ICRA IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION ATLANTA 2025

Bidirectional Lazy Informed Trees

Asymptotically Optimal Sampling-based Motion Planning (SBMP) Through Anytime Incremental Lazy Bidirectional Hueristic Search

Yi Wang[1], Bingxian Mu[2], Oren Salzman[3]

[1]University of New Hampshire, [2]Independent Researcher, [3]Technion-Israel Institute of Technology

University of New Hampshire
TECHNION Israel Institute of Technology

Challenges, Open Questions, Contributions

Challenges: Computational overhead in batch-wise SBMP (Bw-SBMP)

1. Unidirectional heuristic search (Uni-HS) without lazy edge evaluation (e.g.,BIT*, ABIT*).
2. Lazy Uni-HS incurs high computational cost (e.g. AIT*, EIT*).
3. Extensive edge checks are still required as the graph size increases.

Open questions:

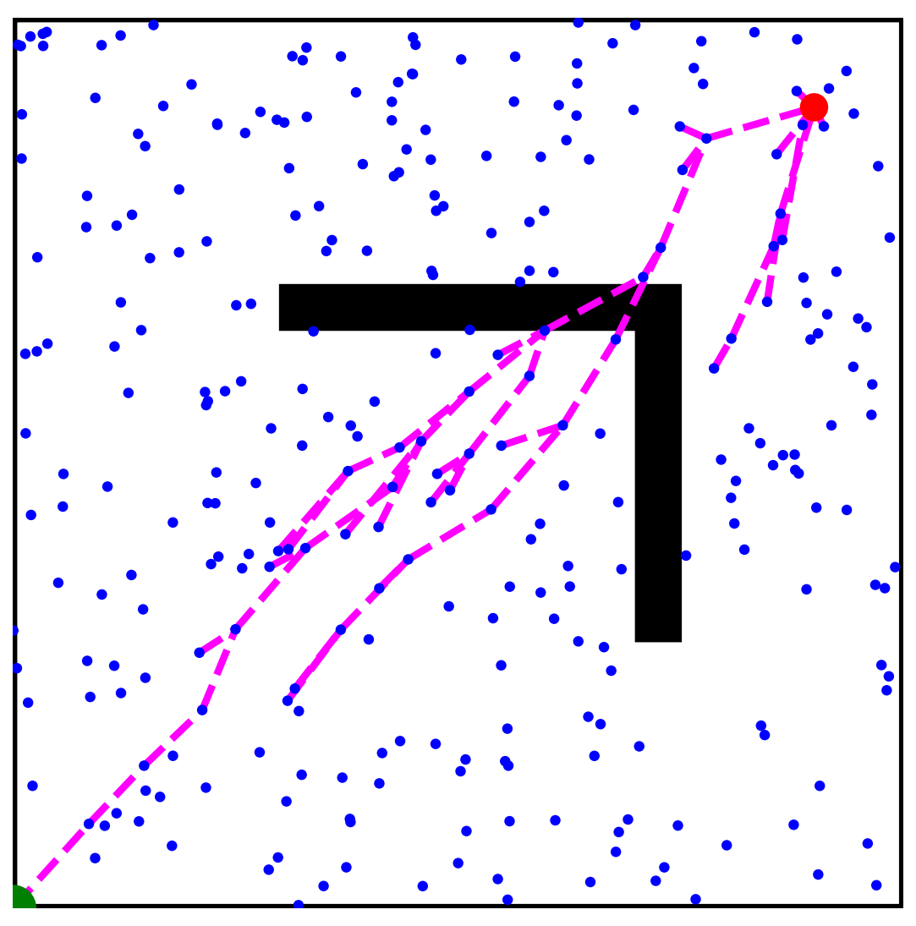
- Anytime, incremental bidirectional heuristic search (Bi-HS), ensuring meet-in-the middle property (MMP) and optimality, remains open

Contributions:

1. First anytime incremental bidirectional heuristic search (AI-Bi-HS).
2. First integration of AI-Bi-HS within Bw-SBMP.
3. Novel lazy edge evaluation strategy.

Computation Burden from Lazy Uni-HS

Lazy Uni-HS, omitting edge detection, may lead to an invalid path, and it requires to either restart from scratch or repair the invalid trees. Repeating this within a batch will incur intensive nearest neighbor access, updating its internal data structures (e.g., reconnected edges, recalculated costs, rewiring steps). This incurs a significant computational burden.



