage, parking is modeled as a complete trajectory optimization problem considering time optimality (and comfort), and the collocation method is used to discretize it into a large-scale nonlinear programming for solution to generate a high-quality dataset; in the online stage, the planning problem is remodeled as a sequential decision-making problem (Markov Decision Process), and the improved MCTS is used to search the action sequence online instead of directly solving the complete optimization problem [100].

$$\min (t_f - t_0) + w_1 \cdot \sum_{k=1}^{N-1} I(v_{k+1} \cdot v_k < 0) + w_2 \cdot \sum_{k=1}^{N_e} u_k \quad (36a)$$

The core innovation in the solution lies in knowledge transfer and search guidance: using the optimal trajectory data obtained from offline NP solution, two neural networks are trained—a policy network for predicting the probability distribution of actions (providing search prior) and a value network for evaluating state quality; the online MCTS search is guided by these two networks, which greatly reduces the simulation depth and computational overhead and achieves real-time performance.

$$a_t(s) = \operatorname{argmax}_a \left(Q(s, a) + c_{puct} P(s, a) \frac{\sqrt{\sum_b N(s, b)}}{1 + N(s, a)} \right) \quad (36b)$$

In addition, this method directly encodes comfort constraints such as jerk into the action expansion logic of MCTS, and designs an online heuristic rule using historical failure experience to further improve the search success rate and efficiency in narrow spaces.

Hu et al. proposed a low-dimensional optimization path planning method based on geometric features [101]. The method avoids the conventional formulation based on vehicle-dynamics differential equations and uses the geometric characteristic that the time-optimal parallel parking path is mainly composed of multiple arc segments. With the shortest path as the objective, the parking task is formulated as a nonlinear programming problem consisting of algebraic equations. The decision variables are the positions (x, y) and orientations (φ) of the connection points between adjacent arc segments. In Eq. (37), the arc-length objective is minimized subject to arc-geometry, continuity, boundary, and collision-avoidance constraints containing x, y , and φ . This formulation reduces the dimension of the optimization variables by approximately 76%.

$$\begin{aligned} \min_{x, y, \varphi} I &= \sum_{k=1}^N |R_k(\varphi_{k+1} - \varphi_k)| \\ \text{subject to } g_j(x, y, \varphi) &= 0, j = 1, \dots, m_e \\ h_l(x, y, \varphi) &\leq 0, l = 1, \dots, m_i \end{aligned} \quad (37)$$

In terms of constraint construction, their innovation lies in proposing a simplified collision avoidance strategy based on analytic geometry. By analyzing the relative positional relationship between the path arc and the boundary line of the rectangular obstacle, a set of geometric discrimination criteria was obtained, thus ensuring the collision-free nature of the entire arc segment by verifying only a few key points (such as the endpoints of the arc), which greatly simplifies the scale and form of the constraints. In practical applications, by using the sequential quadratic programming solver SNOPT, this low-dimensional nonlinear programming problem can be efficiently solved. The essence of this method lies in extracting and utilizing the strong geometric prior knowledge of the optimal path to fundamentally reshape the problem structure, while ensuring the quality and safety of the solution and significantly improving the computational efficiency.

In the study [85], Li et al. innovatively proposed a parallel stitching method to realize online trajectory replanning for sudden environmental changes during automated parking planning; real-world experiments have demonstrated the effectiveness and completeness of the proposed replanner with fast and high solution quality. Also, Li et al. reported a new time-optimal parallel parking planning by establishing a unified dynamic optimization framework with parking planning constraints; an interior-point method (IPM)-based solution is introduced to solve nonlinear programs (NLP) problems [92]. In the work [102], Li et al. systematically proposed a safe driving corridor method suitable for automatic parking, and the core innovation of this study is to adapt the concept of safe flight corridor, and a construction and constraint modeling method of Safe Traveling Corridor (STC) is also designed, aiming at the non-point geometry of

the vehicle, Li et al. simplifies the rectangular vehicle into two disks covering the vehicle body, convert the vehicle into two mass points through equivalent transformation, and expand the environmental obstacles at the same time, on this basis, a series of representative points are selected along the rough trajectory generated by hybrid A*, and a rectangular STC is generated for each representative point. By requiring the two equivalent mass points to always be within their respective STCs, the global complex geometric collision avoidance constraints are completely converted into a series of linear inequality constraints, so that the scale of the optimization problem is independent of the complexity of environmental obstacles. In the solution strategy, this work adopts the collocation method to discretize the continuous-time optimal control problem into a nonlinear programming problem, and selects the interior point method for solution. The objective function of the optimization problem is to minimize the parking time, and the constraint system fully includes vehicle kinematics, actuator limits, and STC constraints. Through corridor processing, this method fundamentally avoids the numerical solution difficulties caused by large-scale and non-convex collision constraints in traditional methods, and shows high solution efficiency and robustness in various random obstacle scenarios. It is worth noting that Li's concept of safe corridors can be applied not only to vehicle parking planning but also to other motion planning scenarios for autonomous driving, especially unstructured underground mining areas.

5.3.2 The Method of Solving in Stages

Gao et al. proposed a fast parking path-planning algorithm based on static optimization, which decomposes the parking task into two stages [103]: the first stage guides the vehicle from the starting point to a transition point, and the second completes precise parking from the transition point to the parking space. The algorithm adopts a reverse-planning strategy: a safe exit path is first generated and then followed in reverse for parking. The Runge-Kutta integration is used in the path discretization. At each discrete step, a static optimization problem determines the next control variables, including steering curvature and step size, while it minimizes position and heading deviations from the target. For collision avoidance, non-convex obstacles are first decomposed into convex components using a dedicated convex-decomposition procedure. Minkowski sums are then used to construct configuration-space obstacles for the convex components, after which collision avoidance is expressed as inequality constraints.

Liu et al. [104] proposed a two-stage trajectory-planning method for vertical parking. The maneuver is formulated as a two-stage dynamic optimization problem with a C-shaped obstacle-avoidance stage followed by a terminal straight-parking adjustment stage. The time horizon is divided accordingly, and a Gaussian pseudospectral discretization based on Legendre-Gauss collocation is applied independently in each subinterval. The resulting nonlinear programming problem is solved offline to generate high-precision, dynamically feasible trajectory data that can support subsequent fast online applications, such as decision-tree queries. This staged discretization improves the alignment between the numerical formulation and the physical parking process while enabling flexible allocation of computational effort.

He et al. proposed a three-stage data-driven trajectory planning method combining Gaussian pseudospectral optimization and fuzzy logic for parallel parking scenarios shown in Fig. 30, aiming to solve the problem that traditional numerical optimization has large online computation and is difficult to meet the demand for fast parking [105]. Its core idea is "offline optimal sampling + online fuzzy mapping": first, decompose the parking process into three stages (approaching, C-shaped parking and terminal adjustment) according to human driving experience, and establish a dynamic optimization model with the shortest parking time as the performance index and multiple constraints including parking space geometric boundaries, vehicle speed, acceleration and front wheel steering angle based on the Ackermann steering model; then propose a three-stage Gaussian pseudospectral method, discretize each segment of state and

control variables through Legendre-Gauss collocation and Lagrange interpolation, convert the infinite-dimensional DOP into a nonlinear programming problem, and solve the path in three stages, which greatly reduces the solution time.

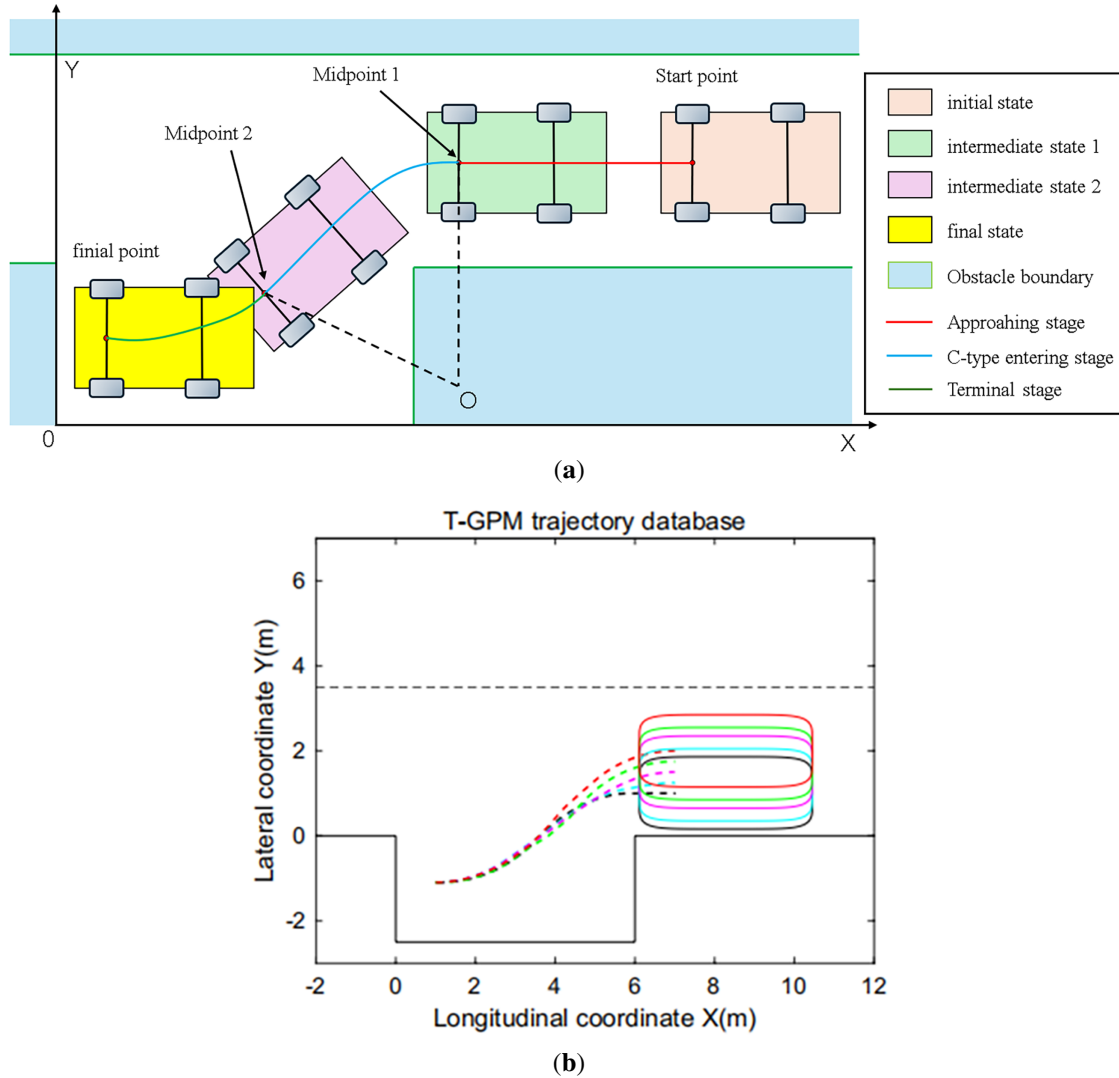


Figure 30: Three-stage parking schematic diagram. (a) Three-stage parking process and intermediate vehicle poses; (b) offline T-GPM trajectory database.

5.3.3 Optimization Solving Method

Zhang et al. proposed a differentiable collision avoidance constraint reconstruction method based on strong duality for optimized trajectory generation. The main innovation of this study is to use the strong duality of convex optimization to accurately reconstruct the originally non-differentiable collision avoidance constraints (such as polyhedral or ellipsoidal obstacles) into a smooth and differentiable constraint form, so that optimization algorithms based on gradients and Hessians (such as Interior Point OPTimizer, IPOPT) can be directly used for solution [106]. The implementation process is divided into three steps: first, model obstacles and controlled objects as convex sets (or unions of convex sets), and define collision constraints based on distance functions or signed distances; second, by introducing dual variables λ (point mass model) or λ and

μ (full-dimensional model), convert the original constraints into a set of smooth expressions including linear equalities and inequalities and dual norm constraints; finally, construct a nonlinear programming problem including system dynamics, state input constraints and reconstructed collision constraints, and solve it through numerical optimization. The characteristics of this method lie in the accuracy and smoothness of constraint reconstruction, supporting full-dimensional object obstacle avoidance, and being able to handle “minimum intrusion” trajectory generation when collision is unavoidable. Experiments show that it has real-time solution capability in tight environments such as automatic parking.

Zhang et al. proposed a collision-free trajectory planning method based on a virtual protection framework [107]. A geometric collision-detection criterion is constructed by introducing the virtual protection framework and amplification parameters; it ensures collision-free motion over the entire time interval between adjacent configurations after discretization. The automatic parking problem is formulated as a constrained dynamic optimization problem including vehicle dynamics, collision avoidance, and physical constraints. The dynamic equations are discretized by the direct multiple shooting method, with fourth-order Runge-Kutta integration applied in each shooting interval. A Hybrid A* trajectory is used to initialize the NLP, after which the trajectory and the minimum amplification parameter α are updated iteratively until convergence.

$$\min_{\{n(k)\}_{k=1}^N} T \cdot N \quad (38)$$

s.t.

$$\begin{aligned} \xi_0 &= \xi_S, \xi_N = \xi_F, \\ \xi_{k+1} &= \xi_k + \frac{T}{6} \cdot (k_1 + 2k_2 + 2k_3 + k_4), \\ \xi_{\min} &\leq \xi \leq \xi_{\max}, u_{\min} \leq u \leq u_{\max}, \\ &\neg \left(\overline{E}_{k \rightarrow k+1}^f \cap \overline{O}_j \right), \\ \forall k &= 0, 1, \dots, N-1, \forall j = 1, 2, \dots, J. \end{aligned} \quad (39)$$

The direct multiple shooting explicitly considers the continuous evolution between configurations, uses geometric coverage to ensure collision avoidance over the complete maneuver, and avoids the limitations of checking only discrete points. The resulting nonlinear programming problem is solved with CasADi, and the reported experiments demonstrate safe and trackable trajectories in several parking scenarios.

Micelli et al. proposed a path planning method based on the numerical solution of the Hamilton-Jacobi-Bellman (HJB) equation with limited maneuver times [108]. The innovation of this study is to model vehicle motion planning as an optimal control problem of switched systems, convert the unbounded minimum time problem into a bounded value function problem through Kruzkov transformation, and then solve it through a sequence of Quasi-Variational Inequalities (QVI) iterations. The implementation process includes: first, model the vehicle as a switched system with two modes (forward and backward), and define obstacle regions and target sets in the state space; second, discretize the HJB equation through the finite difference method, converting the continuous-time problem into a fixed-point iteration problem on a discrete grid; then, use the contraction mapping iteration method to solve each QVI, and gradually generate the optimal value function sequence from 0 to K_{max} maneuvers; finally, synthesize the actual trajectory based on value function interpolation and feedback control. The characteristics of this method are that it explicitly limits the number of direction changes, which is more suitable for the need to reduce gear shifting operations in actual

parking, and ensures deterministic solutions in complex obstacle environments through grid discretization and iterative solution, but the computational load increases with the refinement of the state grid.

5.3.4 Optimizing the Rough Path

Wang et al. [109] proposed a hierarchical parking trajectory planning method by combining an improved RRT* algorithm with non-convex optimization. The first layer uses improved RRT* and Reeds-Shepp curves to generate an initial path rapidly, and the second layer applies nonlinear optimization to obtain a curvature-continuous trajectory. During rough-path generation, a fast rejection test eliminates clearly non-intersecting geometric pairs before detailed collision checking, and the sampling region is divided according to the parking scenario to improve efficiency. The vehicle and obstacles are modeled as convex sets; signed-distance theory and the strong duality of convex optimization are then used to reformulate the collision-avoidance conditions as smooth constraints with dual variables. The resulting non-convex program includes kinematic constraints, convex-set distance constraints, and control-smoothness constraints and is initialized with the rough path for numerical solution.

Lian et al. proposed a two-layer trajectory planning framework for narrow and complex environments [110]. Its core innovation lies in systematic improvements in both rough trajectory acquisition and optimization modeling. In the path search stage, aiming at the low efficiency of traditional hybrid A* in narrow channels, the authors proposed an enhanced hybrid A* algorithm. This method first obtains a global rough path through A*, then constructs a driving corridor and intelligently discriminates wide/narrow channels, and only performs hybrid A* search between key boundary points, thus efficiently generating high-quality dynamic initial guesses. In terms of optimization modeling, the study proposed a two-stage driving corridor construction strategy to balance computational efficiency and solution quality with different expansion steps; at the same time, to handle “if-else” type speed constraints in local areas (such as speed bumps), a differentiable approximation formula based on the LogSumExp function is introduced to convert non-smooth logical constraints into continuous optimization problems. In implementation, the framework models the problem as an optimal control problem with time and control smoothness as objectives, and the constraints include vehicle kinematics, driving corridor boundaries, and smoothed local state constraints. Finally, the primal-dual interior point method (IPOPT) is used for numerical solution, and the solution quality is improved through iterative optimization and corridor reconstruction. This method shows efficient and robust planning capabilities in narrow environments.

