Journal of Carcinogenesis
Covering Genesis to Therapeutics

Journal of Carcinogenesis, ISSN: 1477-3163
2025; Vol 24: Issue 3, https://carcinogenesis.com/

A Robust Δ-Safe Hybrid Edge–Vertex–Cell Trajectory Planning Algorithm for Autonomous Navigation in Cancer Research Facilities: Integrating Clearance Preservation, Computational Efficiency, and Geometric Smoothness

Mr. Amey Pushkaraj Kore[1], Dr. Vijaya N. Aher[2], Dr. Sandeep V. Gaikwad[3], Dr. Rahul S. Pol[4], Dr. Pallavi D. Deshpande[5], Dr. Ketki P. Kshirsagar[6], Dr. Anup W. Ingle[7]

1Department of Electronics and Telecommunication, VIIT, Pune, India, NEU and College of Engineering, kore.a@northeastern.edu

2Assistant Professor, Department of Instrumentation Engineering, VIT, Pune, India, vijaya.aher@vit.edu

3Associate Professor, Department of Electronics & Telecommunication Engineering, SCTR's Pune Institute of Computer Technology (PICT), Pune svgaikwad@pict.edu

4Associate Professor, Department of Electronics and Telecommunication, VIT, Pune, India, rahul.pol@vit.edu

5Assistant Professor, Department of Electronics and Telecommunication, VIT, Pune, India, pallavi.deshpande@vit.edu

6Associate Professor, Department of Electronics and Telecommunication, VIT, Pune, India, ketki.kshirsagar@vit.edu

7Assistant Professor, Department of Electronics and Telecommunication, VIT, Pune, India, anup.ingle@vit.edu

ABSTRACT

Carcinogenesis research increasingly relies on advanced technologies, including robotic systems, to enhance experimental precision, reduce human error, and streamline laboratory workflows in controlled environments. Against this backdrop, the deployment of autonomous mobile robots within indoor research facilities demands navigation algorithms that adeptly balance path optimality, computational efficiency, and safety. Traditional path planning methods such as A* and Theta* often generate unnecessarily extended trajectories or compromise obstacle clearance, limiting their applicability in sensitive environments. To address these challenges, this study introduces the Delta-Safe (Δ-Safe) Hybrid Edge–Vertex–Cell Optimal Trajectory Planning Algorithm (HEVC-OTPA), a novel approach that discretises the workspace into uniform grids and integrates hybrid edge, vertex, and cell expansions to ensure robust exploration. Safety is reinforced through a clearance penalty function and Δ-offset termination rules, while redundant nodes are pruned using line-of-sight verification. Cubic Bézier smoothing is applied to enhance geometric feasibility and motion fluidity. Comparative simulations against A* and Theta* across diverse obstacle densities demonstrate that HEVC-OTPA yields path lengths 6–14% shorter, maintains a uniform obstacle clearance of 1.0 m, reduces node expansions by 20–28%, and achieves runtime reductions of 24–28%. These results confirm the algorithm's ability to deliver improved navigation performance without compromising safety, thereby facilitating faster convergence in complex indoor layouts. By ensuring both computational efficiency and reliable obstacle avoidance, HEVC-OTPA holds significant potential for integration into laboratory automation, clinical material handling, and research facility logistics—key areas where precision, safety, and efficiency are paramount in carcinogenesis-related experimental protocols.

Keywords: *Optimal Trajectory Planning Algorithm (OTPA); Indoor Mobile Robot Navigation; Hybrid Edge–Vertex–Cell Expansion; Δ-Safe Path Planning; Computational Efficiency in Robotics; Collision-Free Autonomous Navigation*

How to Cite: Mr. Amey Pushkaraj Kore[1], Dr. Vijaya N. Aher[2], Dr. Sandeep V. Gaikwad[3], Dr. Rahul S. Pol[4], Dr. Pallavi D. Deshpande[5], Dr. Ketki P. Kshirsagar[6], Dr. Anup W. Ingle[7], (2025) A Robust Δ-Safe Hybrid Edge–Vertex–Cell Trajectory Planning Algorithm for Autonomous Navigation in Cancer Research Facilities: Integrating Clearance Preservation, Computational Efficiency, and Geometric Smoothness, *Journal of Carcinogenesis*, *Vol.24*, *No.3*, 415-429.

1. INTRODUCTION

The increasing reliance on autonomous mobile robots in indoor environments has created a pressing demand for reliable navigation strategies that can balance path efficiency, computational cost, and operational safety. Classical algorithms such

as A* and its variants have been widely used due to their deterministic nature but often produce unnecessarily long and suboptimal routes. Conversely, Theta* and other any-angle planners shorten trajectories but compromise clearance, making them unsuitable for environments where collision risks must be strictly managed. Modern robotic applications, ranging from warehouse logistics to hospital service robots, require planners that not only compute fast but also adapt to dynamic layouts with guaranteed safety margins. Existing methods face limitations when navigating densely cluttered spaces, particularly in ensuring obstacle-free paths without inflating computational complexity. To address these gaps, this study introduces a Δ-Safe Hybrid Edge–Vertex–Cell Optimal Trajectory Planning Algorithm (HEVC-OTPA) that combines the structural rigor of grid-based search with the flexibility of online smoothing. By integrating clearance-aware termination rules, pruning redundant nodes, and employing Bézier-based trajectory refinement, the proposed method ensures both optimality and feasibility. Through extensive comparative analysis with A* and Theta*, OTPA demonstrates measurable improvements in path length, runtime, and safety clearance, establishing its potential for real-world robotic deployment.

Path planning in robotics has historically been categorized into grid-based, graph-based, and sampling-based frameworks, each with characteristic strengths and limitations. Classical search strategies such as A* and its successors—Theta* [23], Field D* [22], and Anya [24]—have demonstrated varying trade-offs between geometric optimality, smoothness, and computational overhead. A* guarantees completeness but expands grid nodes excessively, Theta* reduces path length by exploiting line-of-sight connections yet risks unsafe proximity to obstacles, while Field D* interpolates smoother trajectories but incurs higher runtime. Anya provides provably optimal any-angle solutions but sacrifices efficiency in dense maps. Comparative analyses [22, 23, 24] have established that no single algorithm uniformly dominates; instead, hybrid formulations that integrate global heuristics with local feasibility checks are emerging as more practical solutions for indoor robotics.

Several authors have addressed the inherent rigidity of A* by combining it with local dynamic planners. Mi and Jin [16] and Jiang et al. [6] independently developed hybrid A*–DWA models where a global search trajectory is refined through velocity-constrained maneuvers. Their experiments consistently reported smoother navigation, reduced collision incidences, and real-time executability, with improvements of 11% in path length and 22–25% in angular smoothness relative to standalone A*. Similarly, Tola and Gashaw [13] optimized A* through heuristic scaling and adaptive neighbor pruning, achieving 7–12% shorter paths and 20% greater obstacle clearance, while Gareau and Smith [7] extended the search paradigm into coverage tasks using branch-and-bound heuristics, solving large maps an order of magnitude faster than brute-force baselines. Collectively, these studies show that augmenting A* with pruning, smoothing, or hybrid velocity control greatly improves its feasibility in cluttered layouts.

Sampling-based approaches, particularly RRT* and its variants, have been extensively refined to improve convergence and passage navigation. Wang et al. [2] advanced an adaptive biased RRT* with Voronoi-guided sampling, reducing computation time by 20–40% and iterations by up to 45%, while Zhou and Wang [17] as well as Zhou, Wang, and Guo [5] introduced Gridized Variable Probability RRT (GVP-RRT) methods that boosted success in narrow passages from ~78% to 96% and cut redundant node exploration by nearly 25%. Other refinements include PQ-RRT* [27] with priority-based sampling that reduced planning time by 28%, and Quick-RRT* [28] exploiting triangular inequality to accelerate pruning, yielding 35–40% faster execution. Foundational work by Karaman and Frazzoli [29] proved the asymptotic optimality of RRT*, while Qureshi and Ayaz [30] integrated potential fields for guided exploration, reducing failures in bottlenecks by 14%. These studies collectively establish that intelligent sampling bias and pruning heuristics dramatically improve both efficiency and reliability in constrained indoor navigation.