Zhang et al. proposed a Hierarchical Optimized Collision Avoidance Algorithm (H-OBICA). Its innovation is mainly reflected in the accurate and smooth modeling of collision constraints and the hierarchical solution strategy [111]. The study reconstructs the non-convex and non-differentiable geometric collision avoidance conditions into a set of smooth equality and inequality constraints based on signed distance theory by introducing dual variables, thus fully embedding the obstacle avoidance problem into the continuous optimization framework. In the solution strategy, the method designs a hierarchical initialization process: first, use the hybrid A* algorithm to quickly search a rough path satisfying the simplified vehicle model in the discretized state space; then, extract initial guesses of states, controls and time steps based on this path, and initialize the dual variables of collision constraints, so as to “preheat” the subsequent numerical optimizer with this high-quality initial guess. The final optimization problem aims to minimize time and control effort, and the constraints cover complete vehicle dynamics, actuator limits, and smoothed collision avoidance conditions. This nonlinear programming problem is solved using the interior point method (IPOPT). This method successfully combines the global heuristic ability of search algorithms with the advantage of numerical optimization to generate smooth and dynamically feasible solutions, significantly improving the solution success rate and efficiency in complex parking scenarios.

5.3.5 Multi-Objective Optimization

Zhang et al. [112] proposed an improved fuzzy goal programming method for the multi-objective trajectory optimization problem of automatic parking, its innovation is mainly reflected in the multi-objective optimization strategy, which effectively handles conflicts and uncertainties between objectives through fuzzy goal programming. The optimization objectives are: minimizing completion time, minimizing path length, and maximizing motion smoothness. Traditional weighted sum methods need to preset fixed weights, while goal programming methods rely on decision-makers to provide precise expected values for each objective, both of which are difficult to apply in practice. The FGP method proposed in [112] combines fuzzy set theory, introduces membership functions to describe the satisfaction degree of objectives, and uses a payoff table to assist decision-makers in determining the tolerable upper and lower limits of each objective, thereby converting the multi-objective optimization problem into a single-objective problem of maximizing total membership.

Minimizing the squared-speed integral (motion-cost surrogate)

$$\min J_1 = \int_0^{t_e} v^2(t) dt \quad (40)$$

Minimizing the smoothness index

$$\min J_2 = \int_0^{t_e} \dot{c}^2(t) dt \quad (41)$$

Minimizing the completed time

$$\min J_3 = t_e \quad (42)$$

In implementation, Zhang et al. [112] simultaneously optimizes three objectives: the squared-speed motion cost defined by Eq. (40), the motion smoothness index, and completion time. A complete optimization model is constructed with vehicle kinematics, state and control input constraints, terminal conditions, scene geometric constraints, and continuous collision-avoidance constraints. The model is numerically solved using IPOPT. Simulation and experimental results show that the improved fuzzy goal programming method achieves a coordinated trade-off among the conflicting objectives and improves the practicality and flexibility of the planning results.

Numerical optimization methods play a core role in automatic parking path planning. Their basic process is to establish objective functions (such as shortest time, minimum energy consumption) and various constraints (kinematics, obstacle avoidance, control quantities, etc.), and transform the continuous optimal control problem into a nonlinear programming problem through discretization, which is commonly solved by Sequential Quadratic Programming (SQP) and the Interior Point Method. Researchers mainly improve planning efficiency and quality through optimizing solution algorithms (such as handling uncertainty, constructing safety corridors), staged solution (decomposing tasks into two or three stages), and optimizing rough paths. These methods work synergistically to efficiently generate smooth, safe, and trackable parking trajectories under complex constraints.

In multi-objective optimization, model-predictive-control formulations commonly combine several indicators in a weighted cost function while enforcing hard constraints such as obstacle avoidance and actuator limits. Fuzzy goal programming is complementary to these numerical optimal-control formulations: it represents acceptable ranges for objectives such as squared-speed cost, completion time, and motion smoothness through membership functions, thereby expressing a compromise among competing objectives. It does

not replace the vehicle-dynamics model, trajectory discretization, or nonlinear solver; instead, it supplies a preference formulation that can be solved within a conventional numerical optimization framework.

6 Path Planning Method Based on Artificial Intelligence

Automatic-parking path planning must balance real-time performance, comfort, and obstacle-avoidance safety in narrow spaces with nonholonomic constraints. Data-driven methods based on deep learning and reinforcement learning have therefore become active research directions. Deep-learning models learn mappings from parking data, while reinforcement-learning policies can be trained through interaction with an environment; many reinforcement-learning implementations are model-free at the policy level. These methods can support end-to-end trajectory generation or decision making, but their performance remains dependent on the chosen network or policy architecture, training data, sensor inputs, and safety-validation mechanism.

In the field of automatic parking path planning, the integrated application of machine learning, deep learning, and reinforcement learning technologies has broken the limitations of traditional numerical optimization and sampling search methods, constructing a technical framework with high efficiency, adaptability, and generalization ability. Its core architecture and technical characteristics can be summarized as follows.

From the perspective of the overall framework shown in Fig. 31, most relevant technical solutions adopt a hierarchical collaborative architecture, divided into two stages: “upper-layer coarse path generation” and “lower-layer trajectory optimization/tracking”. The upper layer relies on deep learning or reinforcement learning models to quickly output feasible parking trajectories, while the lower layer completes refined trajectory adjustment through optimization algorithms or control strategies. For example, the HALOES framework first generates a coarse trajectory using Deep Reinforcement Learning (DRL), then achieves obstacle avoidance and trajectory smoothing through nonlinear optimization; the RDNN architecture uses Recurrent Neural Networks (LSTM) to learn the temporal correlation features of optimal trajectories and combines transfer learning to adapt to different vehicle models. Reinforcement learning-based schemes also introduce federated learning to realize distributed training among multiple vehicles, which not only improves model generalization but also ensures data privacy.

Machine learning methods realize trajectory classification, regression, or decision-making through statistical modeling and feature learning of parking data. Typical representatives include Gaussian Process Regression (GPR), Support Vector Machines (SVM), and decision trees, which are suitable for fast decision-making and trajectory fitting of parking strategies [113].

Implementation of Gaussian Process Regression (GPR) for parking trajectory generation involves three steps. First, a feature vector X is constructed from parking features such as the relative distance to the target space, heading angle deviation, and minimum obstacle clearance. Second, the control sequence over a fixed prediction horizon is represented as a finite-dimensional output vector y . Separate scalar GPR models can be trained for each control channel and future time step, or a multi-output GPR model can be used to represent correlations among the outputs. Finally, the real-time feature vector is input to the trained model to predict the control sequence, while the predictive mean and covariance provide confidence information for trajectory evaluation and safety check.

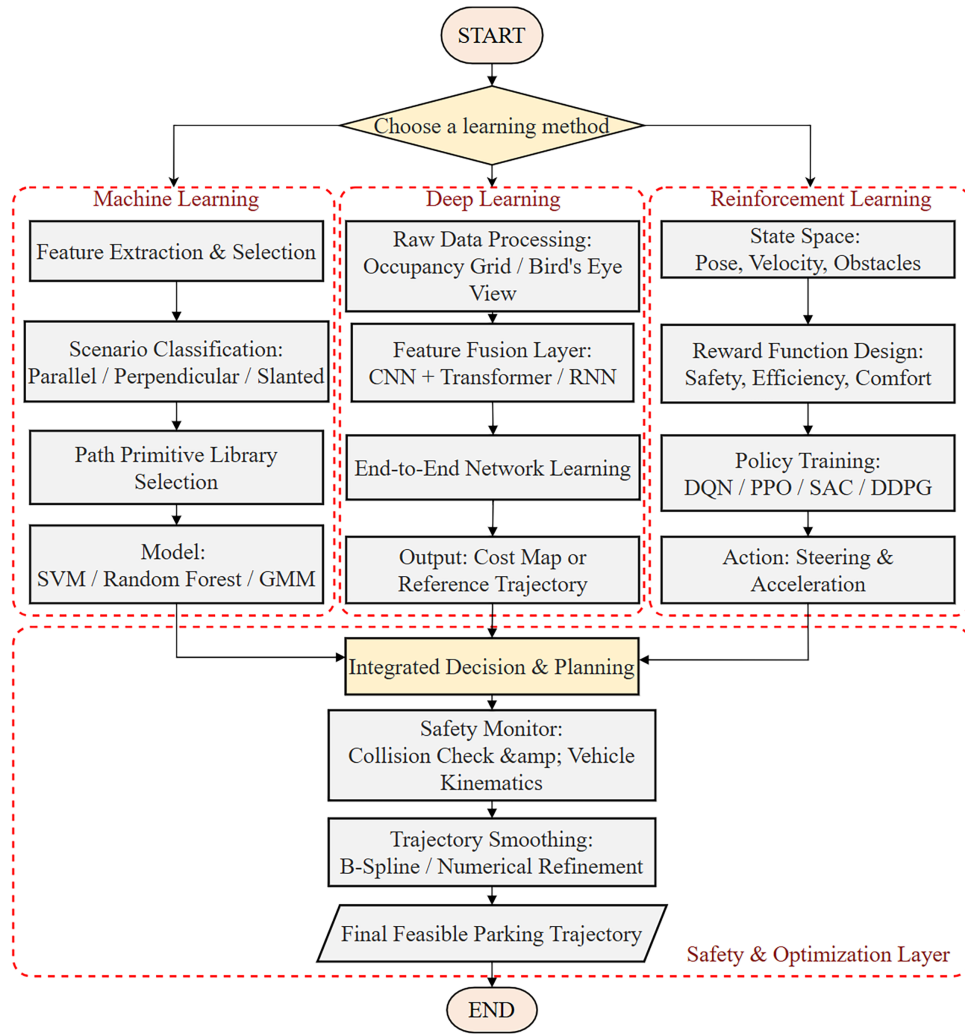


Figure 31: Flowchart of artificial intelligence methods for AVP.

6.1 Decision Tree/Random Forest Methods

The implementation follows three stages: First, a Parking Strategy Dataset is constructed by collecting decision samples under various scenarios (e.g., timing of forward/reverse gear shifts, steering angle magnitude) and labeling them with outcome tags (such as “successful”, “safe”, or “collision”). Second, a decision tree model (e.g., CART or Random Forest) is trained using environmental features (like remaining space and obstacle positions) and vehicle states as inputs to learn explicit parking decision rules. Finally, for online operation, the trained model outputs real-time parking strategies—such as shift timing and steering commands—based on the current scenario features.

Within decision-tree and random-forest methods, interpretable supervised-learning models map environmental and vehicle-state features to parking decisions or trajectory classes. Numerical optimization can supply high-quality offline trajectories for training data, whereas decision trees and random forests provide fast online inference from engineered features. Their generalization depends on feature design, the representativeness of the training set, and the need to retrain or update the model when the parking environment changes substantially.

the Ackermann steering model to respect vehicle dynamics. Second, the action space is defined as continuous control variables, typically acceleration and steering rate or angle, bounded by actuator limits. Third, a shaped reward function balances multiple objectives: it penalizes collisions and excessive time, rewards accurate pose alignment, and encourages control smoothness. Finally, deep reinforcement learning algorithms—such as DDPG, PPO, or SAC—are employed to train a policy network that maps states to optimal control actions, enabling the vehicle to learn parking maneuvers through iterative interaction with the environment [116–119].

Reference [117] adopts a federated DDPG-based hierarchical framework; the upper Actor-Critic layer generates a coarse parking trajectory through federated DDPG, and the lower numerical-optimization layer refines the trajectory to satisfy kinematic and collision-avoidance constraints. In five types of narrow parking scenarios, the method reduces planning time by 12.15%–66.02% relative to Hybrid A* while maintaining collision-free trajectories.

Reference [118] adopts the SAC algorithm, introduces policy entropy into the objective function to enhance the agent's ability to explore the environment, and designs a Segmented Parking Training Framework (SPTF), which decomposes complex parking tasks into two subtasks: “pose adjustment” and “parking space entry”, increasing the parking success rate in narrow parking space scenarios from 71% to 93%.

6.2.2 Deep Learning-Based End-to-End Plannings

Data-Efficient Reinforcement Learning: Reference [116] proposes Data-Efficient Reinforcement Learning (DERL), which evaluates parking states through truncated Monte Carlo Tree Search (MCTS) shown in Fig. 33, guides the search using policy and value networks, and introduces trajectory return weighting and virtual trajectory enhancement strategies to reduce the demand for real interaction data. The algorithm can converge with only 25 parking trial-and-error runs in a Carsim simulation.

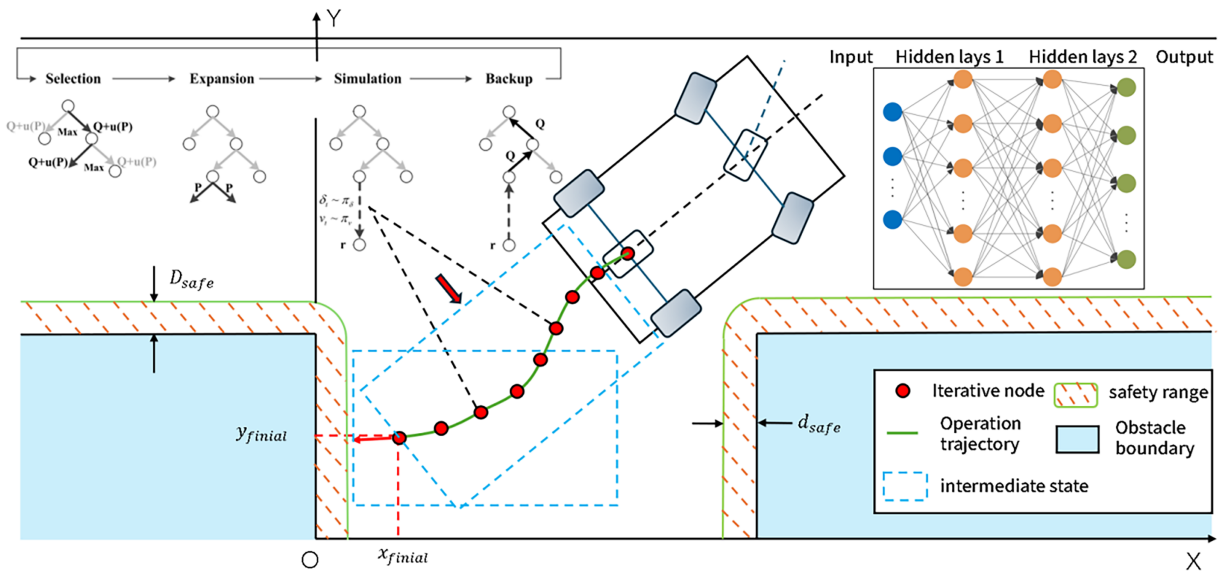


Figure 33: Parallel parking P-MCTS single iteration process and lateral policy network structure.

A multi-objective reward formulation can combine target-pose accuracy, collision avoidance, path guidance, and control smoothness to balance safety, accuracy, and comfort during policy learning.

6.2.3 Federated Reinforcement Learning Methods

To address data privacy and model generalization issues in multi-vehicle parking scenarios, Federated Reinforcement Learning (FRL) realizes multi-agent policy coordination through decentralized training.

(1) Construction of Federated Training Architecture. The central server initializes the global policy model (such as Actor-Critic network parameters) and distributes it to each vehicle agent; each agent completes policy training in its local environment (different parking scenarios) and only uploads model parameters instead of raw data to ensure data privacy. (2) Model Aggregation and Iterative Optimization. The central server aggregates the model parameters of each agent through the federated averaging algorithm to generate a globally updated model; each agent downloads the global model and continues local training until the model converges.

6.2.4 Data-Efficient and Hierarchical Reinforcement Learning

The upper layer generates coarse trajectories through federated DDPG, and the lower layer realizes trajectory refinement through numerical optimization. In 5 types of narrow parking space scenarios, the planning time is reduced by 12.15%~66.02% compared with the hybrid A* algorithm, while ensuring collision-free trajectories. It solves the problem of insufficient generalization of models trained in single scenarios, and the model maintains a success rate of over 90% in parallel, vertical, and oblique parking scenarios through federated training with multi-scenario data.

6.3 Deep Learning-Based Path Planning for Automatic Parking

Deep learning methods fit the “state-trajectory” mapping relationship of parking trajectories through neural networks, which are suitable for parking scenarios with offline training and online rapid invocation. Their core is to construct a high-precision trajectory fitting network and improve model generalization ability [120].

6.3.1 Recurrent Deep Neural Network (RDNN) Methods

To address the problem that traditional fully connected neural networks cannot capture trajectory temporal dependencies, recurrent neural networks based on LSTM/GRU are applied to parking trajectory planning.

6.3.2 Implementation Process

This approach consists of three sequential stages: offline dataset preparation, neural network training, and online deployment [121]. First, an offline trajectory dataset is generated by computing a large set of optimal parking trajectories through numerical optimization methods (e.g., the Gaussian pseudospectral method). This dataset, shown in Fig. 34, covers diverse scenarios with varying initial poses, parking space types, and obstacle layouts and contains paired sequences of vehicle states and corresponding optimal control inputs. Second, a specialized Recurrent Deep Neural Network (RDNN) architecture is constructed, typically using a dual-network design to separately model acceleration and steering control sequences. The model is trained using an optimizer like Adam to minimize the error between predicted and optimal trajectories. To enhance accuracy, a Data Aggregation strategy is employed to iteratively reintroduce high-error trajectory samples into the training set. Finally, in the online phase, the trained RDNN rapidly generates a reference trajectory based on the real-time vehicle state. This trajectory is then refined and validated by a subsequent optimization module (e.g., Optimization-Based Collision Avoidance) to ensure strict adherence to physical and safety constraints, such as obstacle avoidance.

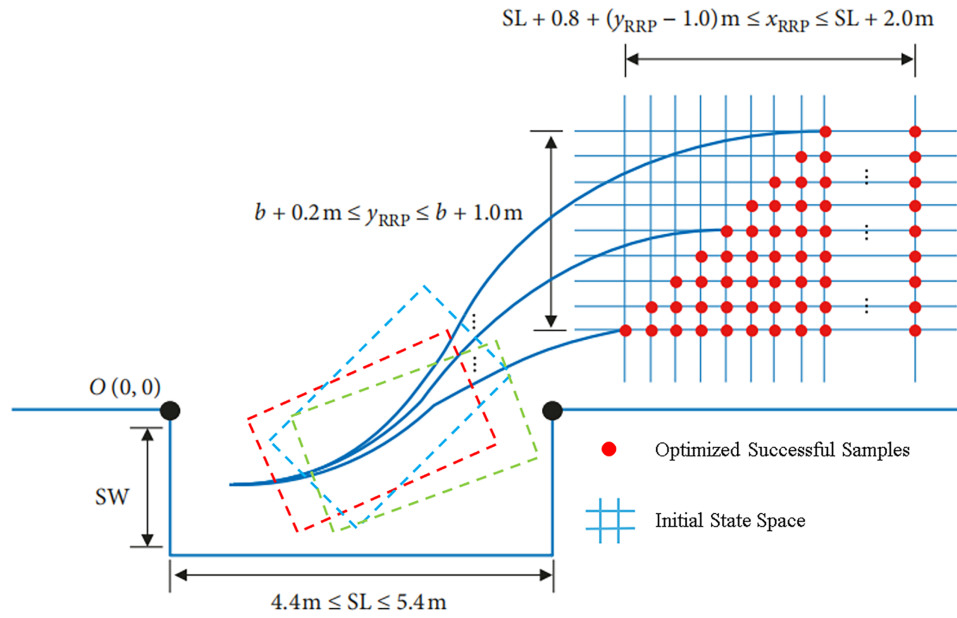


Figure 34: Schematic illustrating zone and RRP of parking for collecting training dataset.

6.3.3 Deep Transfer Learning and Initial Trajectory Optimization for Parking Trajectory Planning

To enhance the performance of automatic parking planning, reference [122] proposed an integrated architecture based on a recurrent deep neural network (RDNN) and transfer learning, effectively addressing the problem of insufficient generalization ability of deep learning models when applied across different vehicle models. This research first trained the basic planner on vehicle A (the source model, with a wheelbase of 2.848 m), and then through two transfer strategies, namely “fixing the feature extraction layer and fine-tuning the dedicated dynamics layer” and “introducing the domain adaptation layer”, it enabled it to quickly adapt to vehicle B (the target model, with a shortened wheelbase of 2.500 m and a weight reduction of approximately 25%).

The reported results show that, after transfer to vehicle B, the terminal trajectory error was 40% lower than that of the non-transferred model. By modeling temporal dependencies, the RDNN produced smoother and more continuous curvature and steering profiles than the baseline fully connected network. These results indicate that transfer learning can improve cross-vehicle adaptation while maintaining trajectory smoothness and passenger comfort.

Traditional numerical optimization methods (such as nonlinear programming) have demonstrated extremely high accuracy in automatic parking trajectory planning, capable of rigorously handling vehicle nonholonomic constraints and obstacle avoidance requirements, and generating smooth and high-quality paths. However, their drawbacks are also quite significant: due to the highly non-convex nature of the parking environment, the optimization solution process is highly dependent on the initial guess value. If the initial value is not properly selected, the algorithm is prone to getting stuck in local optima or not converging at all; in addition, the iterative calculation under large-scale constraints takes a long time, making it difficult to meet real-time requirements.

To balance the accuracy of numerical optimization with the real-time performance of deep learning, Reference [123] proposed an innovative integrated framework of “deep learning prediction and numerical optimization refinement”. This study first utilized CNN or Transformer networks to efficiently extract environmental features (such as parking space contours and obstacle point cloud distributions), and combined

vehicle status to output a “rough” initial trajectory. Its superiority lies in that the deep learning model, with its powerful nonlinear fitting ability, can provide a highly potential feasible region initial value within milliseconds by learning from expert experience. Subsequently, starting from this initial value, a nonlinear programming problem incorporating kinematic and collision avoidance constraints was constructed, and refined iterations were carried out using IPOPT or SQP solvers. This “determine direction first, then seek precision” strategy not only overcomes the drawbacks of numerical optimization being sensitive to initial values and prone to converging to local optima, but also significantly enhances the algorithm’s convergence speed and real-time response capability in dynamic scenarios, achieving a leap from “offline optimal” to “online real-time optimal” for parking trajectories.

7 Geometry-Based Algorithms

Geometry-based algorithms use geometric curves to generate collision-free and smooth paths between initial and terminal vehicle poses while satisfying vehicle kinematic requirements. In automated parking, these methods are attractive because of their relatively simple implementation and high computational efficiency. According to the path-generation principle, they can be divided into curve-combination methods and curve-interpolation methods [124].

7.1 Curve Combination Methods

Curve combination methods plan a collision-free path connecting the initial and terminal poses of a vehicle by using linear segments, circular arcs, and other curves under the condition of low-speed vehicle movement. This approach originates from Dubins curves proposed by Dubins [125] and Reeds-Shepp (RS) curves proposed by Reeds and Shepp [126]. Both two types of curves consist of tangent line segments and circular arcs, where each arc segment has a consistent curvature and satisfies the minimum curvature constraint. Specifically, Dubins curves only allow the vehicle to move forward and are the shortest curves connecting two points in a 2D plane with specified initial and terminal tangent directions, as shown in Fig. 35a. RS curves, on the other hand, add reverse curves to Dubins curves, enabling the vehicle to perform reverse driving operations, as illustrated in Fig. 35b. The increased curve elements enhance the flexibility of path planning, making RS curves significantly superior to Dubins curves in complex and narrow unstructured scenarios.

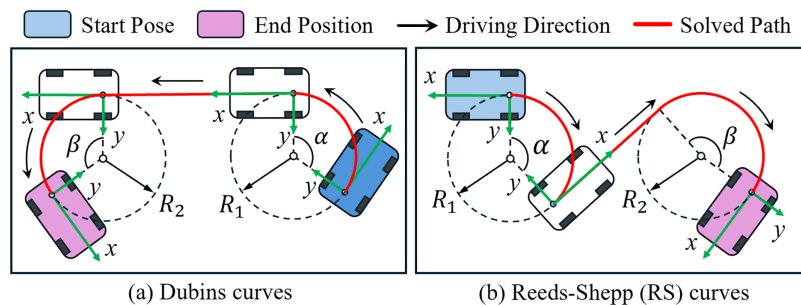


Figure 35: Dubins curves and RS curves.

In the motion planning for automatic parking, the curve combination method typically involves two main steps: The first step is an enumeration process, where 48 classic curve combinations from the RS curve set are attempted to connect the initial and terminal poses. Specific parameters of each curve are solved to establish a candidate curve set. The second step is a two-layer filtering process for the candidate curve set:

the first layer filters the optimal path subset through the minimum curvature constraint and cost function; the second layer retains safe and feasible curves via collision detection.

Among these steps, the two-layer screening process is the core of the curve combination method. In the first layer, the minimum curvature constraint requires that the radius of the total n circular arcs in the path must be greater than or equal to the minimum turning radius R_{min} , as shown in Eq. (43). The cost function generally takes path length as the objective, prioritizing the retention of the shortest curves. After preliminary screening, double circular arcs and circular arc-linear-circular arc combinations are usually retained for parallel parking, as shown in Fig. 36; while linear-arc-linear combinations are typically retained for perpendicular parking.

$$R_1 = R_2 = \dots = R_n = R \geq R_{min} \quad (43)$$

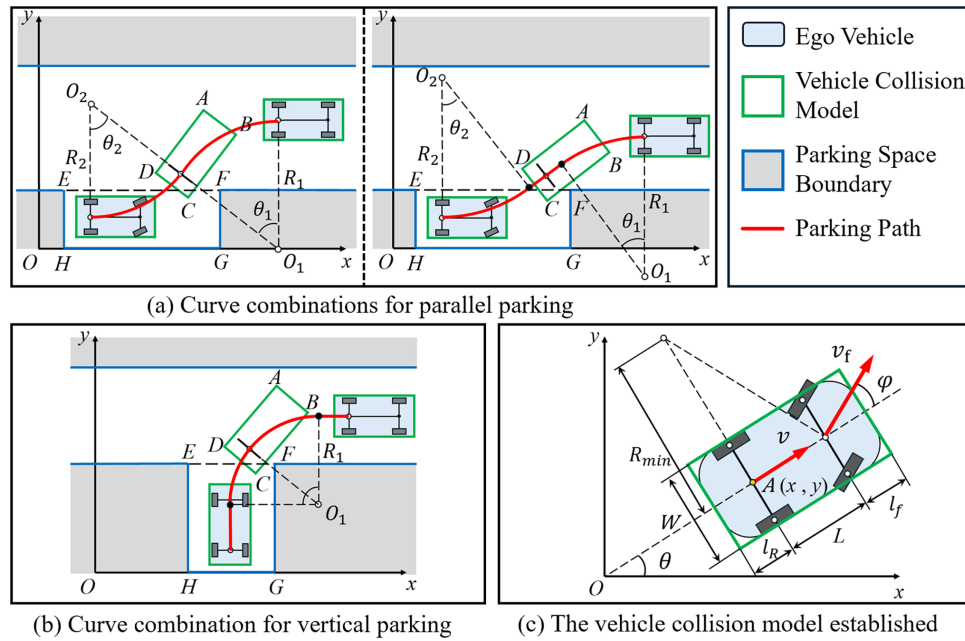


Figure 36: Common basic path models.

In the second layer, to perform collision detection, it is necessary to first establish a vehicle collision model. The vehicle is generally abstracted as a rectangle with $(l_f + L + l_r)$ in length and W in width, as shown in Fig. 36. During parking, the four corners of the vehicle body are very close to boundaries in certain scenarios, so collision detection focuses on these high-risk situations. Considering the impact of control accuracy and response time errors, a safety distance δ is usually reserved between the vehicle model and obstacles. Only curves where the collision model has no intersection with surrounding obstacles and maintains the safety distance are retained. High-risk scenarios vary across different parking scenarios; common high-risk scenarios for parallel parking and perpendicular parking are shown in Fig. 37, and the corresponding collision-free requirement expressions are presented in Eqs. (44)–(49).

$$y_{boundary} - y_{O_1} \geq |O_1A| + \delta_1, (\theta \geq \alpha) \quad (44)$$

$$y_{boundary} - y_{O_1} \geq |O_1A| \cdot \cos(\alpha - \theta) + \delta_2, (\theta < \alpha) \quad (45)$$

$$|O_2F| \geq |O_2B| + \delta_3 \quad (46)$$

$$y_{O_2} \geq |O_2C| + \delta_4 \quad (47)$$

$$x_{O_1} - x_E \geq |O_1D| + \delta_5 \quad (48)$$

$$R_1 - \frac{W}{2} \geq |O_1C| + \delta_6 \quad (49)$$

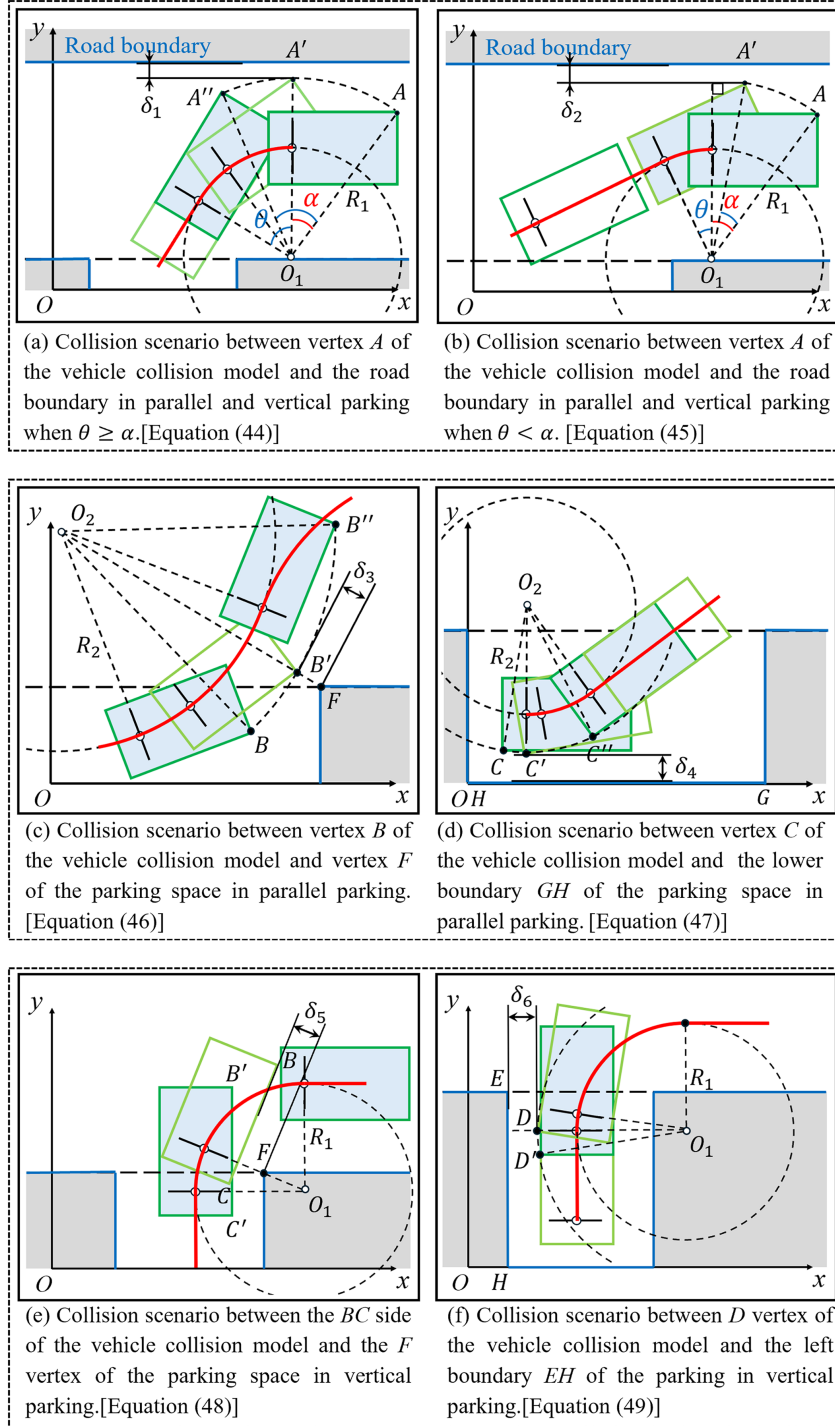


Figure 37: Common high-risk collision scenarios.

Traditional curve combination methods with a flowchart shown in Fig. 38 are widely used in the motion planning for automatic parking due to their simple structure and high generation efficiency. In many studies, RS curves are combined with graph search algorithms or sampling-based methods to ensure the kinodynamic feasibility of paths solved by these two types of algorithms. The previously mentioned Hybrid A* algorithm [37] and the improved bidirectional RRT* algorithm with RS curves used by Jhang et al. [127] are two typical examples.