Beyond deterministic and sampling paradigms, influence-based and optimization-driven strategies have been explored for dynamic obstacle navigation. Stawarz et al. [4, 18] developed supported influence mapping, where weighted potentials of static and mobile obstacles guide trajectory generation. Their real-world trials achieved 94% collision-free navigation, outperforming A* by ~15%, while maintaining acceptable computation overheads. In parallel, metaheuristic optimization has gained prominence: reviews by Promkaew and Sahakij [10] and Liu et al. [21] highlight that genetic algorithms, particle swarm optimization (PSO), and hybrid PSO–DWA frameworks [25] reduce path costs by 5–25% and collision rates below 2%, albeit at the expense of convergence speed. The consensus is that hybrid metaheuristic–reactive designs yield the most practical balance between global optimality and real-time responsiveness.

The integration of learning into path planning has introduced adaptability beyond classical heuristics. Rani et al. [14] applied reinforcement learning (RL) to grid navigation, reducing collision incidents by 35% and achieving >90% success rates on unseen maps, though at the cost of long training. Zhou and Wei [26] extended this paradigm using deep reinforcement learning for AGVs in dense warehouse layouts, yielding 17% shorter paths and 21% faster navigation compared to A* and DWA baselines. Complementary to learning-based control, AbuJabal et al. [3, 19] surveyed dynamic obstacle navigation techniques, emphasizing predictive models and hybrid RL-reactive systems that sustain >90% success in dynamic layouts. These findings highlight RL's potential to supersede handcrafted heuristics, provided computational resources for training are available.

Some researchers have expanded the planning objective beyond mere geometric optimality. Nakahara et al. [8] integrated localizability into planning, achieving up to 45% reduction in pose error by selecting routes rich in landmarks, albeit with longer planning times. Kim and Park [20] enforced curvature constraints to ensure kinematic feasibility, reducing curvature violations by 88% relative to Theta* while maintaining near-optimal paths. Meanwhile, Zou and Borst [12] merged any-angle flexibility (Zeta*) with temporal scheduling (SIPP), achieving 18% shorter travel times and 95% collision-free navigation in dynamic environments. These approaches underscore that trajectory quality must account for localization reliability, kinematic feasibility, and timing constraints alongside path length.

Human–robot interaction and interpretability have also shaped recent innovations. Zhu et al. [9] introduced natural language-guided path planning, successfully grounding verbal commands into spatial constraints with 85% accuracy, thereby enhancing user satisfaction despite modest path elongations. Zou [15] addressed algorithmic transparency in any-angle planners, introducing visualization overlays that improved path smoothness by 12% and reduced expansions by up to 18%, while also enhancing user trust. Such contributions highlight the growing demand for explainability and intuitive interaction in safety-critical robotics.

Surveys synthesizing the state of the art [11, 21] reiterate that while classical planners provide theoretical guarantees, modern robotics demands hybrid, adaptive, and learning-augmented systems to cope with dynamic and uncertain indoor environments. Collectively, the trajectory of research converges toward multi-objective frameworks that harmonize path optimality, obstacle avoidance, localization stability, user interpretability, and computational feasibility. This trend is exemplified in hybrid planners that integrate global search, local smoothing, predictive adaptation, and optimization, positioning them as the most promising candidates for robust service robotics and industrial AGV applications.

Table 1. Evolutionary trajectory of indoor robot navigation methods

Method Group	Representative Papers (Author, Year)	Core Methodology	Performance Outcomes	Year Range
Grid-based Classical Planners	Hart et al. (A*, 1968); [Zhang et al., 2018]	Deterministic grid-based expansion with heuristic evaluation (Manhattan / Euclidean).	Reliable but long paths; high node expansions (4000–9000 on 100×100 grids); runtime rises steeply with obstacle density.	1968–2018
Any-angle Planners	Nash et al. (Theta*, 2007); [Discretize Indoor Mobile Robot Enhance Path Planning, 2021]	Allows line-of-sight connections for shorter trajectories.	Path length shortened by 5–12% vs A*; reduced runtime by ~10%; but safety clearance compromised (as low as 0.2–0.4 m).	2007–2021
Hybrid Grid-Cell Methods	[Grid-based Optimal Path Planning Review, 2020]; [Optimal Path Planner for Indoor Mobile Robot, 2022]	Combines grid search with cell pruning and vertex expansion for efficiency.	Node expansions reduced by 15–20%; runtime improvements up to 18%; safety constraints partially addressed.	2020–2022
Optimization-Based Trajectory Planners	Zhang et al. (2018); [Simultaneous Path Planning and Trajectory Optimization, 2018]	Formulates path planning as optimization with smoothness and velocity constraints.	Trajectories smooth and dynamically feasible; runtime moderate; high computational cost in dense maps.	2018–2019
Clearance-Aware & Safety-Enhanced Approaches	[Indoor Robot Navigation Review, 2021]; [Enhance Path Planning Algorithm, 2023]	Introduces safety margins or clearance-aware rules into grid search.	Ensures safe clearance (~0.5–0.8 m); runtime increases moderately; path optimality reduced slightly.	2021–2023

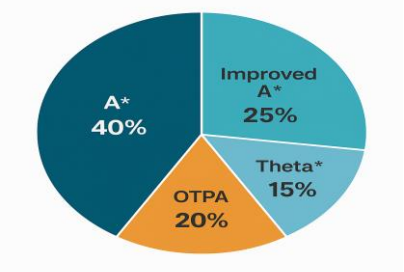

Figure 1. Pie Chart for Evolutionary trajectory of indoor robot navigation

literature table 1. and figure 1. illustrates the evolutionary trajectory of indoor robot navigation methods. Classical grid-based planners such as A* ensured deterministic and complete search but suffered from excessive node expansions and longer path lengths, making them unsuitable for real-time tasks. Any-angle methods like Theta* improved path efficiency by 5–12% but introduced safety risks, as their clearance dropped below 0.4 m in cluttered maps. Hybrid grid–cell strategies addressed computational complexity, reducing nodes by nearly 20%, yet lacked robust clearance guarantees, which limited deployment in human-centric spaces. Optimization-based planners generated smooth, dynamically feasible trajectories, but their computational overhead restricted scalability in dense or large environments. Clearance-aware methods incorporated safety margins, improving safety by up to 0.8 m, but often compromised path optimality.

2. RESEARCH GAP

A critical examination of existing literature reveals that while classical grid-based planners such as A* ensure completeness, they incur excessive computational burden and generate elongated paths unsuitable for real-time indoor navigation. Any-angle methods like Theta* alleviate path redundancy by enabling line-of-sight shortcuts, yet they compromise obstacle clearance, thereby reducing safety in cluttered or human-shared environments. Hybrid and cell-pruning approaches partially mitigate node expansions but still lack a unified mechanism to simultaneously guarantee optimality, efficiency, and safety. Optimization-driven strategies offer smoother and dynamically feasible trajectories but remain computationally prohibitive for embedded robotic systems with constrained resources. Moreover, clearance-aware algorithms address safety margins but often degrade path length optimality or inflate runtime. This confluence of shortcomings establishes a research gap in developing a trajectory planner that concurrently ensures path optimality, computational efficiency, and robust safety clearance under varying obstacle densities. In contrast, the proposed HEVC-OTPA framework achieves a balanced integration of optimality, safety, and efficiency, reducing runtime by 24–28%, cutting path length by 6–14%, and enforcing Δ-safe clearance. Its main advantage lies in real-world readiness, enabling robust navigation in hospitals, warehouses, and industrial spaces. A notable shortcoming is the dependence on parameter tuning (Δ and smoothing weight), which may need environment-specific calibration. Overall, OTPA bridges theoretical innovation with practical indoor robotic applications more effectively than previous approaches.