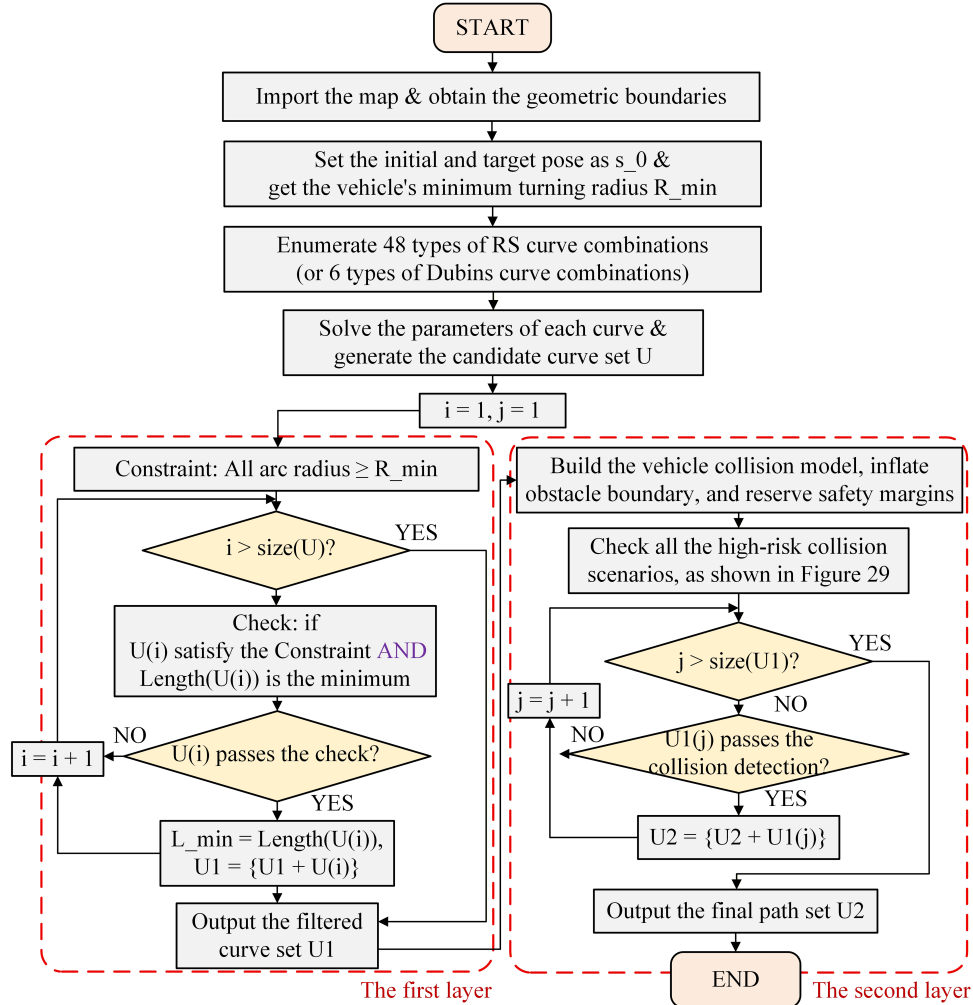


Figure 38: Flowchart of curve combination methods for AVP.

However, the curve combination method has three unavoidable fatal limitations: Firstly, due to the use of a fixed turning radius, it exhibits low path flexibility and space utilization. A slight change in the initial or terminal heading angle can lead to significant variations in the planned path. Secondly, the cost function only minimizes path length, resulting in frequent steering and gear shifting in the solved path, which greatly reduces driving comfort. Thirdly, there are curvature discontinuities at the intersection points of two adjacent curves, forcing the vehicle to stop and steer in place frequently, thereby increasing the burden on the steering mechanism and tires. Additionally, deviations between the actual steering position and the theoretical intersection point may lead to parking failure. To address these three shortcomings, recent

research has mainly focused on two directions: improvements to the cost function and optimizations of curve combinations.

First of all, adding new penalty terms to the cost function can significantly improve path quality and reduce the number of steering and gear shifting operations. Hong et al. [128] designed an RS path generation method based on a heuristic cost function. The function weighted and integrated path distance, gear switching, steering changes, path type, and reverse driving operations to select the optimal path from the set of collision-free paths. Simulation results showed that this method can generate safe paths in narrow parking spaces that are closer to the natural driving and parking behaviors of human drivers.

Column 2 of Table 1, C represents the arc Cycle; S represents a Straight line. ‘|’ indicates a change in the vehicle’s driving direction at the end of the curve. The subscript ‘ β ’ indicates that the central angles of the two arcs are equal, and ‘ $\pi/2$ ’ indicates a 90° arc; Column 3 of Table 1, L and R indicate the Left and Right directions of the arc Cycle, respectively. The superscript ‘+’ indicates forward, and ‘-’ indicates backward.

Besides, optimizations of curve combinations can be divided into two directions: One aims to enhance the flexibility of path generation by allowing RS curves to use a variable curvature radius. For narrow perpendicular parking scenarios, Yin et al. [129] proposed a variable-radius RS curve. First, RS curves were classified into five structural categories, as shown in Table 1, and corresponding variable-radius calculation formulas were derived for each category to ensure radius variability across different structural curves. Then, random functions were used to generate multiple sets of candidate curves with a feasible variable radius. Finally, a cost function integrating path length and steering changes was used to select the optimal path.

Table 1: Five structural categories of RS curves.

Categories by Yin et al. [129]	Basic Curve Combinations	Sequences of Motion Primitives
A. Three arc composition	C C C	$(L^+R^-L^+)$ $(L^-R^+L^-)$ $(R^+L^-R^+)$ $(R^-L^+R^-)$
	CC C	$(L^+R^+L^-)$ $(L^-R^-L^+)$ $(R^+L^+R^-)$ $(R^-L^-R^+)$
	C CC	$(L^+R^-L^-)$ $(L^-R^+L^+)$ $(R^+L^-R^-)$ $(R^-L^+R^+)$
B. Both ends of the arc and line segment composition	CSC	$(L^+S^+L^+)$ $(L^-S^-R^-)$ $(R^+S^+R^+)$ $(R^-S^-R^-)$
		$(L^+S^+R^+)$ $(L^-S^-R^-)$ $(R^+S^+L^+)$ $(R^-S^-L^-)$
C. Four-end arc composition	$CC_\beta C_\beta C$	$(L^+R_\beta^+L_\beta^-R^-)$ $(L^-R_\beta^-L_\beta^+R^+)$ $(R^+L_\beta^+R_\beta^-L^-)$ $(R^-L_\beta^-R_\beta^+L^+)$
	$C C_\beta C_\beta C$	$(L^+R_\beta^-L_\beta^-R^+)$ $(L^-R_\beta^+L_\beta^+R^-)$ $(R^+L_\beta^-R_\beta^-L^+)$ $(R^-L_\beta^+R_\beta^+L^-)$
D. Three circular arcs and a line segment	$C C_{\pi/2}SC$	$(L^+R_{\pi/2}^-S^-R^-)$ $(L^-R_{\pi/2}^+S^+R^+)$ $(R^+L_{\pi/2}^-S^-L^-)$ $(R^-L_{\pi/2}^+S^+L^+)$
		$(L^+R_{\pi/2}^-S^-L^-)$ $(L^-R_{\pi/2}^+S^+L^+)$ $(R^+L_{\pi/2}^-S^-R^-)$ $(R^-L_{\pi/2}^+S^+R^+)$

(Continued)

Table 1 (continued)

Categories by Yin et al. [129]	Basic Curve Combinations	Sequences of Motion Primitives
	$CSC_{\pi/2} C$	$(L^+S^+L_{\pi/2}^-R^-)$ $(L^-S^-L_{\pi/2}^+R^+)$ $(R^+S^+R_{\pi/2}^-L^-)$ $(R^-S^-R_{\pi/2}^+L^+)$ $(R^+S^+L_{\pi/2}^+R^-)$ $(R^-S^-L_{\pi/2}^-R^+)$ $(L^+S^+R_{\pi/2}^+L^-)$ $(L^-S^-R_{\pi/2}^-L^+)$
E. Four arcs and a line segment are composed	$C C_{\pi/2}SC_{\pi/2} C$	$(L^+R_{\pi/2}^-S^-L_{\pi/2}^+R^+)$ $(L^-R_{\pi/2}^+S^+L_{\pi/2}^-R^-)$ $(R^+L_{\pi/2}^-S^-R_{\pi/2}^+L^+)$ $(R^-L_{\pi/2}^+S^+R_{\pi/2}^-L^-)$

The other aims to address curvature discontinuity by connecting the linear segments and the circular arcs with curves that have continuous curvature. In research focused on parallel parking, Zou et al. [130] adopted the double-arc method. First, a parking initial region was generated based on constraints, and path points were obtained by sampling within this region. Then, polynomial curves were used to connect each sampling point with the second arc to generate a set of reference trajectories. Daniali et al. [131] used a Clothoid sequence-linear method, in which each Clothoid sequence consisted of two Clothoid curves and one arc, as shown in Fig. 39. A multi-objective particle swarm optimization algorithm was employed to optimize Clothoid velocity parameters, and path sets were selected by comprehensively considering parking time and required space. Li and Wang [132] proposed a three-stage Clothoid-arc-quintic polynomial method: the Clothoid curve was used to connect the parking terminal, and the quintic polynomial was used to connect the parking initial position. A feasible initial region was determined based on constraints, and the optimal path was selected by solving a constrained optimization problem.

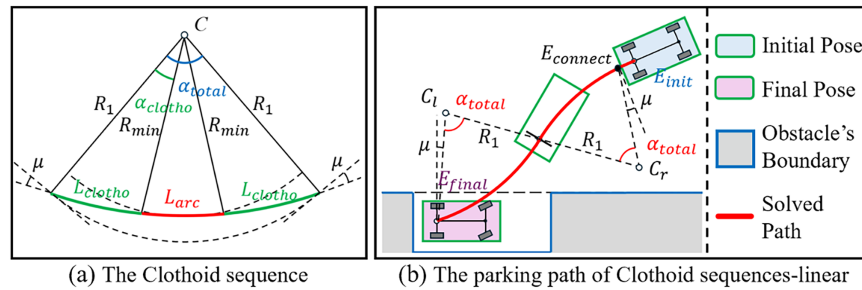


Figure 39: Schematic diagram of the improved path smoothed by Clothoid curves.

In research focused on vertical parking, Chen [133] focused on the scenarios with non-zero initial heading angles and used Clothoid curves to adjust the heading angle and ensure path smoothness. If one-time parking was feasible, the path was planned directly; otherwise, the heading angle was first adjusted to zero before path planning, and the parking range was limited by coordinate and distance constraints. Arasawa et al. [134] addressed the scenarios where the distance d between the initial position and the road edge is variable and used a combination of linear segments and circular arcs. According to the value of d , Clothoid curves or Sigmoid functions were used to connect the circular arc with the parking initial position, solving the problem of curvature discontinuity. Ding et al. [135] focused on scenarios with different road widths

and arbitrary initial heading angles, adopting a circular arc-linear method. A two-stage path was designed for wide scenarios, and a multi-stage path for narrow scenarios, as shown in Fig. 40. The two paths were smoothed by Clothoid curves, and safety thresholds were set to ensure distance from obstacles.

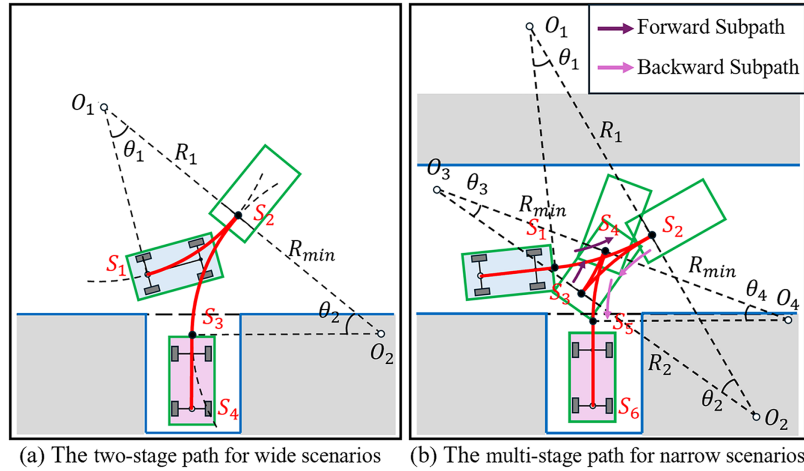


Figure 40: Schematic diagram of the two-stage and multi-stage paths.

7.2 Curve Interpolation Methods

Curve interpolation methods refer to a class of techniques that construct a smooth path connecting the start and end points based on specific curve models, guided by a series of given discrete points. Notably, depending on the underlying curve model, the generated path does not necessarily pass through every discrete point. In the motion planning of automatic parking, their core objective is to fit a set of collision-free points on the original parking path to create a subpath that satisfies dynamic constraints. Common curve models include the spline curve, the Bezier curve, and the B-spline curve.

They begin with the spline curve. A spline is defined as a continuous curve formed by connecting multiple segments of simple curves under smooth constraints. Given $n + 1$ known points $P_0, P_1, \dots, P_n$, the core idea of interpolation using a k -th order spline curve is to divide the entire interval into n subintervals separated by each point, and connect two adjacent points in each subinterval using a parameterized k -th order polynomial. Finally, the unknown parameters are solved through continuity conditions, including the start and end constraints of each polynomial and the equality of the first and second derivatives at the intersection points, thereby obtaining a smooth curve composed of n polynomial segments. Eqs. (50) and (51) represent the parametric and continuity conditions of the cubic spline curve, respectively. The value of t ranges between 0 and 1, where each value corresponds to a point on the polynomial curve, and $t = 0$ & 1 represents the start and end points, respectively.

$$\begin{cases} x = f_i(t) = a_i t^3 + b_i t^2 + c_i t + d_i, & t \in [0, 1] \\ y = g_i(t) = e_i t^3 + f_i t^2 + g_i t + h_i \end{cases} \quad (50)$$

$$\begin{cases} f_i(0) = x_i, f_i(1) = x_{i+1}, & i = 1, \dots, n \\ f'_i(1) = f'_{i+1}(0), & i = 1, \dots, n-1 \\ f''_i(1) = f''_{i+1}(0), & i = 1, \dots, n-1 \end{cases} \quad (51)$$

A Bezier curve is a parametric curve defined by a weighted sum of control points. Except for its endpoints under the standard Bernstein representation, it does not generally pass through the control points.

In two dimensions, the parametric Equation of an n -th order Bezier curve is shown in Eq. (52), where $\vec{P}_i$ denotes the $n + 1$ control points and $B_{n,t}(t)$ is the Bernstein basis function defined in Eq. (53). The Bernstein basis functions satisfy the partition-of-unity property $\sum_{i=0}^n B_{n,t}(t) \equiv 1$ for t in $[0, 1]$. Consequently, $P(t)$ is a weighted sum of the control points, with non-negative weights that vary continuously as t increases from 0 to 1, as illustrated in Fig. 41.

$$\vec{P}(t) = \sum_{i=0}^n \vec{P}_i \cdot B_{n,t}(t), t \in [0, 1] \quad (52)$$

$$\begin{cases} B_{i,n}(t) = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i}, i = 0, 1, \dots, n \\ \sum_{i=0}^n B_{i,n}(t) \equiv 1 \end{cases} \quad (53)$$

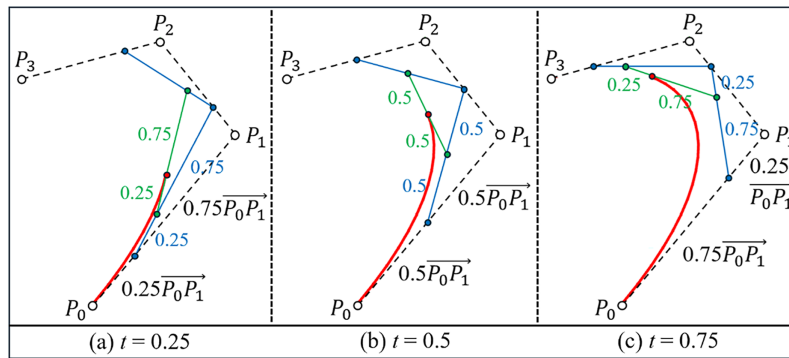


Figure 41: Schematic diagram of the Bezier curve generation process.

The limitations of Bezier curves lie in the inability to locally control the curve shape—changing the position of any control point affects the entire curve. Additionally, the order of a Bezier curve is related to the number of control points; an excessive number of control points leads to a sharp increase in computational complexity. These issues are addressed by the B-spline curve.

A B-spline curve of order k (degree $k - 1$) with $n + 1$ control points is a piecewise-polynomial curve defined over a knot vector, as shown in Eqs. (54)–(56). For a clamped knot vector with no repeated interior knots, the curve contains $n - k + 2$ non-zero knot spans; repeated knots may reduce the number of non-zero spans. The basis function $N_{i,k}(u)$ is evaluated by the Cox-de Boor recursion over the parameter u within the knot range. The basis functions depend on the relative knot spacing and are invariant under an affine scaling and translation of the complete knot vector; this invariance does not require the parameter u itself to be normalized. The recursive construction of the basis functions is illustrated by the pyramid model in Fig. 42.

$$\vec{P}(u) = \sum_{i=0}^n \vec{P}_i N_{i,k}(u), u \in (u_{k-1}, u_{n+1}) \text{ and } k \in [2, n + 1] \quad (54)$$

$$u \in \{u_0, u_1, \dots, u_{k-1}, u_k, \dots, u_n, u_{n+1}, \dots, u_{n+k-1}, u_{n+k}\} \quad (55)$$

$$N_{i,1}(u) = \begin{cases} 1, & u \in [u_i, u_{i+1}] \\ 0, & \text{Otherwise} \end{cases} \quad (56)$$

$$N_{i,k}(u) = \frac{u - u_i}{u_{i+k-1} - u_i} N_{i,k-1}(u) + \frac{u_{i+k} - u}{u_{i+k} - u_{i+1}} N_{i+1,k-1}(u)$$

$$\sum_{i=0}^n N_{i,k}(u) \equiv 1, u \in (u_{k-1}, u_{n+1}) \quad (57)$$

$$N_{i,k}(u) = \begin{cases} \geq 0, & u \in [u_i, u_{i+k}] \\ = 0, & \text{Otherwise} \end{cases} \quad (58)$$

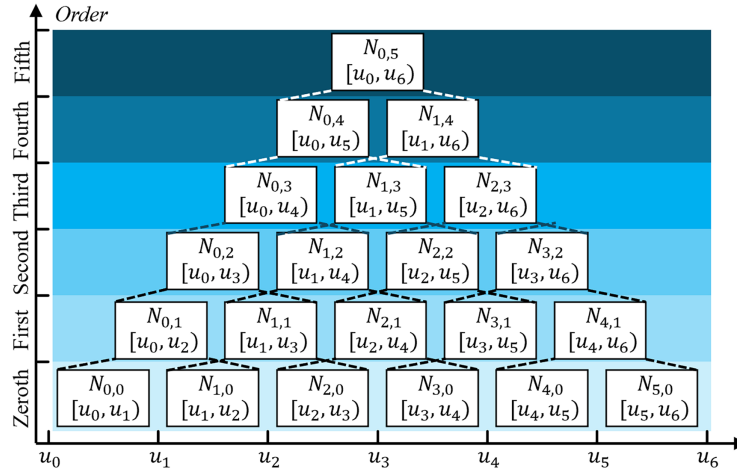


Figure 42: The pyramid model of the recurrence relation for B-spline curve basis functions.

In parking path planning, the methods for establishing constraints and detecting collisions in curve interpolation are consistent with those in the curve combination method, with the flowchart shown in Fig. 43, with the difference lying in curve generation. The key steps are the selection of interpolation points and the solution of the curve Equation. The selection of interpolation points involves setting interpolation points on the parking path according to constraint limitations; the solution of the curve Equation involves first establishing a system of Equations by combining the continuity requirements of the curve at each interpolation point with various constraints, then solving for the unknown parameters in the system of Equations. The core issue of interpolation methods lies in the selection of interpolation points. Recent research can be mainly divided into two directions: the direct selection method and the stepwise selection method.

7.2.1 Direct Selection Method

The direct selection method sets interpolation points based on key points of the parking scenario, typically including the initial point, parking point, and points prone to collision along the path. This method offers advantages such as low computational complexity, high flexibility, and strong real-time performance. If an obstacle shifts slightly during parking, the position of the interpolation point near it can be adjusted without reconstructing the entire path. However, it has disadvantages: the selection of points is highly dependent on fixed rules, making it prone to failure in scenarios with irregular parking spaces, narrow spaces, or multiple obstacles. Additionally, the generated path is often not the optimal solution, as point selection prioritizes path feasibility and struggles to balance multiple objectives such as parking efficiency and comfort. Vieira et al. [136] adopted fifth-order polynomials for path planning. First, a collision model was established using four circular arcs connected to the four corners of the vehicle. Then, a solution matrix was constructed based on the positions of the start and end points and derivative constraints. Finally, genetic algorithm optimization was used to ensure curvature continuity. Duong et al. [137] used third B-spline curves for path planning. The start and end points were selected as interpolation points, and the coordinate

relationship between the two points was established to satisfy curvature constraints. Then, the basis function was solved based on a recursive formula, and an expanded rectangular model was established for real-time collision detection.

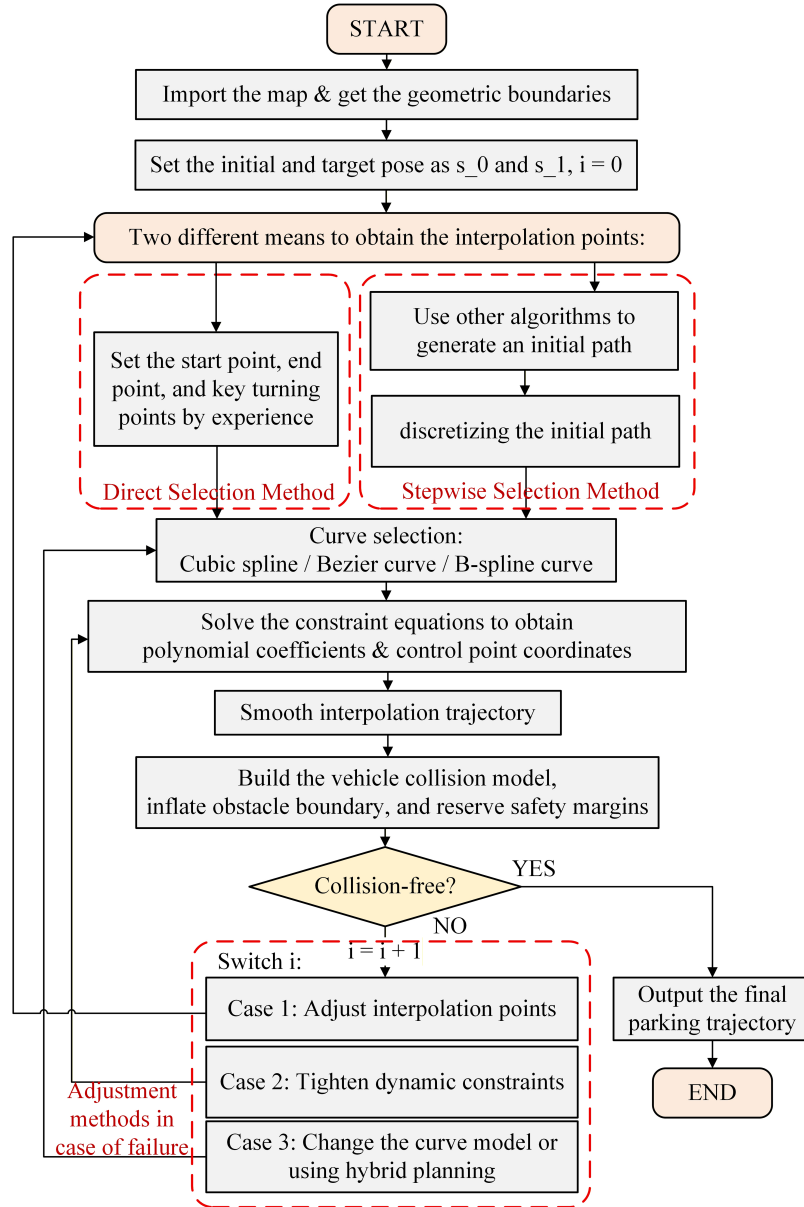


Figure 43: Flowchart of curve combination methods for AVP.

7.2.2 Stepwise Selection Method

The stepwise selection method derives interpolation points from an existing initial path. Yan et al. [138] generated a collision-free double-arc path and then applied a quasi-uniform cubic B-spline for smoothing the path. Guo [139] selected the start point, endpoint, and tangency points of one- or multi-arc initial paths and used a quartic spline to remove curvature discontinuities. Tang et al. [140] combined A* with RRT*FN for global planning and used a double cubic Bezier curve near the parking endpoint, with overlapping control

points enforcing continuity, as shown in Fig. 44. Hossain et al. [141] smoothed the initial path generated by Hybrid A by using a cubic spline and verified clearance with multi-circle collision detection. Li et al. [142] combined a geometric planner with numerical optimization for parallel parking; it utilizes the geometric solution as a feasible initialization for subsequent refinement. Lv et al. [143] optimized the final parking state and applied two-stage path planning in a dynamic parking lot while considering passenger boarding and alighting comfort. These studies show that stepwise selection improves feasibility and multi-objective path quality, although it requires an additional initial-planning stage and therefore increases computation.

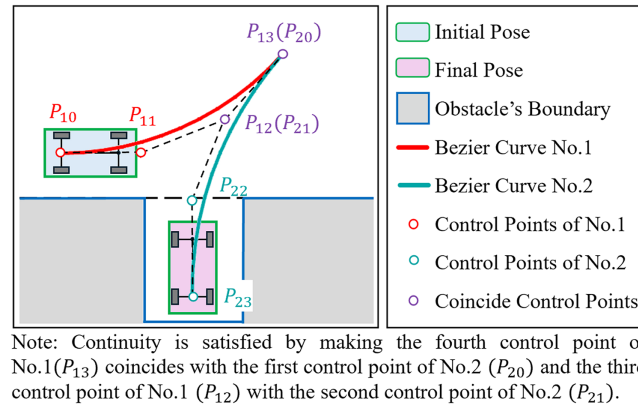


Table 2: Methodologies, internal limitations, external limitations for various parking trajectory planning.

Family	Internal Limitations					External Limitations		
Use Case	Real-Time	Adaptability	Stability	Smoothness	Replanning	Sensor Limitations	Environmental Complexity	Dynamic Environment
Graph Search-Based Algorithm								
-Dijkstra [rarely used]	+	+	++	-	-	++	+	-
-A* [31–34,45]	++	+	+	-	-	+	-	-
-Hybrid A* [36–44,46]	-	-	-	+	-	+	-	-
Sampling								
-PRM [47–50,55–59,72]	-	+	+	-	-	+	+	-
-RRT [48,55–60]	+	++	-	-	+	+	+	-
-RRT* [60–65]	-	++	-	-	-	+	+	-
-RRT-Connect [65–69]	+	+	-	-	+	+	+	-
APF [84,86–91,93]	++	-	-	-	++	-	-	++
Numerical Optimization								
One-stage [97–102]	+	+	++	-	+	-	++	+
multi-stage [103–106]	++	+	-	++	-	-	++	-
Optimize the solution method [107,108]	-	+	++	+	-	-	+	-
Optimizing the Rough Path [109–111]	+	++	+	++	+	-	++	+
Multi-objective Optimization [112]	+	+	++	++	+	-	-	-
AI								
reinforcement learning [113–117]	++	++	-	-	++	+	+	+
machine learning [118,119]	-	++	+	+	-	+	++	+
deep learning [120–123]	++	+	-	-	++	+	-	+
Geometry-Based Algorithms								
-Dubins & Reeds-Shepp [128–135,142,143]	++	-	+	+	+	-	-	-
-Curve interpolation method [136–141,144,145]	++	-	+	++	-	-	-	-

Artificial Potential Field (APF) methods are widely used for their high real-time efficiency and computational simplicity. Their main limitations are local minima, goal non-reachability, potential-barrier effects, sensitivity to potential-field gains, and possible oscillation in complex environments. APF methods rely on sufficiently reliable obstacle-state estimates for safe operation, but they do not inherently require higher sensor accuracy than all other planning categories. They can react quickly to environmental changes, although they are commonly combined with global planners to avoid local-minimum failures.

Numerical optimization approaches overcome computational bottlenecks through dimensionality reduction or data-driven techniques, achieving marked improvements in real-time performance. Paths generated under kinematic constraints are smooth, and stability is ensured through hierarchical control.

By leveraging techniques such as safe corridors, these methods demonstrate strong adaptability to environmental complexity and efficient dynamic obstacle avoidance. Although dependent on sensor accuracy, their overall performance is outstanding.

Multi-stage numerical optimization decomposes the parking maneuver into phases with stage-specific constraints and boundary conditions. This decomposition improves computational organization and adaptability to structured parking scenarios, while discontinuities at stage interfaces may reduce trajectory smoothness and stability. These methods depend on accurate environmental perception and generally have limited online replanning capability because a change in the environment may require the stage boundaries and optimization problem to be reconstructed.

Artificial intelligence has become an important research direction in autonomous parking. Reinforcement-learning methods, including SAC and model-based RL, learn parking policies through interaction and can support online execution and replanning in dynamic environments, although trajectory smoothness and stability still require refinement. Genetic algorithms are evolutionary optimization methods that can search for multi-objective solutions, whereas traditional machine-learning methods such as Gradient Boosted Decision Trees map engineered environmental features to parking decisions or motion commands. Deep-learning methods use multilayer neural networks, including recurrent neural networks, to learn nonlinear mappings from large offline datasets and provide low-latency inference. Their unconstrained outputs generally require post-processing, numerical refinement, or safety filtering, and their generalization depends on dataset coverage and augmentation.

Dubins and Reeds-Shepp curves rapidly connect initial and final poses and provide strong real-time performance, stability, and geometric smoothness, but their adaptability is limited because different scenes require different curve combinations. Collision checking can be performed against a pre-existing obstacle map. Real-time sensor input is required when the obstacle map must be constructed or updated online, particularly in dynamic environments.

Curve Interpolation Methods fit parameterized curves to discrete path points, offering advantages in real-time performance, stability, and smoothness. Yet, these methods heavily rely on the selection of control or interpolation points and are almost incapable of performing complete planning independently; this leads to poor adaptability. In complex environments, the required degree or number of curve segments increases significantly, and then the real-time performance is impacted substantially. Additionally, these methods are unable to handle dynamic targets.

8.2 Future Research Directions

Although significant progress has been made in trajectory planning for automated parking, bridging the gap between theoretical algorithms and commercial Level 4 Automated Valet Parking (AVP) requires addressing several critical challenges. Drawing from the reviewed literature, we identify four promising directions for future research: hybrid hierarchical architectures, uncertainty-aware robust planning, dynamic-obstacle avoidance, and vehicle-infrastructure cooperation.

A. Standardization of Hybrid Hierarchical Architectures

A single algorithm, such as Hybrid A* or RRT*, struggles to simultaneously meet the multiple demands of global route finding, obstacle avoidance, and kinematic smoothness in automated parking scenarios [146,147]. For example, while graph-search methods offer completeness, their computational efficiency decreases significantly in high-resolution maps. Numerical optimization methods can generate smooth and optimal trajectories but are highly prone to converging to local minima. Particularly in complex parking scenarios, no single algorithm can simultaneously satisfy the requirements of completeness, optimality, and

real-time computation. Therefore, the prevailing trend for the future is to combine different algorithms in a complementary manner, leveraging their core strengths to form a standardized and structured hierarchical parking planning architecture.

- (a) **Algorithm Fusion:** Resolution-complete graph-search methods or probabilistically complete sampling-based methods are employed to explore complex parking spaces and generate collision-free but potentially non-smooth coarse paths. The first stage finds a feasible path without overemphasizing trajectory quality. A subsequent optimization layer uses safe corridors or numerical trajectory optimization to improve smoothness and dynamic feasibility. The complementary integration of search, sampling, and optimization algorithms remains an important research direction in automated parking.
- (b) **Appropriate Warm-Start:** This constitutes the intermediate layer of the hybrid hierarchical architecture. The convergence speed and solution quality of nonlinear solvers such as IPOPT depend strongly on the initial guess [104]. State and control variables extracted from the coarse path form an initial sequence near a promising solution basin; this reduces computation time and the likelihood of convergence to a poor local minimum. The numerical solver then generates a smooth, dynamically feasible, locally optimal trajectory. How to obtain a reliable warm start remains a significant challenge.
- (c) **Enhancement of Algorithmic Properties and Complementary Synergy:** To address the low sampling efficiency of methods such as RRT in narrow parking spaces, sampling-based planners can be combined with approaches such as Artificial Potential Field (APF) to improve guidance. The attractive and repulsive forces of the potential field can guide tree growth, while the probabilistic completeness of the sampling-based method is retained and APF can accelerate convergence. The future will see increasing complementary integration of different planning algorithms.

B. Robust Planning for Uncertainty

Automated parking systems face severe challenges in practical applications, as real-world environments are neither ideal, static, nor deterministic. Sensor noise, perceptual occlusions, and behavioral uncertainties of surrounding dynamic obstacles (e.g., pedestrians, vehicles moving unexpectedly due to driver error) render traditional deterministic planning methods either overly fragile or conservative. A core direction for future research will involve actively quantifying risks, intervening in uncertainties, and performing parking re-planning.

Traditional robust optimization often assumes worst-case perceptual errors to guarantee safety, which can easily render the planner infeasible in tight spaces.

- (a) **Chance Constraints:** A major future direction is the use of chance constraints. The core idea acknowledges that absolute safety is unattainable; a minimal probability of constraint violation is accepted in exchange for greater planning feasibility. This is highly effective in narrow scenarios, allowing trajectories to approach obstacles more closely to complete automated parking and resolve certain deadlock situations. However, strict obstacle avoidance conditions still require careful verification and consideration.
- (b) **Deep Integration of Perception, Prediction, and Planning:** Planning algorithms alone cannot anticipate dynamic obstacles. Future development is inseparable from the deep integration of perception, prediction, and planning.
- (c) **AI Planning Integrated with Safety Guarantees:** Currently, AI is a major research hotspot and is widely used in autonomous driving. Although data-driven methods like Deep Reinforcement Learning and end-to-end learning demonstrate adaptive capabilities surpassing traditional planning algorithms in unstructured environments, their “black-box” nature and the lack of interpretability in their outputs hinder deployment. To bridge the gap between academic research and industrial application, future

work will not solely pursue end-to-end “large models” but will focus on building “trustworthy, verifiable” hybrid AI architectures, addressing the safety bottleneck.