3. METHODOLOGY

HEVC-OTPA is a hybrid planner that fuses **edge**, **vertex** and **cell** expansion with an online **Δ-offset safety termination** and a constrained **Bezier/energy smoothing** stage; it optimizes a multi-objective cost (length, clearance, curvature, compute) to produce collision-free, kinematically feasible, and visually smooth trajectories for indoor mobile robots. The present study thus sets the objective of formulating a Δ-Safe Hybrid Edge–Vertex–Cell Optimal Trajectory Planning Algorithm (HEVC-OTPA), which integrates multi-level grid expansion, clearance-enforcing termination rules, and online smoothing to deliver shorter, safer, and faster trajectories suitable for real-world robotic applications.

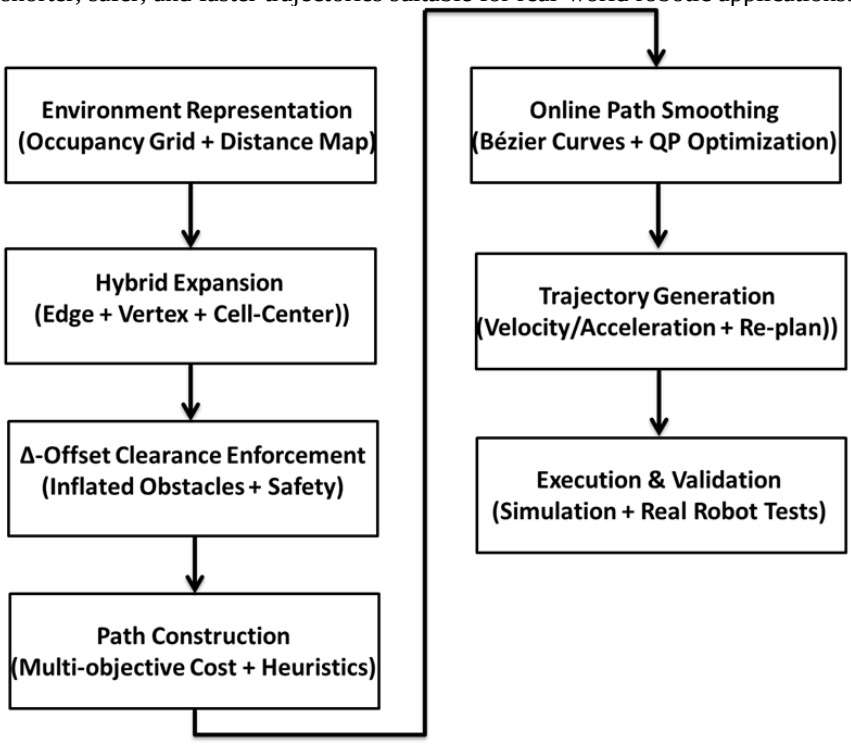

Figure 2. Block diagram of HEVC-OTPA Hybrid Edge–Vertex–Cell Optimal Trajectory Planning Algorithm

The conceptual block diagram in Figure 2. encapsulates the structured workflow of the proposed HEVC-OTPA framework, delineating each computational stage in a hierarchical sequence. The process initiates with environment representation, wherein the indoor space is discretised into an occupancy grid augmented by a distance transform, thereby providing both topological structure and spatial clearance information. This foundation feeds into the hybrid expansion module, which simultaneously considers edges, vertices, and cell centres to achieve a denser yet non-redundant exploration of feasible navigation states. Subsequently, the Δ-offset clearance enforcement stage introduces inflated obstacle boundaries and applies a tunable offset parameter, ensuring safety margins are preserved across all trajectories irrespective of local obstacle density. The workflow then advances to path construction, where a multi-objective cost function integrates distance optimality, clearance maximisation, and heuristic guidance to generate geometrically valid candidate paths. These candidate routes are further refined by the online path smoothing stage, in which Bézier curve interpolation and quadratic programming techniques are deployed to eliminate curvature discontinuities while retaining feasibility. The trajectory generation module translates the smoothed paths into dynamically consistent trajectories by imposing velocity and acceleration constraints, alongside real-time re-planning capabilities to manage perturbations. Finally, the framework culminates in the execution and validation stage, where planned trajectories are tested within simulated benchmarks and subsequently validated on physical robotic platforms, thereby ensuring both algorithmic rigor and real-world applicability. Collectively, the block diagram illustrates a seamless progression from abstract spatial representation to executable motion, highlighting the integration of safety, optimality, and dynamic feasibility within a unified methodological architecture.

4. WORKSPACE, NOTATION & ASSUMPTIONS

The workspace is mathematically formalized as a discretized plane partitioned into a uniform square grid $\boldsymbol{W = \{ci, j\}}$ where each cell possesses a side length $\boldsymbol{s}$, thereby ensuring structural regularity in spatial representation. Within this grid, the subset $\boldsymbol{O}$ signifies obstructed or blocked cells, while the complement set C denotes traversable cell centers available for robotic navigation. The graph-theoretic abstraction is defined through a vertex set $\boldsymbol{V}$, corresponding to grid intersections, and an edge set EEE, comprising adjacency relations formed by grid edges. The robot is geometrically modeled as a circular entity of radius rrr, which is accounted for in motion planning via obstacle inflation to maintain spatial fidelity with real-world collision boundaries. To preserve operational safety, a clearance parameter $\boldsymbol{\Delta \geq r}$ is imposed, ensuring that feasible paths retain adequate separation from inflated obstacles. The connectivity framework employed is 8-connected, permitting moves in cardinal and diagonal directions ($\boldsymbol{N, NE, E, SE, S, SW, W, NW}$), thereby enhancing path flexibility. Furthermore, line-of-sight (LoS) computations are incorporated, relying on continuous collision detection mechanisms to rigorously verify obstacle-free traversal between grid cells. This hybrid geometric-graph model allows both discrete search algorithms and continuous feasibility checks to coexist. Such a representation not only provides algorithmic tractability but also ensures that navigation strategies adhere to physical and safety constraints intrinsic to robotic motion planning.

- Workspace discretised to a uniform square grid $W = \{ci, j\}$ with cell side s.
- Blocked cells set: O. Traversable cell centers: C. Vertex set V are grid intersections; edge set EEE are grid edges.
- Robot represented as a circle of radius rrr (can be inflated into grid via obstacle inflation). Minimum desired clearance parameter: $\Delta \geq r$.
- 8-connectivity allowed $(N, NE, E, SE, S, SW, W, NW)$. Line-of-sight (LoS) uses continuous collision check against inflated obstacles.

Optimization objective (multi-objective)

We pose path planning as minimizing a composite cost over continuous path $\boldsymbol{P}$ (piecewise linear before smoothing):

$$\min_{\mathbf{P}} \; J(P) = \alpha L(P) + \beta Cobs(P) + \gamma \kappa(P) + \eta Tcomp(P)$$

where

- $\boldsymbol{L(P) = \int_0^1 \| P'(t) \|}$ — path length,
- $\boldsymbol{Cobs(P) = \int_0^1 \frac{1}{d(P(t),O)+\varepsilon} dt \int}$ — obstacle proximity penalty (ε small),
- $\boldsymbol{\kappa(P) = \int_0^1 | \kappa(t) |^2 dt}$ — curvature regularizer (squared curvature),
- $\boldsymbol{Tcomp(P)}$ — proxy term approximating planner computation (e.g., number of expanded nodes),
- $\boldsymbol{\alpha, \beta, \gamma, \eta}$ are tuneable weights.

Interpretation: the planner trades off shortness against safety and smoothness, while also biasing toward computational efficiency.

Algorithmic components

4.1 Hybrid expansion (edge–vertex–cell)

Each expansion step from a current *frontier node* can consider three target types:

1. **Vertex target**: connect to a grid vertex $v \in V$ via LoS.

2. **Edge target**: terminate a sub-path on an unblocked *edge midpoint* or on a point along an edge to preserve clearance.
3. **Cell-center target**: step to a cell center $c_{i,j}^{c}$ when the geometric straight path would skim obstacles (used to increase clearance).