Advanced AI planning systems must possess the capability to recognize their own cognitive limitations. This entails enabling the system to quantify uncertainties in both perception and prediction stages, and subsequently transmit these uncertainty metrics to the planning module. The planner can then dynamically adjust the conservatism of its behavior based on the assessed risk levels, adopting more cautious strategies in regions of high uncertainty.

In highly dynamic shared spaces, such as parking lots with mixed pedestrian and vehicle traffic, planning must evolve from unilateral optimization to multi-agent interactive game theory. This allows the vehicle to infer the latent intentions of other traffic participants and proactively communicate its own intent through interpretable motion patterns, thereby guiding interactions toward safer and more efficient outcomes.

C. Dynamic obstacle avoidance in automatic parking

In the graph search algorithm, from static topology to spatio-temporal search: Traditional graph search methods (such as A* and Hybrid A*) search for the optimal path under given conditions in a static map. However, in dynamic scenarios, their strategies evolve to expand in the spatio-temporal dimension. By introducing the time axis, the algorithm maps the predicted trajectory of obstacles into the “unreachable area” in a four-dimensional space, thereby planning paths that do not collide in both space and time in advance. Additionally, when the environment undergoes changes beyond predictions or the vehicle deviates from the planned path, by introducing a replanning mechanism, the improved algorithm will either use the current pose as the starting point and leverage the original algorithm to find a path to the destination (complete reset replanning), or clear a short distance ahead of the vehicle’s current position and search within a local range for a path that can bypass this sudden change and return to the main route (local replanning).

Fast re-planning and reconnection mechanism in sampling algorithms. The sampling algorithm (such as RRT*) is renowned for its ability to handle nonlinear constraints in high-dimensional spaces. During dynamic obstacle avoidance, it mainly employs a local re-sampling strategy: it retains most of the original branches while pruning and re-growing the local paths affected by dynamic obstacles. Combined with the RS (Reeds-Shepp) curve, it can generate obstacle avoidance branches that comply with the vehicle dynamics within an extremely short time; the exploration speed and obstacle avoidance accuracy can be balanced.

In the artificial potential field method, a dynamic repulsive field can provide one of the fastest local responses among the six method categories. Dynamic obstacles are treated as moving repulsive sources whose influence changes with position and velocity. When an obstacle approaches, the resultant-force direction changes and guides the vehicle to generate a timely avoidance displacement. This method has a small computational load and can respond promptly to sudden obstacles, but it generally requires cooperation with a global planner to avoid local minima during obstacle avoidance.

In numerical optimization methods, dynamic-obstacle avoidance can be formulated as a time-varying constrained optimization problem, for example within MPC or an IPOPT-based trajectory optimizer. A Safe Traveling Corridor (STC) abstracts the evolving environment into a sequence of convex spatial constraints. Within each control cycle, the optimizer computes a smooth trajectory that satisfies vehicle constraints, maintains clearance from dynamic obstacles, and remains within the available corridor.

Curve-interpolation methods use B-spline or Bezier curves to smooth a fixed set of waypoints and cannot independently predict or avoid dynamic obstacles. In a hybrid online replanning system, perception and prediction modules generate updated collision-free control points, and the interpolation module moves or adds local points to reshape the curve while preserving its overall trend. Therefore, dynamic obstacle

avoidance depends on the external replanning framework, while the interpolation method provides local path smoothness.

Integrated prediction-planning AI methods, including reinforcement-learning frameworks such as HALOES, learn obstacle-avoidance policies from extensive scenario data. Temporal sequence models such as LSTM can extract motion history and predict obstacle intent. Gaussian Process Regression has a different role: it can provide probabilistic regression and uncertainty estimates for a defined feature-to-control or feature-to-trajectory mapping, but it is not grouped here with LSTM as a temporal feature extractor. These components can be combined with numerical optimization in a hybrid architecture of AI decision-making, uncertainty estimation, and constraint-based trajectory refinement.

D. Vehicle-Infrastructure Cooperation and Infrastructure-Assisted Planning

As autonomous driving technology progresses toward SAE Level 4, future planning will gradually shift from a single-vehicle mode to a system-level intelligence led by infrastructure, leveraging a global perspective to address complex parking scenarios. Global path planning and multi-vehicle scheduling tasks will be offloaded to edge servers at the infrastructure side. Utilizing overhead surveillance cameras or Roadside Units (RSUs) deployed in parking lots, the infrastructure can obtain a global dynamic map without blind spots and compute globally optimal scheduling paths using algorithms like Mixed-Integer Linear Programming (MILP), coordinating the passage sequence of multiple vehicles uniformly.

Vehicle-infrastructure cooperation highly depends on low-latency, high-reliability V2X communication networks (e.g., 5G/C-V2X) [148–151]. Jitter or packet loss in the communication link may cause asynchrony between infrastructure-issued trajectory commands and the real-time vehicle state, posing safety risks.

Cloud-Edge-End Computation Offloading and Global Scheduling: To handle the computational pressure from large-scale concurrent vehicles, future AVP systems will establish a well-defined Cloud-Edge-End collaborative architecture for rational offloading and allocation of planning computational resources.

9 Conclusions

This review systematically examined graph-search, sampling-based, artificial-potential-field, numerical-optimization, artificial-intelligence, and geometry-based methods for automated parking trajectory planning. The comparison shows that no single method simultaneously guarantees completeness, real-time performance, trajectory smoothness, robustness, and adaptability to dynamic environments. Graph-search and sampling methods are effective for finding feasible paths, numerical optimization improves smoothness and dynamic feasibility, artificial-intelligence methods enable rapid prediction and adaptive decision-making, and geometry-based methods offer efficient structured solutions. Future progress will therefore depend on hybrid hierarchical architectures that combine these complementary strengths with uncertainty-aware planning, verifiable safety mechanisms, dynamic-obstacle handling, and vehicle-infrastructure cooperation. Such integration is essential for translating current planning algorithms into reliable Level 4 automated valet parking systems.

For dynamic-obstacle avoidance, the six method categories form a complementary framework. Graph-search methods provide globally informed and, when extended in time, spatiotemporal route planning; sampling-based methods explore feasible maneuvers under complex geometric constraints; APF methods provide rapid local reactive guidance; numerical optimization enforces smoothness and dynamic feasibility; geometry-based interpolation improves local path smoothness; and artificial-intelligence methods can support prediction and adaptive decision making. Practical systems commonly combine these complementary functions rather than relying on a single method.

Finally, vehicle-infrastructure cooperation can support global scheduling and system-level intelligence, but its effectiveness depends heavily on low-latency, high-reliability V2X communication. This dependence is both a technological advantage and a potential safety vulnerability, making hierarchical safety redundancy essential for practical deployment.

Acknowledgement: The authors gratefully acknowledge the financial support from National Natural Science Foundation of China and Natural Science Foundation of Shanghai.

Funding Statement: This work was funded by National Natural Science Foundation of China (52572452) and Natural Science Foundation of Shanghai (25ZR1401118).

Author Contributions: The authors confirm contribution to the paper as follows: conceptualization: Xianjian Jin, Yinchun Tao, Yuhuai Zhang, Haoze Wu, Jianning Lu, Nonsly Valerienne Opinat Ikiela; supervision: Xianjian Jin; conception and design: Xianjian Jin, Yinchun Tao; collection and assembly of data: Xianjian Jin, Yinchun Tao, Yuhuai Zhang, Haoze Wu, Jianning Lu; manuscript writing: Xianjian Jin, Yinchun Tao, Yuhuai Zhang, Haoze Wu, Jianning Lu, Nonsly Valerienne Opinat Ikiela; funding: Xianjian Jin. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: All data are contained within the article.

Ethics Approval: Not applicable.

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

References

1. González D, Pérez J, Milanés V, Nashashibi F. A review of motion planning techniques for automated vehicles. *IEEE Trans Intell Transp Syst.* 2016;17(4):1135–45. doi:10.1109/TITS.2015.2498841.
2. Yamamoto K, Teng R, Sato K. Simulation evaluation of vehicle movement model using spatio-temporal grid reservation for automated valet parking. *IEEE Open J Intell Transp Syst.* 2023;4(3):261–6. doi:10.1109/OJITS.2023.3266556.
3. Li B, Fan L, Ouyang Y, Tang S, Wang X, Cao D, et al. Online competition of trajectory planning for automated parking: benchmarks, achievements, learned lessons, and future perspectives. *IEEE Trans Intell Veh.* 2023;8(1):16–21. doi:10.1109/TIV.2022.3228963.
4. Jeong Y, Kim S, Jo BR, Shin H, Yi K. Sampling based vehicle motion planning for autonomous valet parking with moving obstacles. *Int J Automot Eng.* 2018;9(4):215–22. doi:10.20485/jsaeijae.9.4_215.
5. Zhao H, Yang H, Wang Z, Xia Y. Nonlinear MPC on parallel parking for autonomous vehicles under state-dependent switching. *IEEE Trans Autom Sci Eng.* 2025;22:6377–87. doi:10.1109/TASE.2024.3443848.
6. Gorinevsky D, Kapitanovsky A, Goldenberg A. Neural network architecture for trajectory generation and control of automated car parking. *IEEE Trans Control Syst Technol.* 1996;4(1):50–6. doi:10.1109/87.481766.
7. Kim S, Kim H. Simple and complex obstacle detection using an overlapped ultrasonic sensor ring. In: 2012 12th International Conference on Control, Automation and Systems; 2012 Oct 17–21; Jeju, Republic of Korea. p. 2148–52.
8. Pohl J, Sethsson M, Degerman P, Larsson J. A semi-automated parallel parking system for passenger cars. *Proc Inst Mech Eng Part D J Automob Eng.* 2006;220(1):53–65. doi:10.1243/095440705x69650.
9. Li W, Xie Z, He P. Research on the application of multi-sensor fusion in autonomous driving. In: Proceedings of the 2025 5th International Conference on Sensors and Information Technology; 2025 Mar 21–23; Nanjing, China. p. 389–92. doi:10.1109/ICSI64877.2025.11010055.
10. Ren H, Li Y, Zhang B. Research on perceptual quantitative static evaluation model of intelligent cockpit. In: Proceedings of the 2023 International Conference on Power, Electrical Engineering, Electronics and Control (PEEEEC); 2023 Sep 25–27; Athens, Greece. p. 560–4. doi:10.1109/PEEEEC60561.2023.00115.

11. Han W, Zhou Q, Wang B, Liu B, Xiong L. Path tracking cascade control for automated valet parking of AGV systems. *IEEE Trans Veh Technol.* 2026;75(7):12363–74. doi:10.1109/TVT.2026.3658875.
12. Wang Y, Hansen E, Ahn H. Hierarchical planning for autonomous parking in dynamic environments. *IEEE Trans Control Syst Technol.* 2024;32(4):1386–98. doi:10.1109/TCST.2024.3367468.
13. Dudaklı N, Baykasoglu A. A simulation-optimization-based planning and control system for operations of fully automated parking systems. *Comput Ind Eng.* 2024;189(6):109977. doi:10.1016/j.cie.2024.109977.
14. Kim D, Chung W, Park S. Practical motion planning for car-parking control in narrow environment. *IET Control Theory Appl.* 2010;4(1):129–39. doi:10.1049/iet-cta.2008.0380.
15. Claussmann L, Revilloud M, Gruyer D, Glaser S. A review of motion planning for highway autonomous driving. *IEEE Trans Intell Transp Syst.* 2020;21(5):1826–48. doi:10.1109/TITS.2019.2913998.
16. Ozcelik MB, Agin B, Caldiran O, Sirin O. Decision making for autonomous driving in a virtual highway environment based on generative adversarial imitation learning. In: *Proceedings of the 2023 Innovations in Intelligent Systems and Applications Conference (ASYU)*; 2023 Oct 11–13; Sivas, Turkiye. p. 1–6. doi:10.1109/ASYU58738.2023.10296611.
17. Li Y, Chen P, Li H, Xu G, Wang C. Parking trajectory planning for autonomous mining trucks: a global optimal method based on divide-and-conquer strategy. *IEEE Trans Veh Technol.* 2026;75(2):2026–42. doi:10.1109/TVT.2025.3604481.
18. Teng J, Li Y, Bian Y, Hu M, Hu Y, Li G, et al. Multimodal classification network guided trajectory planning for 4WIS autonomous parking considering obstacle attributes. *IEEE Internet Things J.* 2026;13(12):26666–81. doi:10.1109/JIOT.2026.3678248.
19. Lee S, Lim W, Sunwoo M, Jo K. Limited visibility aware motion planning for autonomous valet parking using reachable set estimation. *Sensors.* 2021;21(4):1520. doi:10.3390/s21041520.
20. Cao Y, Li B, Deng Z. Optimization-based automated parking trajectory planning in unstructured environments with efficient obstacle query. *IEEE Trans Intell Transp Syst.* 2025;26(10):15422–35. doi:10.1109/TITS.2025.3605300.
21. Liu W, Deng T, Yan F. VP-YOLO: robust vehicle-pedestrian detection in challenging traffic scenarios via a human visual perception-inspired network. In: *Proceedings of the 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*; 2025 Apr 6–11; Hyderabad, India. p. 1–5. doi:10.1109/ICASSP49660.2025.10889912.
22. Zhao K, Guan D. Research on environmental perception algorithm of intelligent connected vehicles based on deep learning. In: *Proceedings of the 2024 IEEE 6th International Conference on Civil Aviation Safety and Information Technology (ICCASIT)*; 2024 Oct 23–25; Hangzhou, China. p. 1595–600. doi:10.1109/ICCASIT62299.2024.10827881.
23. Guo D, Li Z, Zhou B, Ma T, Zhang S, Gao C, et al. An automatic parking decision framework based on interactive perception prediction and multiple strategy planning. *Robot Auton Syst.* 2026;202:105520. doi:10.1016/j.robot.2026.105520.
24. Su D, Zhao Z, Zhao K, Liang K, Yu Q. Hierarchical optimal planning and real-time tracking of parking trajectories based on risk field. *Control Eng Pract.* 2025;163(2):106423. doi:10.1016/j.conengprac.2025.106423.
25. Liu P, Zhao K, Jiao X, Gao B, Dai H, Wang C. A roadside end-to-end model for rapid parking path planning over the occupancy grid map. *IET Intell Transp Syst.* 2026;20(1):e70177. doi:10.1049/itr2.70177.
26. Qin Z, Liang W, Zang Z, Chen L, Hu M, Cui Q, et al. A longitudinal and lateral coordinated control method of autonomous vehicles considering time-varying delay. *IEEE Trans Intell Veh.* 2024;9(11):7125–37. doi:10.1109/TIV.2024.3393983.
27. Yao Q, Tian Y, Wang Q, Wang S. Control strategies on path tracking for autonomous vehicle: state of the art and future challenges. *IEEE Access.* 2020;8:161211–22. doi:10.1109/ACCESS.2020.3020075.
28. Huang Y, Yong SZ, Chen Y. Stability control of autonomous ground vehicles using control-dependent barrier functions. *IEEE Trans Intell Veh.* 2021;6(4):699–710. doi:10.1109/TIV.2021.3058064.
29. Dijkstra EW. A note on two problems in connexion with graphs. *Numer Math.* 1959;1(1):269–71. doi:10.1007/BF01386390.