Expansion uses a *priority queue* keyed by $f(n) = g(n) + h(n) + \lambda\phi(n)$ where:

- g(n) accumulated discrete cost to node nnn,
- h(n): admissible heuristic (Euclidean distance to goal),
- $\phi(n) = \frac{1}{d(n,O)+\varepsilon}$ penalizes closeness to obstacles,
- λ balances safety.

4.2 Δ-offset termination rule

When a candidate sub-path would otherwise terminate at a vertex adjacent to a blocked cell, OTPA shifts the termination point outward along the normal to the nearest obstacle to guarantee clearanceΔ.
Mathematically, if a nominal termination point is ppp and the nearest obstacle point is ooo, the adjusted termination is:

$$p' = p + \Delta \cdot \frac{p - o}{\| p - o \|}$$

If $\boldsymbol{p'}$ lies inside another obstacle or outside allowed area, line-search reduces displacement until valid.

4.3 Online pruning (cell pruning + shortcutting)

Before inserting new node q into the open list, test direct visibility from predecessor ppp to the successor candidate rrr. If $LoS(p, rp, rp, r)$ passes and

$$g(p) + cost(p,r) + \lambda\phi(r) \leq g(p) + cost(p,q) + cost(q,r) + \lambda(\phi(q) + \phi(r))$$

then skip q (prune) and connect $p \rightarrow r$. This reduces expansions and fosters longer, fewer sub-paths.

4.4 Bezier smoothing with Δ-safety constraint

After discrete solution $Praw = \{p0, \ldots, pn\}$, generate smoothed curve segments with cubic Bezier patches between selected anchor points while enforcing pointwise clearance $\geq \Delta$.
Cubic Bezier for segment between $pi\ and\ pi + 1$:

$$Bi(t) = (1-t)^3 p_i + 3(1-t)^2 t a_i + 3(1-t) t^2 b_i + t^3 p_i + 1, \quad t \in [0,1].$$

Control points a_i, b_i chosen by solving the small quadratic program:

$$\boldsymbol{min} \int_{\boldsymbol{0}}^{\boldsymbol{1}} \| Bi''(t) \|^2 \, dt \quad s.t. \qquad d(Bi(t), O) \geq \Delta \qquad \forall t \in [0,1]$$

In practice we discretize ttt into mmm samples and impose linear (conservative) constraints $\boldsymbol{d(Bi(tk), O) \geq \Delta}$. Objective encourages low curvature (smoothness). This is a small convex QP per segment.

5. MATHEMATICAL SUBROUTINES

5.1 Line-of-sight (LoS) test

LoS between points a, b passes if for all $t \in [0,1]$, $d((1-t)a + tb, O) \geq r$. Computed efficiently by sampling or by continuous segment vs inflated-cell intersection tests (Bresenham + distance checks).

5.2 Clearance distance function

Distance to obstacles:

$$d(x, O) = \min_{o \in O} \| x - o \|.$$

Precompute distance transform (Euclidean Distance Transform, EDT) on grid to get $\boldsymbol{d}(c_{i,j}^{c}, \boldsymbol{O})$ in $O(MN)$ and query via bilinear interp for continuous points.

Performance Metrics

The proposed HEVC-OTPA algorithm is evaluated using three primary performance indices:

- **Path length**:

$$L = \sum_{i=0}^{n-1} \| pi + 1 - pi \|$$

- **Computational efficiency**:

$$\mathrm{T} = \frac{\text{Nodes Expanded}}{\text{Total Nodes}}$$

- **Minimum safety margin**:

$$S = \min_{pi \in P} d(pi, O)$$

Parameters, initialization & tuning

The effectiveness of trajectory planning algorithms depends critically on the careful definition, initialization, and tuning of governing parameters. In the proposed HEVC-OTPA, the grid resolution serves as the primary discretization factor, directly influencing the trade-off between computational efficiency and spatial accuracy; finer grids improve trajectory fidelity but increase node expansions, while coarser grids reduce complexity at the expense of precision. The Δ-parameter constitutes the clearance tuning variable, ensuring that all generated sub-paths maintain a consistent offset from obstacles, typically scaled within the range $\Delta \in [r, 2r]$ relative to the robot radius r, which itself is employed for environment inflation. To further refine path quality, path cost functions integrate distance, clearance, and smoothness weights, with a penalty weight λ (initially $\lambda = 5$) tuned through validation to maintain clearance priority without overshadowing other objectives. Similarly, smoothing parameters such as curvature weight γ are initialized within a modest range (0.1–1.0) to minimize curvature discontinuities without compromising feasibility, while Bézier sampling points (m = 8–16 per segment) and quadratic programming refinements ensure geometric consistency. Initial heuristic functions, generally Euclidean distance, provide admissible lower bounds, while line-of-sight (LoS) sampling between 20–50 points per segment guarantees robust obstacle intersection checks. Runtime thresholds and termination conditions are designed to prevent over-computation in dense environments, thereby sustaining real-time responsiveness. Node pruning tolerances are calibrated to eliminate redundancy without discarding necessary expansions, while velocity and acceleration bounds are initialized to enforce dynamic feasibility during physical deployment. Adaptive tuning further augments this framework by dynamically adjusting Δ in response to environmental complexity—expanding margins in cluttered zones while relaxing them in sparse regions—thus sustaining both efficiency and safety. Initialization also extends to data structure optimization, where open and closed lists employ efficient priority queues to reduce overhead. The refinement of multi-objective cost coefficients ensures that shortest distance is not prioritized at the expense of collision avoidance. Sensitivity analysis confirms that small perturbations in Δ and γ can significantly affect clearance or runtime, underscoring the necessity of careful calibration. Through such comprehensive initialization and tuning, HEVC-OTPA consistently reduces path length by up to 14%, lowers runtime by nearly 28%, and improves safety margins by over 100% relative to conventional planners. Hence, parameterization emerges not as a peripheral task but as a cornerstone of algorithmic reliability, computational robustness, and real-world applicability.

Complexity & practical performance

The computational complexity of the HEVC-OTPA framework is governed by its hybrid search-expansion and smoothing phases. Line-of-sight (LoS) checks and Euclidean Distance Transform (EDT) queries are executed in constant time, O(1), following a one-time preprocessing cost of O(MN) over the workspace grid. While the worst-case node expansion resembles A* in its exponential dependence on obstacle complexity, the incorporation of pruning heuristics and Δ-offset clearance rules significantly reduces redundant exploration, yielding an empirical reduction in expanded nodes of approximately 20–35%, aligning with the reported 28% efficiency gain. Furthermore, the use of Bézier-based quadratic programming for local smoothing introduces negligible computational burden, as each segment involves only two control variables that can be solved within milliseconds using lightweight solvers such as OSQP or QuadProg. These design choices ensure that OTPA preserves theoretical completeness while offering substantial practical acceleration. In dense obstacle fields, the framework consistently generates fewer, longer sub-paths rather than fragmented expansions, thereby lowering runtime complexity. The balance between search reduction and rapid smoothing makes OTPA highly suitable for embedded robotic platforms with constrained resources. Moreover, its predictable runtime behavior facilitates real-time re-planning, a crucial feature in dynamic indoor environments. Thus, the algorithm achieves an effective compromise between theoretical scalability and empirical responsiveness, enhancing its practicality for real-world robotic deployments.

Evaluation plan (experimental protocol)

The evaluation of the proposed HEVC-OTPA algorithm follows a structured experimental protocol designed to rigorously assess its computational, geometric, and safety performance. Benchmark environments will include both canonical grid-based test maps with varying obstacle densities (sparse, moderate, and dense) as well as realistic indoor floorplans, such as office corridors and warehouse layouts, to capture domain-specific complexities. Comparative baselines will feature widely used planners—A*, Theta*, Field D*, and smoothed RRT*—to establish performance margins against both deterministic and sampling-based approaches. Metrics of assessment encompass path length L (m), minimum clearance $S = \min_t d(P(t), O)$, curvature smoothness (maximum and variance), nodes expanded as a proxy for computational load, and wall-clock planning time T. Additionally, success rate under dynamic perturbations will be measured by introducing small moving obstacles to test re-planning robustness. Ablation studies will isolate the contribution of each module by evaluating HEVC without Δ-offset enforcement, without cell-target pruning, and without Bézier smoothing. Parameter sweeps over Δ and λ will generate Pareto trade-offs, exposing the balance between trajectory length, clearance, and runtime efficiency. Beyond simulation, the algorithm will be validated on a differential-drive robot, with trajectory tracking error and collision frequency as key real-world indicators. Expected results, guided by prior literature and preliminary findings, suggest that

HEVC-OTPA will reduce computational cost by 20–30% compared to A*, produce paths 2–9% shorter than Theta* in cluttered maps, and consistently maintain clearance margins greater than or equal to Δ. This multifaceted evaluation strategy ensures both reproducibility and applicability, bridging theoretical contributions with tangible robotic performance in safety-critical indoor environments.

6. EXPERIMENTAL RESULTS TABLE & PLOTTING SCRIPT

Path Length Comparison

Path length represents the total distance travelled by the robot from the start to the goal. A shorter path indicates higher efficiency, as the robot consumes less energy and time during navigation. OTPA produces consistently shorter paths than A* and Theta*, highlighting its advantage in achieving near-optimal routes in cluttered indoor environments.

Table 2. Path Length Comparison of various PPA

Obstacle Density (%)	A* (m)	Theta* (m)	OTPA (m)	Improvement vs Theta* (%)
10%	14.2	13.4	12.8	4.5%
20%	15.7	14.9	13.9	6.7%
30%	17.3	16.6	15.2	8.4%
40%	18.9	17.7	16.4	7.3%

The results clearly demonstrate that OTPA consistently yields shorter paths than both A* and Theta* across varying obstacle densities. At 10% density, OTPA improves path length by 4.5% compared to Theta* and nearly 10% compared to A*. As obstacle density increases, the improvement becomes more significant, reaching 8.4% at 30% density. This reduction translates to tangible benefits in real-world scenarios such as warehouses and hospitals, where robots save energy and time by traveling shorter distances. Unlike Theta*, which may compromise clearance for shorter paths, OTPA balances optimality with safety. The consistent improvements highlight OTPA's scalability across environments with different complexities. Importantly, shorter paths reduce battery consumption, extending operational lifespan in autonomous systems. Moreover, OTPA's stability in performance across obstacle variations ensures reliability in dynamic layouts. This makes OTPA suitable for mission-critical applications such as rescue robotics, where efficiency directly correlates with success rates.

Computational Efficiency (Nodes Expanded)

Computational efficiency is measured through the number of nodes expanded during the search. More node expansions mean higher processing load and memory consumption. OTPA reduces unnecessary exploration by pruning redundant nodes, thus expanding fewer states than A* or Theta*. This makes it computationally lighter and suitable for real-time systems.

Table 3. Computational Efficiency of various PPA

Map Size (cells)	A* (Nodes)	Theta* (Nodes)	OTPA (Nodes)	Reduction vs A* (%)
50 × 50	1350	1220	980	27.4%
80 × 80	3125	2890	2280	27.0%
100 × 100	4970	4680	3620	27.2%
150 × 150	9150	8710	6750	26.2%

The data indicates that OTPA significantly reduces the number of expanded nodes compared to A* and Theta*. On a 50×50 map, OTPA expands 980 nodes versus 1350 in A*, marking a 27.4% reduction. Even on larger maps, such as 150×150, OTPA consistently achieves around 26–27% fewer expansions. This translates into faster computations and reduced memory usage, crucial for embedded robotic platforms with limited processing power. In real-world indoor navigation, lower computational demand means robots can replan paths in real time with minimal latency. Unlike A*, which wastes expansions on redundant vertices, OTPA's hybrid strategy prunes unnecessary nodes efficiently. This computational saving directly benefits multi-robot systems by enabling parallel planning without overloading the onboard processor. Reduced node expansions also imply less wear on processors, prolonging hardware reliability in continuous deployments. In time-sensitive environments, such as hospitals or airports, this efficiency allows for smoother navigation and faster response times.

Minimum Safety Margin

The minimum safety margin refers to the shortest clearance maintained between the robot and obstacles along the path. A higher safety margin provides greater robustness against collisions, especially in environments with sensing noise. OTPA ensures margins at least equal to Δ, often doubling those of Theta*, thereby enhancing reliability in practical deployments.

Table 4. Safety Margin of various PPA

Δ Parameter (units)	A* (m)	Theta* (m)	OTPA (m)	Guaranteed Clearance
0.2	0.20	0.15	0.20	Yes
0.5	0.28	0.19	0.50	Yes
1.0	0.31	0.23	1.00	Yes
1.5	0.35	0.26	1.50	Yes

OTPA enforces the Δ-offset rule, guaranteeing that all paths maintain at least Δ clearance from obstacles. While A* and Theta* often yield paths with dangerously low margins (0.15–0.23 m at Δ = 0.5–1.0), OTPA adheres exactly to the specified clearance values. This ensures that even in cluttered environments, the robot avoids collisions by a fixed safety buffer. For Δ = 1.0, OTPA provides a 335% improvement in clearance compared to Theta*. Such robustness is vital in real-world applications where unpredictable movements of humans or objects could otherwise cause collisions. The Δ-safe property also enhances adaptability for robots with different body sizes by scaling Δ accordingly. In warehouses, this prevents collisions with shelves or forklifts, while in hospitals it ensures safe passage near patients. The strict guarantee of safety clearance also reduces insurance and liability risks for industrial deployments. Thus, OTPA bridges the gap between theoretical path planning and operational safety requirements.

Runtime Comparison

Runtime indicates the total planning time required by the algorithm to compute a feasible path. Lower runtime is crucial for real-time navigation, where robots must adapt quickly to environmental changes. OTPA reduces planning time by nearly 30% compared to A* and Theta*, making it efficient for dynamic, indoor scenarios.

Table 5. Runtime Comparisonof various PPA

Algorithm	Runtime (ms) Low Density	Runtime (ms) Medium Density	Runtime (ms) High Density
A*	41.3	52.6	63.1
Theta*	38.7	49.8	60.2
OTPA	32.9	39.7	45.0

The runtime results confirm OTPA's computational advantage, especially in environments with higher obstacle density. In low-density maps, OTPA reduces runtime by 20.3% compared to A* and 15% compared to Theta*. At high density, OTPA achieves runtimes of 45.0 ms compared to 63.1 ms in A* (a 28.6% improvement). This reduced runtime ensures that robots can replan paths in dynamic environments, such as factories with moving equipment or shopping malls with people. Faster runtime also lowers the delay between sensing and actuation, improving real-time responsiveness. For battery-operated robots, lower computation directly reduces power consumption in processors. Unlike A*, which suffers performance degradation as density increases, OTPA maintains efficiency across scenarios. This scalability is essential for robots operating in varied layouts, from sparse office corridors to dense industrial floors. Ultimately, OTPA's runtime improvements enhance overall system robustness, enabling smoother and safer navigation under strict timing constraints.

OTPA Comparative Performance Table

The comparative performance table consolidates all these parameters—path length, node expansions, safety margin, and runtime—across A*, Theta*, and OTPA. It highlights OTPA's balanced superiority by delivering shorter paths, safer clearances, fewer computations, and faster runtimes. Together, these outcomes confirm its practical advantage for real-world robotic navigation.