30. Hart PE, Nilsson NJ, Raphael B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans Syst Sci Cybern.* 1968;4(2):100–7. doi:10.1109/TSSC.1968.300136.
31. Liu T, Yang J, Wang W. A hierarchical optimization framework for parking maneuver of automated vehicle. In: *Proceedings of the 2022 6th International Conference on Automation, Control and Robots (ICACR)*; 2022 Sep 23–25; Shanghai, China. p. 156–60. doi:10.1109/ICACR55854.2022.9935542.
32. Leu J, Wang Y, Tomizuka M, Di Cairano S. Autonomous vehicle parking in dynamic environments: an integrated system with prediction and motion planning. In: *Proceedings of the 2022 International Conference on Robotics and Automation (ICRA)*; 2022 May 23–27; Philadelphia, PA, USA. p. 10890–7. doi:10.1109/ICRA46639.2022.9812309.
33. He J, Li H. Fast A* anchor point based path planning for narrow space parking. In: *Proceedings of the 2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*; 2021 Sep 19–22; Indianapolis, IN, USA. p. 1604–9. doi:10.1109/ITSC48978.2021.9564837.
34. Gan N, Zhang M, Zhou B, Chai T, Wu X, Bian Y. Spatio-temporal heuristic method: a trajectory planning for automatic parking considering obstacle behavior. *J Intell Connect Veh.* 2022;5(3):177–87. doi:10.1108/JICV-01-2022-0002.
35. Dolgov D, Thrun S, Montemerlo M, Diebel J. Path planning for autonomous driving in unknown environments. In: *Experimental robotics*. Berlin/Heidelberg, Germany: Springer; 2009. p. 55–64. doi:10.1007/978-3-642-00196-3_8.
36. Zeng D, Yu Z, Xiong L, Xia L, Zhang P, Kang R, et al. A motion planning method addressing arbitrary lots for autonomous parking vehicle. In: *Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC)*; 2019 Oct 27–30; Auckland, New Zealand. p. 1462–7. doi:10.1109/ITSC.2019.8916913.
37. Xiong L, Gao J, Fu Z, Xiao K. Path planning for automatic parking based on improved Hybrid A* algorithm. In: *Proceedings of the 2021 5th CAA International Conference on Vehicular Control and Intelligence (CVCI)*; 2021 Oct 29–31; Tianjin, China. p. 1–5. doi:10.1109/CVCI54083.2021.9661197.
38. Huang J, Liu Z, Chi X, Hong F, Su H. Search-based path planning algorithm for autonomous parking: multi-heuristic hybrid A. In: *Proceedings of the 2022 34th Chinese Control and Decision Conference (CCDC)*; 2022 Aug 15–17; Hefei, China. p. 6248–53. doi:10.1109/CCDC55256.2022.10033530.
39. Nawaz F, Sung M, Gadginmath D, D'Sa J, Bae S, Isele D, et al. Graph-based path planning with dynamic obstacle avoidance for autonomous parking. In: *Proceedings of the 2025 IEEE Intelligent Vehicles Symposium (IV)*; 2025 Jun 22–25; Cluj-Napoca, Romania. p. 702–9. doi:10.1109/IV64158.2025.11097823.
40. Tian Z, Zhang P, Feng K, Tang Y, Lu Y. Research on vertical parking path planning of semi-trailer train based on improved hybrid A* algorithm. In: *Proceedings of the 2022 2nd International Conference on Computer Science, Electronic Information Engineering and Intelligent Control Technology (CEI)*; 2022 Sep 23–25; Nanjing, China. p. 776–80. doi:10.1109/CEI57409.2022.9950219.
41. Pang J, Zhang S, Fu J, Liu J, Zheng N. Curvature continuous path planning with reverse searching for efficient and precise autonomous parking. In: *Proceedings of the 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*; 2022 Oct 8–12; Macau, China. p. 2798–805. doi:10.1109/ITSC55140.2022.9922212.
42. Li K, Lu JG, Zhang QH. A novel scenario-based path planning method for narrow parking space. In: *Proceedings of the 2023 35th Chinese Control and Decision Conference (CCDC)*; 2023 May 20–22; Yichang, China. p. 754–9. doi:10.1109/CCDC58219.2023.10327190.
43. Sheng W, Li B, Zhong X. Autonomous parking trajectory planning with tiny passages: a combination of multistage hybrid A-star algorithm and numerical optimal control. *IEEE Access.* 2021;9:102801–10. doi:10.1109/ACCESS.2021.3098676.
44. Su J, Zhang L, Xu C, He T. Guided-TargetTree-Hybrid_A* path planning algorithm for vertical parking. In: *Proceedings of the 2023 3rd International Conference on Computer Science, Electronic Information Engineering and Intelligent Control Technology (CEI)*; 2023 Dec 15–17; Wuhan, China. p. 813–6. doi:10.1109/CEI60616.2023.10527966.

45. Wang Y. Improved A-search guided tree construction for kinodynamic planning. In: International Conference on Robotics and Automation (ICRA); 2019 May 20–24; Montreal, QC, Canada. p. 5530–6. doi:10.1109/ICRA.2019.8793705.
46. Shi Y, Wang P, Wang X. An autonomous valet parking algorithm for path planning and tracking. In: Proceedings of the 2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall); 2022 Sep 26–29; London, UK. p. 1–7. doi:10.1109/VTC2022-Fall57202.2022.10012883.
47. Luna R, Şucan IA, Moll M, Kavraki LE. Anytime solution optimization for sampling-based motion planning. In: Proceedings of the 2013 IEEE International Conference on Robotics and Automation; 2013 May 6–10; Karlsruhe, Germany. p. 5068–74. doi:10.1109/ICRA.2013.6631301.
48. An B, Kim J, Park FC. An adaptive stepsize RRT planning algorithm for open-chain robots. *IEEE Robot Autom Lett.* 2018;3(1):312–9. doi:10.1109/LRA.2017.2745542.
49. Kim M, Esquerre-Pourtère A, Park J. Robust real-time sampling-based motion planner for autonomous vehicles in narrow environments. *IEEE Trans Autom Sci Eng.* 2025;22:16250–65. doi:10.1109/TASE.2025.3574262.
50. Kiss D. A continuous curvature rate path planner for autonomous cars in narrow environments. In: Proceedings of the 2025 29th International Conference on Methods and Models in Automation and Robotics (MMAR); 2025 Aug 26–29; Miedzyzdroje, Poland. p. 408–13. doi:10.1109/MMAR65820.2025.11150874.
51. Chen W, Wang Z, Wang N, Xu Y, Guo L, Zhang F, et al. Fusion of improved RRT and dynamic window algorithms for automatic parking path planning. *Proc Inst Mech Eng Part D J Automob Eng.* 2026;240(8):5486–500. doi:10.1177/09544070251366153.
52. Aydemir E, Unel M. Motion planning and path following for autonomous navigation and reversing of a full-scale mining truck and trailer system. *Int J Automot Technol.* 2025;26(3):595–606. doi:10.1007/s12239-024-00174-9.
53. Shi J, Su Y, Piao C, Tang Y, Liang Y, Wang Z. Integrated automatic parking path planning and trajectory tracking optimization method. *Trans Inst Meas Control.* 2025;47(14):2998–3012. doi:10.1177/01423312241282832.
54. Zhao M, Shen T, Wang F, Yin G, Li Z, Zhang Y. APTEN-planner: autonomous parking of semi-trailer train in extremely narrow environments. *IEEE Trans Intell Transp Syst.* 2024;25(5):4116–32. doi:10.1109/TITS.2023.3328245.
55. Fuji H, Xiang J, Tazaki Y, Levedahl B, Suzuki T. Trajectory planning for automated parking using multi-resolution state roadmap considering non-holonomic constraints. In: Proceedings of the 2014 IEEE Intelligent Vehicles Symposium Proceedings; 2014 Jun 8–11; Dearborn, MI, USA. p. 407–13. doi:10.1109/IVS.2014.6856433.
56. Tazaki Y, Okuda H, Suzuki T. Parking trajectory planning using multiresolution state roadmaps. *IEEE Trans Intell Veh.* 2017;2(4):298–307. doi:10.1109/TIV.2017.2769882.
57. Alpkiray N, Torun Y, Kaynar O. Probabilistic roadmap and artificial bee colony algorithm cooperation for path planning. In: Proceedings of the 2018 International Conference on Artificial Intelligence and Data Processing (IDAP); 2018 Sep 28–30; Malatya, Turkey. p. 1–6. doi:10.1109/IDAP.2018.8620808.
58. Han Z, Chen P, Zhou B, Yu G. Dual-layer path planning for unmanned ground vehicles based on probabilistic roadmap and proximal policy optimization. In: Proceedings of the 2024 IEEE 22nd International Conference on Industrial Informatics (INDIN); 2024 Aug 18–20; Beijing, China. p. 1–6. doi:10.1109/INDIN58382.2024.10774269.
59. Padmaraja VP, Ranjith R. MAGV navigation in warehouse environments using probabilistic roadmaps. In: Proceedings of the 2025 4th International Conference on Innovative Mechanisms for Industry Applications (ICIMIA); 2025 Sep 3–5; Tirupur, India. p. 584–9. doi:10.1109/icimia67127.2025.11200961.
60. Kim M, Ahn J, Park J. TargetTree-RRT*: continuous-curvature path planning algorithm for autonomous parking in complex environments. *IEEE Trans Autom Sci Eng.* 2024;21(1):606–17. doi:10.1109/TASE.2022.3225821.
61. Yang J, Wang J, Li J, Meng X, Jiang X, Lu C. Sobol sequence RRT* and numerical optimal joint algorithm-based automatic parking trajectory planning of four-wheel steering vehicles. *Robot Auton Syst.* 2025;186(8):104909. doi:10.1016/j.robot.2024.104909.
62. Dong Y, Zhong Y, Hong J. Knowledge-biased sampling-based path planning for automated vehicles parking. *IEEE Access.* 2020;8:156818–27. doi:10.1109/ACCESS.2020.3018731.

63. Jiang C, Hu Z, Mourelatos ZP, Gorsich D, Jayakumar P, Fu Y, et al. R2-RRT*: reliability-based robust mission planning of off-road autonomous ground vehicle under uncertain terrain environment. *IEEE Trans Autom Sci Eng.* 2022;19(2):1030–46. doi:10.1109/TASE.2021.3050762.
64. Yin J, Hu Z, Mourelatos ZP, Gorsich D, Singh A, Tau S. Efficient reliability-based path planning of off-road autonomous ground vehicles through the coupling of surrogate modeling and RRT. *IEEE Trans Intell Transp Syst.* 2023;24(12):15035–50. doi:10.1109/TITS.2023.3296651.
65. Schörner P, Conzelmann M, Fleck T, Zofka M, Zöllner JM. Park my car! automated valet parking with different vehicle automation levels by V2X connected smart infrastructure. In: *Proceedings of the 2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*; 2021 Sep 19–22; Indianapolis, IN, USA. p. 836–43. doi:10.1109/ITSC48978.2021.9565095.
66. Lattarulo R, Pérez J, Murgotio J. RRT trajectory planning approach for automated semi-trailer truck parking. In: *Proceedings of the 2022 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*; 2022 Nov 14–16; Bogota, Colombia. p. 1–7. doi:10.1109/ICVES56941.2022.9986832.
67. Solmaz S, Muminovic R, Civgin A, Stettinger G. Development, analysis, and real-life benchmarking of RRT-based path planning algorithms for automated valet parking. In: *Proceedings of the 2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*; 2021 Sep 19–22; Indianapolis, IN, USA. p. 621–8. doi:10.1109/itsc48978.2021.9564413.
68. Manav AC, Lazoglu I. A novel cascade path planning algorithm for autonomous truck-trailer parking. *IEEE Trans Intell Transp Syst.* 2022;23(7):6821–35. doi:10.1109/TITS.2021.3062701.
69. Ma H, Meng F, Ye C, Wang J, Meng MQH. Bi-risk-RRT based efficient motion planning for autonomous ground vehicles. *IEEE Trans Intell Veh.* 2022;7(3):722–33. doi:10.1109/TIV.2022.3152740.
70. Chen H, Yang J, Ye X, Chen W, Guo W, Wang Y, et al. A hierarchical RRT-DWA planner for autonomous parking in dynamic environments. *Bull Pol Acad Sci Tech Sci.* 2026;158296. doi:10.24425/bpasts.2026.158296.
71. Lee S, Lim W, Sunwoo M. Robust parking path planning with error-adaptive sampling under perception uncertainty. *Sensors.* 2020;20(12):3560. doi:10.3390/s20123560.
72. Mudiyansele MW, Aghdam FH, Kazemi-Razi SM, Chaudhari K, Marzband M, Ikpehai A, et al. A multiagent framework for electric vehicles charging power forecast and smart planning of urban parking lots. *IEEE Trans Transp Electrif.* 2024;10(2):2844–57. doi:10.1109/TTE.2023.3289196.
73. Wang X, Shi H, Zhang C. Path planning for intelligent parking system based on improved ant colony optimization. *IEEE Access.* 2020;8:65267–73. doi:10.1109/ACCESS.2020.2984802.
74. Canale M, Cerrito F, Borodani P. An ego-based approach to planning and control for automated valet parking applications. In: *Proceedings of the 2024 IEEE 63rd Conference on Decision and Control (CDC)*; 2024 Dec 16–19; Milan, Italy. p. 8193–8. doi:10.1109/CDC56724.2024.10886152.
75. Zhang H, Yu Y, Liu Y, Jiang F, Chen Z, Zhang Y. Efficient path planning and tracking control of autonomous vehicles in park scenarios based on sparrow search and stepwise prediction. *IEEE Trans Transp Electrif.* 2024;10(4):8346–61. doi:10.1109/TTE.2024.3361853.
76. Zhao J, Zhang Z, Xue Z, Li L. A hierarchical vehicle motion planning method for cruise in parking area. In: *Proceedings of the 2021 5th CAA International Conference on Vehicular Control and Intelligence (CVCI)*; 2021 Oct 29–31; Tianjin, China. p. 1–6. doi:10.1109/CVCI54083.2021.9661211.
77. Khatib O. Real-time obstacle avoidance for manipulators and mobile robots. *Int J Robot Res.* 1986;5(1):90–8. doi:10.1177/027836498600500106.
78. Chiang HT, Malone N, Lesser K, Oishi M, Tapia L. Path-guided artificial potential fields with stochastic reachable sets for motion planning in highly dynamic environments. In: *Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA)*; 2015 May 26–30; Seattle, WA, USA. p. 2347–54. doi:10.1109/ICRA.2015.7139511.
79. Dolgov D, Thrun S, Montemerlo M, Diebel J. Path planning for autonomous vehicles in unknown semi-structured environments. *Int J Robot Res.* 2010;29(5):485–501. doi:10.1177/0278364909359210.

80. Martin A, Lattarulo R, Zubizarreta A, Perez J, Lopez-Garcia P. Trajectory planning for automated buses in parking areas. In: *Proceedings of the 2021 25th International Conference on System Theory, Control and Computing (ICSTCC)*; 2021 Oct 20–23; Iasi, Romania. p. 688–94. doi:10.1109/icstcc52150.2021.9607247.
81. Azevedo J, D'orey PM, Ferreira M. High-density parking for automated vehicles: a complete evaluation of coordination mechanisms. *IEEE Access*. 2020;8:43944–55. doi:10.1109/ACCESS.2020.2973494.
82. Liu K, Du Q, Zhan W, Dong H, Zhang Y, Gong Y. Intelligent vehicle park path planning based on improved artificial potential field method. *Int J Automot Technol*. 2026. doi:10.1007/s12239-026-00424-y.
83. Shangguan L, Thomasson JA, Gopalswamy S. Motion planning for autonomous grain carts. *IEEE Trans Veh Technol*. 2021;70(3):2112–23. doi:10.1109/TVT.2021.3058274.
84. Tao F, Ding Z, Fu Z, Li M, Ji B. Efficient path planning for autonomous vehicles based on RRT* with variable probability strategy and artificial potential field approach. *Sci Rep*. 2024;14(1):24698. doi:10.1038/s41598-024-76299-9.
85. Li B, Yin Z, Ouyang Y, Zhang Y, Zhong X, Tang S. Online trajectory replanning for sudden environmental changes during automated parking: a parallel stitching method. *IEEE Trans Intell Veh*. 2022;7(3):748–57. doi:10.1109/TIV.2022.3156429.
86. Wolf MT, Burdick JW. Artificial potential functions for highway driving with collision avoidance. In: *Proceedings of the 2008 IEEE International Conference on Robotics and Automation*; 2008 May 19–23; Pasadena, CA, USA. p. 3731–6. doi:10.1109/ROBOT.2008.4543783.
87. Rasekhipour Y, Khajepour A, Chen SK, Litkouhi B. A potential field-based model predictive path-planning controller for autonomous road vehicles. *IEEE Trans Intell Transp Syst*. 2017;18(5):1255–67. doi:10.1109/TITS.2016.2604240.
88. Dong Y, Zhang Y, Ai J. Experimental test of artificial potential field-based automobiles automated perpendicular parking. *Int J Veh Technol*. 2016;2016(1):2306818. doi:10.1155/2016/2306818.
89. Kim D, Huh K. Neural motion planning for autonomous parking. *Int J Control Autom Syst*. 2023;21(4):1309–18. doi:10.1007/s12555-022-0082-z.
90. Chen C, Wu B, Xuan L, Chen J, Wang T, Qian L. A trajectory planning method for autonomous valet parking via solving an optimal control problem. *Sensors*. 2020;20(22):6435. doi:10.3390/s20226435.
91. Kondak K, Hommel G. Computation of time optimal movements for autonomous parking of non-holonomic mobile platforms. In: *Proceedings 2001 ICRA IEEE International Conference on Robotics and Automation*; 2001 May 21–26; Seoul, Republic of Korea. p. 2698–703. doi:10.1109/ROBOT.2001.933030.
92. Li B, Wang K, Shao Z. Time-optimal maneuver planning in automatic parallel parking using a simultaneous dynamic optimization approach. *IEEE Trans Intell Transp Syst*. 2016;17(11):3263–74. doi:10.1109/TITS.2016.2546386.
93. Park G, Kim S, Kang H. Optimal driving control for autonomous electric vehicles based on in-wheel motors using an artificial potential field. *IEEE Access*. 2024;12:113799–809. doi:10.1109/ACCESS.2024.3443869.
94. Li B, Acarman T, Peng X, Zhang Y, Bian X, Kong Q. Maneuver planning for automatic parking with safe travel corridors: a numerical optimal control approach. In: *Proceedings of the 2020 European Control Conference (ECC)*; 2020 May 12–15; St. Petersburg, Russia. IEEE; 2020. p. 1993–8.
95. Om Aditya MVK, Sujatha CN, Adithya J, Kiran BSRPS. Automated valet parking using double deep Q learning. In: *Proceedings of the 2023 International Conference on Advances in Electronics, Communication, Computing and Intelligent Information Systems (ICAECIS)*; 2023 Apr 19–21; Bangalore, India. p. 259–64. doi:10.1109/ICAECIS58353.2023.10170029.
96. Cai L, Guan H, Zhou ZY, Xu FL, Jia X, Zhan J. Parking planning under limited parking corridor space. *IEEE Trans Intell Transp Syst*. 2023;24(2):1962–81. doi:10.1109/TITS.2022.3219651.
97. Liu T, Chai R, Chai S, Xia Y. Chance-constrained trajectory optimization for automatic parking based on conservative approximation. *IEEE Trans Veh Technol*. 2024;73(5):6259–69. doi:10.1109/TVT.2023.3342421.
98. Kim DJ, Jeong YW, Chung CC. Lateral vehicle trajectory planning using a model predictive control scheme for an automated perpendicular parking system. *IEEE Trans Ind Electron*. 2023;70(2):1820–9. doi:10.1109/TIE.2022.3163567.