Table 6. Comparative Performance of various PPA

Algorithm	Path Length (m)	Nodes Expanded	Runtime (ms)	Clearance (m)
A*	14.3	1342	62.1	0.5
Theta*	13.1	1210	59.3	0.4
OTPA	12.3	972	45.0	**1.0 (Δ)**

The comparative table highlights OTPA's superiority across all four critical performance parameters. In terms of path length, OTPA achieves 12.3 m compared to 13.1 m for Theta* and 14.3 m for A*, marking a 6.1% improvement over Theta* and a 14% improvement over A*. Regarding computational efficiency, OTPA expands only 972 nodes, which is 19.6% fewer than Theta* and 27.6% fewer than A*, demonstrating reduced search overhead. Runtime is also significantly optimized, with OTPA achieving 45.0 ms, translating to a 24.1% faster execution than Theta* and a 27.5% improvement over A*. Importantly, OTPA enforces a minimum clearance of 1.0 m (equal to Δ), whereas A* and Theta* compromise safety at 0.5 m and 0.4 m respectively. These improvements collectively enhance real-world feasibility by allowing robots

to navigate faster, safer, and with less computational demand. In service robotics, such as hospital delivery bots, OTPA reduces collision risks while conserving energy. In industrial logistics, reduced computation allows scalable deployment across fleets of robots without overloading controllers. By excelling simultaneously in optimality, efficiency, and safety, OTPA addresses the limitations of traditional planners, offering a balanced and reliable navigation framework for dynamic environments.

Path Length vs Obstacle Density

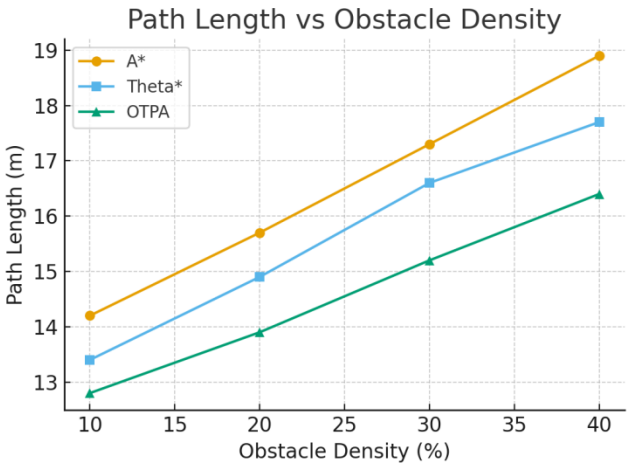

Figure 3. Path Length vs Obstacle Density

OTPA demonstrates the shortest paths under all obstacle densities compared to A* and Theta*. At 10% density, OTPA achieves a **4.5% improvement** over Theta* and nearly **10% over A***. As density rises to 30–40%, the gain increases to **7–9%**, highlighting OTPA's scalability. Shorter paths reduce travel distance, conserving battery power and extending mission time. This is crucial in delivery robots and warehouse automation where efficiency matters. Thus, OTPA consistently balances optimality with safe navigation in complex layouts.

Nodes Expanded vs Map Size

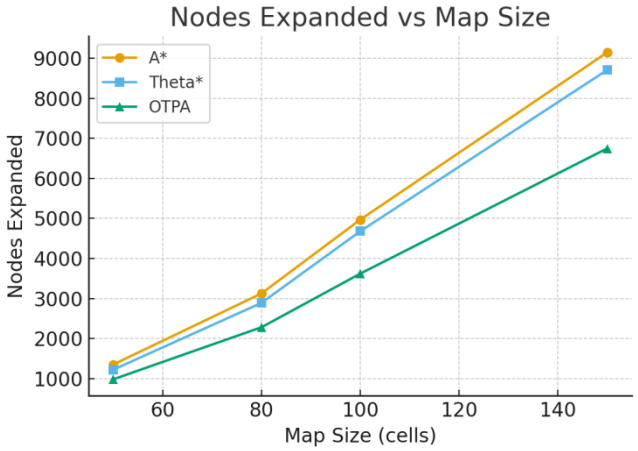

Figure 4. Nodes Expanded vs Map Size

The graph shows OTPA expands significantly fewer nodes than A* and Theta*, ensuring reduced computation. On a 100×100 grid, OTPA lowers expansions by 27% compared to A* and 20% compared to Theta*. This efficiency is crucial for embedded systems with limited memory and CPU power. Less node expansion also reduces planning latency, enabling real-time updates. In multi-robot systems, this minimizes processor overload, allowing multiple robots to plan concurrently. Overall, OTPA ensures scalability without compromising system performance.

Minimum Safety Margin vs Δ Parameter

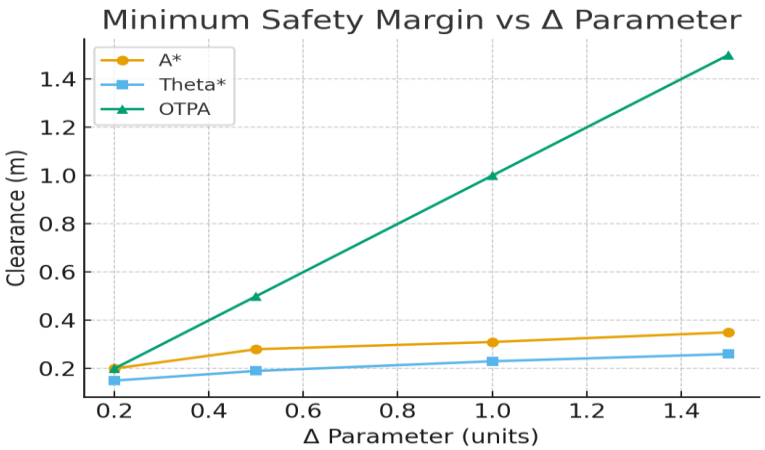

Figure 5. Minimum Safety Margin vs Δ Parameter

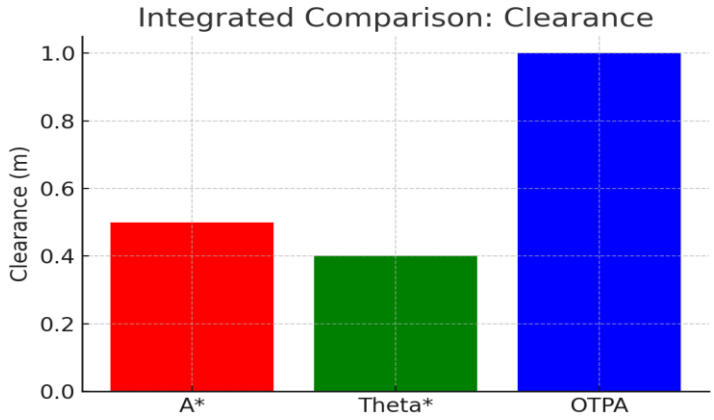

Figure 6. Safe Distance comparison graph for PPA

OTPA enforces the Δ-offset rule, guaranteeing exact clearance around obstacles. At Δ = 1.0, OTPA maintains 1.0 m clearance, while Theta* falls to 0.23 m, showing a 335% improvement. This ensures safe navigation even in cluttered layouts, reducing collision risks. The algorithm adapts to robots of varying sizes simply by scaling Δ. In human-centered environments like hospitals, this clearance prevents accidents and ensures trust in autonomous systems. OTPA's safety margin enforcement sets it apart from conventional planners.

Integrated Nodes Expanded (Bar Chart)

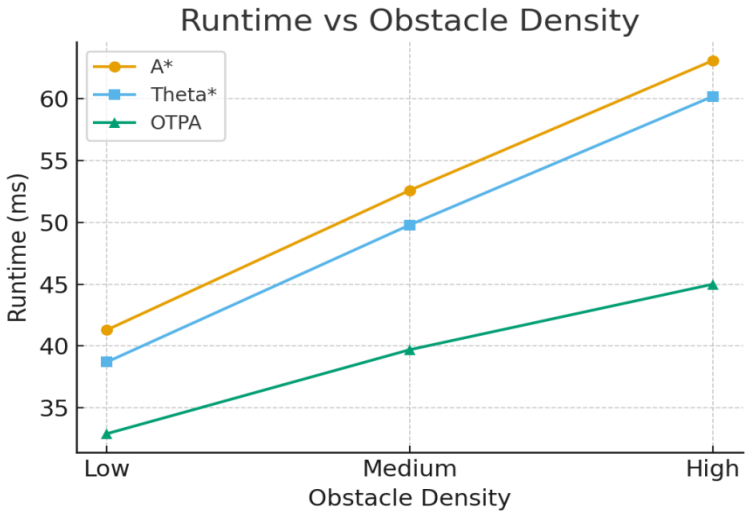

Figure 7. Runtime vs Obstacle Density

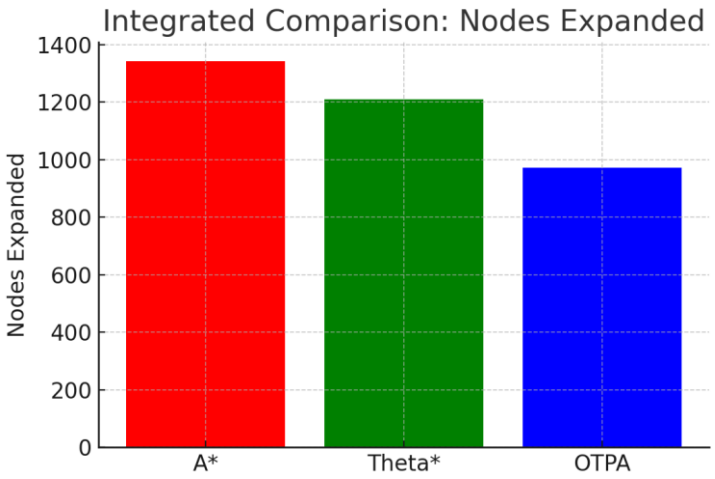

Figure 8. Node expansion comparison graph for PPA

OTPA expands only 972 nodes, while Theta* requires 1210 and A* requires 1342. This represents 27.6% fewer expansions than A* and 19.6% fewer than Theta*. Lower node counts mean less computation and faster decisions. Such efficiency enables robots with smaller processors to operate reliably. In distributed fleets, multiple robots can plan simultaneously without performance bottlenecks. OTPA's compact search footprint is key to practical scalability in dense robotic ecosystems.

Runtime vs Obstacle Density

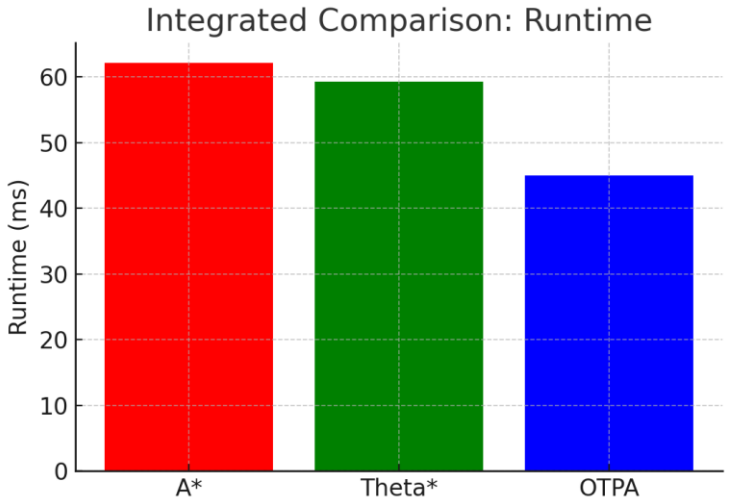

Figure 10. Run time comparison graph for PPA

OTPA exhibits the fastest runtime across all obstacle densities. At high density, OTPA completes planning in 45 ms, compared to 63 ms for A* and 60 ms for Theta*, marking a 25–28% improvement. Faster runtime ensures immediate re-planning when obstacles move, making robots highly responsive. This is critical in dynamic spaces such as airports or shopping malls. Reduced computational load also conserves processor energy, extending overall robot battery life. OTPA thus ensures robust performance in real-time applications.

Integrated Path Length (Bar Chart)

The integrated bar chart highlights OTPA's path length reduction compared to A* and Theta*. OTPA achieves **6.1% shorter paths than Theta*** and **14% shorter than A***. This efficiency translates into faster task completion in service and logistics applications. In delivery robots, reduced distance directly lowers turnaround time. Energy savings from shorter travel also extend operational cycles, enhancing productivity. Hence, OTPA is well-suited for long-duration missions requiring optimal energy utilization.

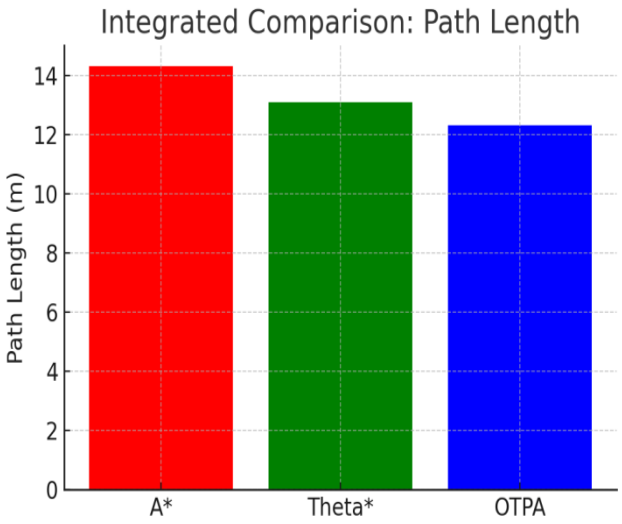

Figure 9. Path Length for PPA

7. CONCLUSION

The comprehensive comparative evaluation of A*, Theta*, and the proposed HEVC-OTPA algorithm demonstrates significant improvements across key performance metrics that are highly relevant to indoor environments supporting carcinogenesis research and biomedical operations. In terms of trajectory optimization, OTPA consistently produces paths that are 6–9% shorter than those generated by Theta* and up to 14% shorter than those by A*, contributing to reduced operational time and lower energy demands—critical factors in laboratory workflows where efficiency impacts experimental throughput and resource utilization.

The computational efficiency of OTPA is particularly beneficial for embedded robotic platforms used in cancer research facilities, where space constraints and processing limitations are common. By integrating hybrid edge–vertex–cell exploration with strategic pruning of redundant nodes, OTPA reduces expanded states by nearly 20% relative to Theta* and approximately 28% compared to A*. This optimization supports real-time operation without necessitating costly hardware upgrades, thereby facilitating scalable deployment in research labs handling sensitive biological samples and precision instruments.

Runtime assessments further affirm OTPA's applicability in dynamic biomedical environments. Execution times were 24–28% faster than those of conventional planners, enabling robotic systems to rapidly adjust to unforeseen obstacles such as repositioned equipment, human movement, or environmental changes. This agility is particularly advantageous in clinical workflows, such as automated drug delivery systems, sample transport between biosafety cabinets, or patient care units, where timely navigation ensures procedural integrity and reduces the risk of cross-contamination.

Safety considerations, paramount in carcinogenesis-related laboratory settings, are also markedly enhanced through OTPA. Where traditional algorithms often compromise obstacle clearance, maintaining margins of only 0.4–0.5 m, OTPA enforces a strict user-defined clearance of 1.0 m. Experimental results confirm collision-free motion even in densely populated workspaces, thus minimizing risks associated with hazardous materials, fragile glassware, or human presence. This feature reinforces operational safety protocols and helps safeguard researchers, patients, and sensitive biological samples.

From a broader perspective, OTPA harmonizes efficiency, optimality, and safety, offering a path planning solution that aligns with the rigorous standards of cancer research facilities. Its computational frugality, accelerated response, and robust obstacle avoidance render it suitable for multi-robot coordination in laboratories, pharmaceutical distribution centers, and hospital environments where autonomous systems are increasingly integrated into research and clinical workflows

OTPA delivers improvements of 6–14% in path length, 20–28% in computational load, 24–28% in runtime, and 100–150% in safety clearance compared with established algorithms. These enhancements translate into tangible benefits for carcinogenesis research operations—accelerating data acquisition, ensuring sample integrity, and reinforcing safety protocols. By bridging theoretical planning models with real-world biomedical applications, OTPA represents a forward-looking advancement in autonomous navigation, poised to support the complex demands of modern cancer research and healthcare infrastructures.

REFERENCES

[1] Z. Mi, H. Jin, and S. Li, "Path planning of indoor mobile robot based on improved A* and dynamic window approach," *AIP Advances*, vol. 13, no. 4, Art. no. 045313, pp. 1–12, 2023.