99. Qiu D, Qiu D, Wu B, Gu M, Zhu M. Hierarchical control of trajectory planning and trajectory tracking for autonomous parallel parking. *IEEE Access*. 2021;9:94845–61. doi:10.1109/ACCESS.2021.3093930.
100. Song S, Chen H, Sun H, Liu M, Xia T. Time-optimized online planning for parallel parking with nonlinear optimization and improved Monte Carlo tree search. *IEEE Robot Autom Lett*. 2022;7(2):2226–33. doi:10.1109/LRA.2021.3139950.
101. Hu Q, Zhan G, Gao F, Liu H. A low-dimensional optimization method for parallel parking path planning. In: *Proceedings of the 2025 International Conference on Electrical Automation and Artificial Intelligence (ICEAAI)*; 2025 Jan 10–12; Guangzhou, China. p. 1160–5. doi:10.1109/ICEAAI64185.2025.10956230.
102. Li B, Acarman T, Zhang Y, Ouyang Y, Yaman C, Kong Q, et al. Optimization-based trajectory planning for autonomous parking with irregularly placed obstacles: a lightweight iterative framework. *IEEE Trans Intell Transp Syst*. 2022;23(8):11970–81. doi:10.1109/TITS.2021.3109011.
103. Gao H, Liu Y, Zhang X, Chen G, Li D. Path planning algorithm regarding rapid parking based on static optimization. In: *Proceedings of the 2014 IEEE 3rd International Conference on Cloud Computing and Intelligence Systems*; 2014 Nov 27–29; Shenzhen, China. p. 672–8. doi:10.1109/CCIS.2014.7175819.
104. Liu P, Chen Z, Liu M, Piao C, Wan K, Huang H. Vertical parking trajectory planning with the combination of numerical optimization method and gradient lifting decision tree. *IEEE Trans Consum Electron*. 2024;70(1):1845–56. doi:10.1109/TCE.2023.3321109.
105. He W, Chen Y, Liu T, Ren F, Wan K. Gaussian pseudo-spectrum optimization-based fuzzy logic parallel parking trajectory planning. *Int J Automot Technol*. 2025;26(1):115–28. doi:10.1007/s12239-024-00127-2.
106. Zhang X, Liniger A, Borrelli F. Optimization-based collision avoidance. *IEEE Trans Control Syst Technol*. 2021;29(3):972–83. doi:10.1109/TCST.2019.2949540.
107. Zhang Z, Lu S, Xie L, Su H, Li D, Wang Q, et al. A guaranteed collision-free trajectory planning method for autonomous parking. *IET Intell Transp Syst*. 2021;15(2):331–43. doi:10.1049/itr2.12028.
108. Micelli P, Consolini L, Locatelli M. Path planning with limited numbers of maneuvers for automatic guided vehicles: an optimization-based approach. In: *Proceedings of the 2017 25th Mediterranean Conference on Control and Automation (MED)*; 2017 Jul 3–6; Valletta, Malta. p. 204–9. doi:10.1109/MED.2017.7984119.
109. Wang J, Li J, Yang J, Meng X, Fu T. Automatic parking trajectory planning based on random sampling and nonlinear optimization. *J Frankl Inst*. 2023;360(13):9579–601. doi:10.1016/j.jfranklin.2023.06.037.
110. Lian J, Ren W, Yang D, Li L, Yu F. Trajectory planning for autonomous valet parking in narrow environments with enhanced hybrid A* search and nonlinear optimization. *IEEE Trans Intell Veh*. 2023;8(6):3723–34. doi:10.1109/TIV.2023.3268088.
111. Zhang X, Liniger A, Sakai A, Borrelli F. Autonomous parking using optimization-based collision avoidance. In: *Proceedings of the 2018 IEEE Conference on Decision and Control (CDC)*; 2018 Dec 17–19; Miami, FL, USA. p. 4327–32. doi:10.1109/CDC.2018.8619433.
112. Zhang G, Chai S, Chai R, Garcia M, Xia Y. Fuzzy goal programming algorithm for multi-objective trajectory optimal parking of autonomous vehicles. *IEEE Trans Intell Veh*. 2024;9(1):1909–18. doi:10.1109/TIV.2023.3311536.
113. Chen G, Hou J, Dong J, Li Z, Gu S, Zhang B, et al. Multiobjective scheduling strategy with genetic algorithm and time-enhanced A* planning for autonomous parking robotics in high-density unmanned parking lots. *IEEE/ASME Trans Mechatron*. 2021;26(3):1547–57. doi:10.1109/TMECH.2020.3023261.
114. Li H, Chen P, Yu G, Zhou B, Li Y, Liao Y. Trajectory planning for autonomous driving in unstructured scenarios based on deep learning and quadratic optimization. *IEEE Trans Veh Technol*. 2024;73(4):4886–903. doi:10.1109/TVT.2023.3330581.
115. Zhang J, Chen H, Song S, Hu F. Reinforcement learning-based motion planning for automatic parking system. *IEEE Access*. 2020;8:154485–501. doi:10.1109/ACCESS.2020.3017770.
116. Song S, Chen H, Sun H, Liu M. Data efficient reinforcement learning for integrated lateral planning and control in automated parking system. *Sensors*. 2020;20(24):7297. doi:10.3390/s20247297.
117. Yuan Z, Wang Z, Li X, Li L, Zhang L. Hierarchical trajectory planning for narrow-space automated parking with deep reinforcement learning: a federated learning scheme. *Sensors*. 2023;23(8):4087. doi:10.3390/s23084087.

118. Shi J, Li K, Piao C, Gao J, Chen L. Model-based predictive control and reinforcement learning for planning vehicle-parking trajectories for vertical parking spaces. *Sensors*. 2023;23(16):7124. doi:10.3390/s23167124.
119. Du Z, Miao Q, Zong C. Trajectory planning for automated parking systems using deep reinforcement learning. *Int J Automot Technol*. 2020;21(4):881–7. doi:10.1007/s12239-020-0085-9.
120. Gamal O, Imran M, Roth H, Wahrburg J. Assistive parking systems knowledge transfer to end-to-end deep learning for autonomous parking. In: *Proceedings of the 2020 6th International Conference on Mechatronics and Robotics Engineering (ICMRE)*; 2020 Feb 12–15; Barcelona, Spain. p. 216–21. doi:10.1109/ICMRE49073.2020.9065014.
121. Tang X, Yang Y, Liu T, Lin X, Yang K, Li S. Path planning and tracking control for parking via soft actor-critic under non-ideal scenarios. *IEEE/CAA J Autom Sin*. 2024;11(1):181–95. doi:10.1109/JAS.2023.123975.
122. Moon J, Bae I, Kim S. Automatic parking controller with a twin artificial neural network architecture. *Math Probl Eng*. 2019;2019(1):1–18. doi:10.1155/2019/4801985.
123. Chai R, Liu D, Liu T, Tsourdos A, Xia Y, Chai S. Deep learning-based trajectory planning and control for autonomous ground vehicle parking maneuver. *IEEE Trans Autom Sci Eng*. 2023;20(3):1633–47. doi:10.1109/TASE.2022.3183610.
124. Li B, Zhang YM, Shao ZJ. Motion planning methodologies for automated vehicles: a critical review. *Control Inf Technol*. 2018;(6):1–6. (In Chinese). doi:10.13889/j.issn.2096-5427.2018.06.100.
125. Dubins LE. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *Am J Math*. 1957;79(3):497. doi:10.2307/2372560.
126. Reeds JA, Shepp LA. Optimal paths for a car that goes both forwards and backwards. *Pac J Math*. 1990;145(2):367–93. doi:10.2140/pjm.1990.145.367.
127. Jhang JH, Lian FL, Hao YH. Forward and backward motion planning for autonomous parking using smooth-feedback bidirectional rapidly-exploring random trees with pattern cost penalty. In: *Proceedings of the 2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*; 2020 Aug 20–21; Hong Kong, China. p. 260–5. doi:10.1109/CASE48305.2020.9216968.
128. Hong HJ, Gwang Choi Y, Kim EH, Hyeon Park H, Jeon JW. Reeds-shepp path planning with a heuristic cost function for autonomous parking. In: *Proceedings of the 2025 IEEE/IEIE International Conference on Consumer Electronics-Asia (ICCE-Asia)*; 2025 Oct 27–29; Busan, Republic of Korea. p. 1–6. doi:10.1109/icce-asia67487.2025.11263680.
129. Yin A, Yan YJ, Wang HL, Tao SL, Wang Y, Pi DW. Research on automatic parking motion control based on a variable radius reeds-shepp method. In: *Proceedings of the 2024 8th CAA International Conference on Vehicular Control and Intelligence (CVCI)*; 2024 Oct 25–27; Chongqing, China. p. 1–6. doi:10.1109/CVCI63518.2024.10830040.
130. Zou R, Wang S, Wang Z, Zhao P, Zhou P. A reverse planning method of autonomous parking path. In: *Proceedings of the 2020 5th Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)*; 2020 Jul 17–19; Singapore. p. 92–8. doi:10.1109/ACIRS49895.2020.9162616.
131. Daniali SM, Khosravi A, Sarhadi P, Tavakkoli F. An automatic parking algorithm design using multi-objective particle swarm optimization. *IEEE Access*. 2023;11:49611–24. doi:10.1109/ACCESS.2023.3276858.
132. Li S, Wang J. Parallel parking path planning in narrow space based on a three-stage curve interpolation method. *IEEE Access*. 2023;11:93841–51. doi:10.1109/ACCESS.2023.3310256.
133. Chen X. Automatic vertical parking path planning based on clothoid curve and Stanley algorithm. In: *Proceedings of the 2022 IEEE 5th International Conference on Information Systems and Computer Aided Education (ICISCAE)*; 2022 Sep 23–25; Dalian, China. p. 761–6. doi:10.1109/ICISCAE55891.2022.9927631.
134. Arasawa K, Hoshi Y, Oya H. A path planning method for automated parking systems based on circular arcs and lines with smoothly curves. In: *Proceedings of the 2024 IEEE 3rd Industrial Electronics Society Annual On-Line Conference (ONCON)*; 2024 Dec 8–10; Beijing, China. p. 1–7. doi:10.1109/ONCON62778.2024.10931450.
135. Ding N, Cao L, Duan C, Liao J. Geometric path plans for perpendicular parking based on clothoid curve. In: *Proceedings of the 2023 2nd International Conference on Sensing, Measurement, Communication and Internet of Things Technologies (SMC-IoT)*; 2023 Dec 29–31; Changsha, China. p. 161–6. doi:10.1109/SMC-IoT62253.2023.00036.

136. Vieira R, Argento E, Revoredo T. Trajectory planning for car-like robots through curve parametrization and genetic algorithm optimization with applications to autonomous parking. *IEEE Lat Am Trans.* 2022;20(2):309–16. doi:10.1109/TLA.2022.9661471.
137. Duong MT, Au DT, Trinh HA, Le MH. Automatic parking control algorithm for four-wheeled vehicles with ackerman steering mechanisms. *IEEE Access.* 2025;13(3):194382–400. doi:10.1109/ACCESS.2025.3631858.
138. Yan G, Wang H, Zhang C, Wang J. iPPSys: a parallel parking system for path planning and tracking based on multi-obstacle analysis and model predictive control. In: *Proceedings of the 2025 1st International Symposium on E-CARGO and Applications (E-CARGO)*; 2025 Jul 21–23; Guangzhou, China. p. 68–73. doi:10.1109/e-cargo65996.2025.11139336.
139. Guo H. Intelligent vehicle perpendicular parking path planning and tracking control. In: *Proceedings of the 2022 International Symposium on Intelligent Robotics and Systems (ISoIRS)*; 2022 Oct 14–16; Chengdu, China. p. 86–96. doi:10.1109/ISoIRS57349.2022.00026.
140. Tang W, Yang M, Le F, Yuan W, Wang B, Wang C. Micro-vehicle-based automatic parking path planning. In: *Proceedings of the 2018 IEEE International Conference on Real-time Computing and Robotics (RCAR)*; 2018 Aug 1–5; Kandima, Maldives. p. 160–5. doi:10.1109/RCAR.2018.8621775.
141. Hossain MI, Alam MU, Rahat MK, Rahman MA, Shufian A, Zishan MSR. Autonomous parking valet system: path planning and control in complex environments. In: *Proceedings of the 2025 2nd International Conference on Advanced Innovations in Smart Cities (ICAISC)*; 2025 Feb 9–11; Jeddah, Saudi Arabia. p. 1–6. doi:10.1109/ICAISC64594.2025.10959602.
142. Li B, Huang J, Zhang Y, Chen H. A hybrid strategy for parallel parking by combining geometric and optimization planners. In: *Proceedings of the 2024 4th International Symposium on Artificial Intelligence and Intelligent Manufacturing (AIIM)*; 2024 Dec 20–22; Chengdu, China. p. 1036–40. doi:10.1109/AIIM64537.2024.10934415.
143. Lv X, Jin W, Qiu X, Mo F, Fang M, Li J, et al. Final parking state optimization and two-stage path planning in dynamic parking lot considering comfort of passenger boarding and alighting. *IEEE Trans Intell Transp Syst.* 2025;26(10):15486–501. doi:10.1109/TITS.2025.3602157.
144. Liang Z, Zheng G, Li J. Automatic parking path optimization based on Bezier curve fitting. In: *Proceedings of the 2012 IEEE International Conference on Automation and Logistics*; 2012 Aug 15–17; Zhengzhou, China. p. 583–7. doi:10.1109/ICAL.2012.6308145.
145. Song J, Zhang W, Wu X, Gao Q. Automatic perpendicular parking trajectory planning and following for vehicle. In: *Proceedings of the 2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE)*; 2019 Oct 18–20; Xiamen, China. p. 932–5. doi:10.1109/EITCE47263.2019.9095133.
146. Wang Z, Yang P, Guo G, Wei Y, Han M. Trajectory planning in narrow environments by integrating enhanced target-tree and hybrid A* for automated parking systems. *IEEE Trans Intell Transp Syst.* 2026;27(3):3309–24. doi:10.1109/TITS.2025.3642158.
147. Li H, Chen Z, Huang Z, Gao Z. Research on automatic parking path planning based on bidirectional hybrid A-star algorithm. In: *Proceedings of the 2024 11th International Forum on Electrical Engineering and Automation (IFEEA)*; 2024 Nov 22–24; Shenzhen, China. p. 1251–4. doi:10.1109/IFEEA64237.2024.10878513.
148. Zhang C, Wei J, Qu S, Huang C, Dai J, Fu P, et al. Implementation of a V2P-based VRU warning system with C-V2X technology. *IEEE Access.* 2023;11:69903–15. doi:10.1109/ACCESS.2023.3293122.
149. Naik G, Choudhury B, Park JM. IEEE 802.11bd & 5G NR V2X: evolution of radio access technologies for V2X communications. *IEEE Access.* 2019;7:70169–84. doi:10.1109/ACCESS.2019.2919489.
150. Rehman MA, Numan M, Tahir H, Rahman U, Khan MW, Iftikhar MZ. A comprehensive overview of vehicle to everything (V2X) technology for sustainable EV adoption. *J Energy Storage.* 2023;74(3):109304. doi:10.1016/j.est.2023.109304.
151. Oh S, Chen Q, Tseng HE, Pandey G, Orosz G. Sharable clothoid-based continuous motion planning for connected automated vehicles. *IEEE Trans Control Syst Technol.* 2025;33(4):1372–86. doi:10.1109/TCST.2024.3448328.