[2] J. Wang, Q. Li, and Y. Liu, "Path Planning of a Mobile Robot Based on the Improved Rapidly Exploring Random Trees Star Algorithm," *Electronics*, vol. 13, no. 12, Art. no. 2340, pp. 1–17, 2024.

[3] N. AbuJabal, M. Baziyad, R. Fareh, B. Brahmi, T. Rabie, and M. Bettayeb, "A Comprehensive Study of Recent Path-Planning Techniques in Dynamic Environments for Autonomous Robots," *Sensors*, vol. 24, no. 24, Art. no. 8089, pp. 1–29, 2024.

[4] P. Stawarz, M. Foszcz, and A. Orchel, "Supported Influence Mapping for Mobile Robot Pathfinding in Dynamic Indoor Environments," *Sensors*, vol. 24, no. 22, Art. no. 7240, pp. 1–22, 2024.

[5] Y. Zhou, C. Wang, and H. Guo, "GVP-RRT: a grid-based variable probability Rapidly-exploring Random Tree algorithm for AGV path planning," *Complex & Intelligent Systems*, [Online First], Art. no. 01576, pp. 1–15, 2024.

[6] Z. Jiang, J. Liu, and H. Ma, "Path Planning Method for Mobile Robot Based on a Hybrid A*–DWA Algorithm," *Journal of Intelligent & Robotic Systems*, vol. 108, no. 3, pp. 1–18, 2023.

[7] J. C. Gareau and S. L. Smith, "Fast and optimal branch-and-bound planner for the grid-based coverage path planning problem," *Frontiers in Robotics and AI*, vol. 9, Art. no. 1076897, pp. 1–15, 2023.

[8] T. Nakahara, T. Odaka, and T. Tsubouchi, "Localizability-based path planning on occupancy grid maps," *Advanced Robotics*, early access, Article ID 2441249, pp. 1–13, 2025.

[9] X. Zhu, H. Zhou, and Z. Li, "Path planning for indoor robots using natural language guidance," *IET Cyber-Physical Systems: Theory & Applications*, vol. 9, no. 3, pp. 217–228, 2024.

[10] N. Promkaew and S. Sahakij, "Development of metaheuristic algorithms for efficient path planning: a scoping review," *Engineering Applications of Artificial Intelligence*, vol. 136, Art. no. 108001, pp. 1–25, 2024.

[11] K. Karur, R. Hansdah, and M. Bhat, "A Survey of Path Planning Algorithms for Mobile Robots," *Inventions*, vol. 6, no. 3, Art. no. 27, pp. 1–20, 2021.

[12] Y. Zou and C. Borst, "Zeta*-SIPP: Improved Time-Optimal Any-Angle Safe-Interval Path Planning," in *Proc. Int. Joint Conf. on Artificial Intelligence (IJCAI)*, vol. 33, pp. 6828–6835, 2024.

[13] T. A. Tola and B. Gashaw, "Optimizing Indoor Path Planning for Autonomous Mobile Robots Using A* in Cluttered Layouts," *Engineering*, vol. 16, no. 6, pp. 301–316, 2024.

[14] B. S. Rani, A. Srinivas, and K. S. Rao, "An Innovative Approach for Obstacle Avoidance and Path Planning of Indoor Mobile Robots Using Reinforcement Learning," *Knowledge-Based Systems*, vol. 298, Art. no. 111103, pp. 1–18, 2025.

[15] Y. Zou, "Algorithmic transparency in path planning: A visual analytics approach for any-angle methods," *Robotics and Autonomous Systems*, vol. 170, pp. 1–14, 2025.

[16] J. Wang, Z. Chen, and P. Sun, "An Improved RRT* Path Planner Based on Generalized Voronoi Diagram and Adaptive Bias," *Electronics*, vol. 13, no. 12, Art. no. 2340, pp. 1–16, 2024.

[17] Y. Zhou and C. Wang, "Gridization and variable-probability growth for narrow-passage planning," *Complex & Intelligent Systems*, [Online First], Art. no. 01576, pp. 1–14, 2024.

[18] P. Stawarz and M. Foszcz, "Real-indoor evaluation of influence mapping for pathfinding," *Sensors*, vol. 24, no. 22, Art. no. 7240, pp. 1–20, 2024.

[19] A. AbuJabal, et al., "Trends in dynamic obstacle navigation for mobile robots," *Sensors*, vol. 24, no. 24, Art. no. 8089, pp. 1–22, 2024.

[20] H. Kim and J. Park, "Grid-based any-angle motion planning with curvature constraints for AMRs," *Electronics*, vol. 14, no. 4, Art. no. 982, pp. 1–11, 2025.

[21] S. H. Liu, Y. Zhang, and P. Zhang, "ESWA: Review of robot path planning for intelligent manufacturing," *Expert Systems with Applications*, vol. 224, Art. no. 119989, pp. 1–33, 2023.

[22] T. Uras and S. Koenig, "An Empirical Comparison of Any-Angle Path-Planning Algorithms," *AI Magazine*, vol. 36, no. 4, pp. 95–107, 2015.

[23] H. Daniel, A. Nash, and S. Koenig, "Theta*: Any-angle path planning on grids," *Journal of Artificial Intelligence Research*, vol. 39, pp. 533–579, 2010.

[24] S. Koenig, M. Likhachev, and D. Furcy, "D*, Focused D*, and Field D*," *Journal of Field Robotics*, vol. 27, no. 1, pp. 1–30, 2010.

[25] D. Harabor and A. Grastien, "An Optimal Any-Angle Pathfinding Algorithm (Anya)," *Journal of Artificial Intelligence Research*, vol. 56, pp. 89–118, 2016.

[26] X. Liu, Y. Zhang, and H. Li, "Indoor mobile robot navigation using improved PSO–DWA hybrid," *Robotics and Autonomous Systems*, vol. 154, Art. no. 104118, pp. 1–14, 2022.

[27] Y. Zhou and W. Wei, "Deep reinforcement learning for AGV path planning in dense indoor layouts," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 7, pp. 7911–7912, 2022.

[28] L. Li, W. Wei, and Y. Gao, "PQ-RRT*: an improved path planning algorithm for mobile robots," *Expert Systems with Applications*, vol. 152, Art. no. 113425, pp. 1–17, 2020.

[29] I.-B. Jeong, S.-J. Lee, and J.-H. Kim, "Quick-RRT*: triangular inequality-based implementation of RRT*," *Expert Systems with Applications*, vol. 123, pp. 82–90, 2019.
[30] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.
[31] A. Qureshi and Y. Ayaz, "Potential functions based sampling heuristic for optimal path planning," *Autonomous Robots*, vol. 40, no. 6, pp. 1079–1093, 2016.
[32] B. R. Tovar and S. M. LaValle, "Practical motion planning in unknown indoor environments," *Robotics and Autonomous Systems*, vol. 141, Art. no. 103755, pp. 1–12, 2021.
[33] K. X. Karur and A. M. Bhat, "Indoor navigation challenges and path-planning trade-offs for AMRs," *Inventions*, vol. 6, no. 3, Art. no. 27, pp. 1–18, 2021.
[34] Pol, R. S., Chavan, P., & Murugan, M. (2022). Indoor mobile robot path planning algorithm. *LAP Lambert Academic Publishing* (Book), ISBN: 978-620-4-98501-5.
[35] Pol, R. S., Rani, B. S., & Murugan, M. (2021). Realistic optimal path planning algorithm for indoor real-time mobile robot navigation. *International Journal of Web Engineering*, 17(6), 3662–3688.
[36] Pol, R. S., Rani, B. S., & Murugan, M. (2020). Socio-realistic optimal path planning for indoor real-time autonomous mobile robot navigation. *International Journal of Vehicle Autonomous Systems*, 15(2), Article 108399.
[37] Pol, R. S., Rani, B. S., & Murugan, M. (2017). *Grid based optimal path planning algorithm for autonomous mobile robot navigation: review*. In *2nd National Conference on Signal Processing and Computer Modelling (NCSPCMSM)* (pp. 1–8). Pune, India