● valid informed state, — obstacle, (— —) unidirectional lazy search tree

Open Question

Why still open:

1. Achieving MMP took over 50 years due to key challenges such as proving optimality, frontier crossing, missing frontiers, etc.
2. **MM** algorithm incurs high computational due to requiring to access the minimal g-value and f-value at each iteration

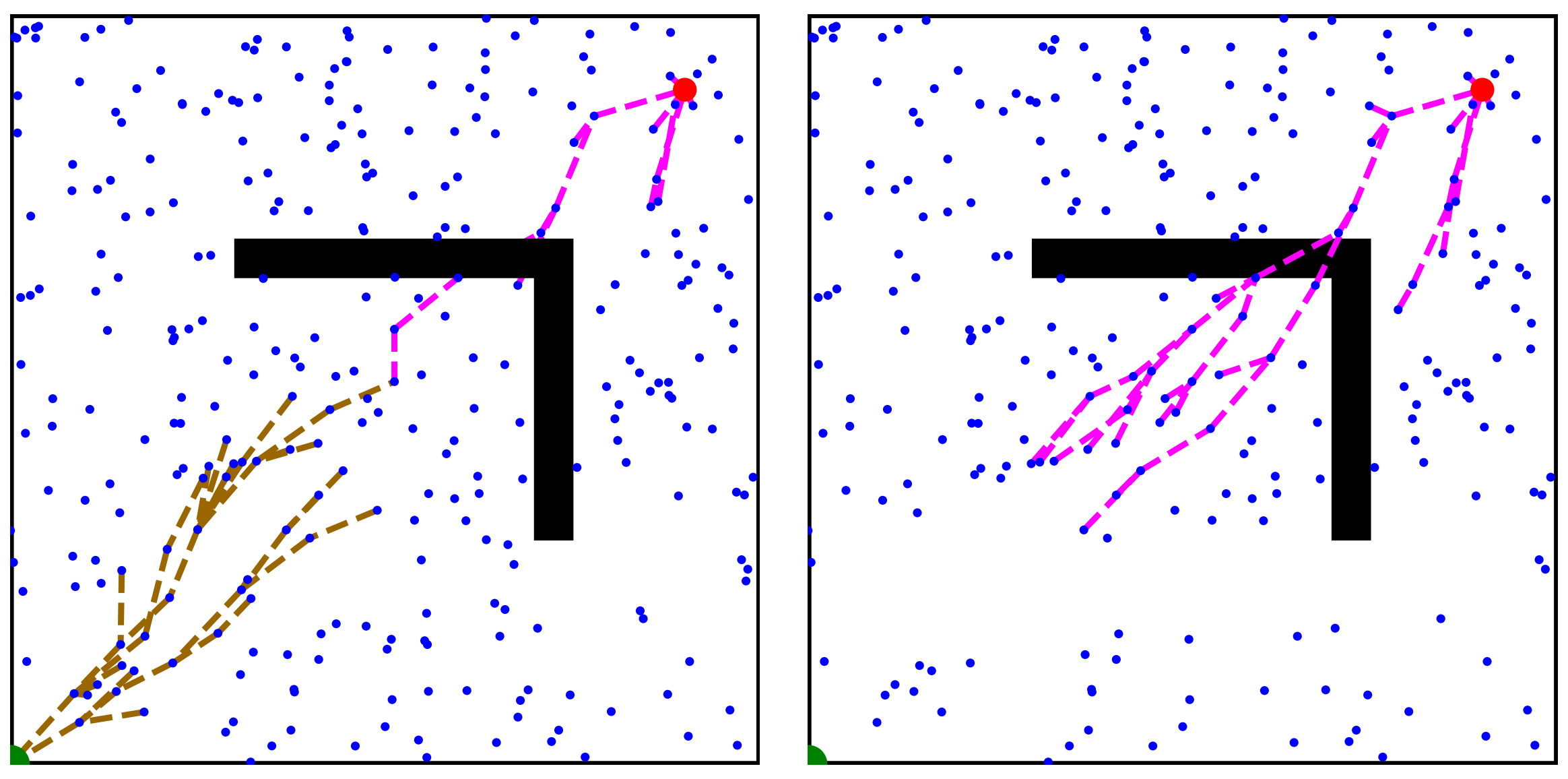
Comparison:

Concept	MM	BLIT*
First In Class	Yes	Yes
Adapted to Invalid Graph	No	Yes
Lazy Edge Evaluation	No	Yes
Path Generation	$s \in \mathrm{Open}_{\mathcal{F}} \cap \mathrm{Open}_{\mathcal{B}}$	$s \in Q_{\mathcal{F}} \cap \mathcal{T}_{\mathcal{B}}$, $\mathcal{T}_{\mathcal{B}} = Q_{\mathcal{B}} \cup V_{\mathcal{B}}$
State Priority Function	$\max(f, 2g)$	f with adaptive h
Admissible heuristic	Yes	Yes
Termination Condition	Access g_{min}-and f_{min}-values	only f-value
MMP and Optimality	Yes	Yes

f, g, h: estimated total solution cost, cost-to-come, and cost-to-go, $\mathrm{f} = \mathrm{g} + \mathrm{h}$, $\mathcal{T}$: motion tree, Open/Q: states to be explored next, V: states already expanded, $\mathcal{F}$: Forward direction, $\mathcal{D}$: Backward direction

Computational Opportunity

1 A candidate path (π): forward and backward paths, leading to repair or restart in solution-depth/2, or only needing operation on half tree.
2 Insufficient samples incur $\nexists\pi$, Bi-HS can efficiently detect $\nexists\pi$



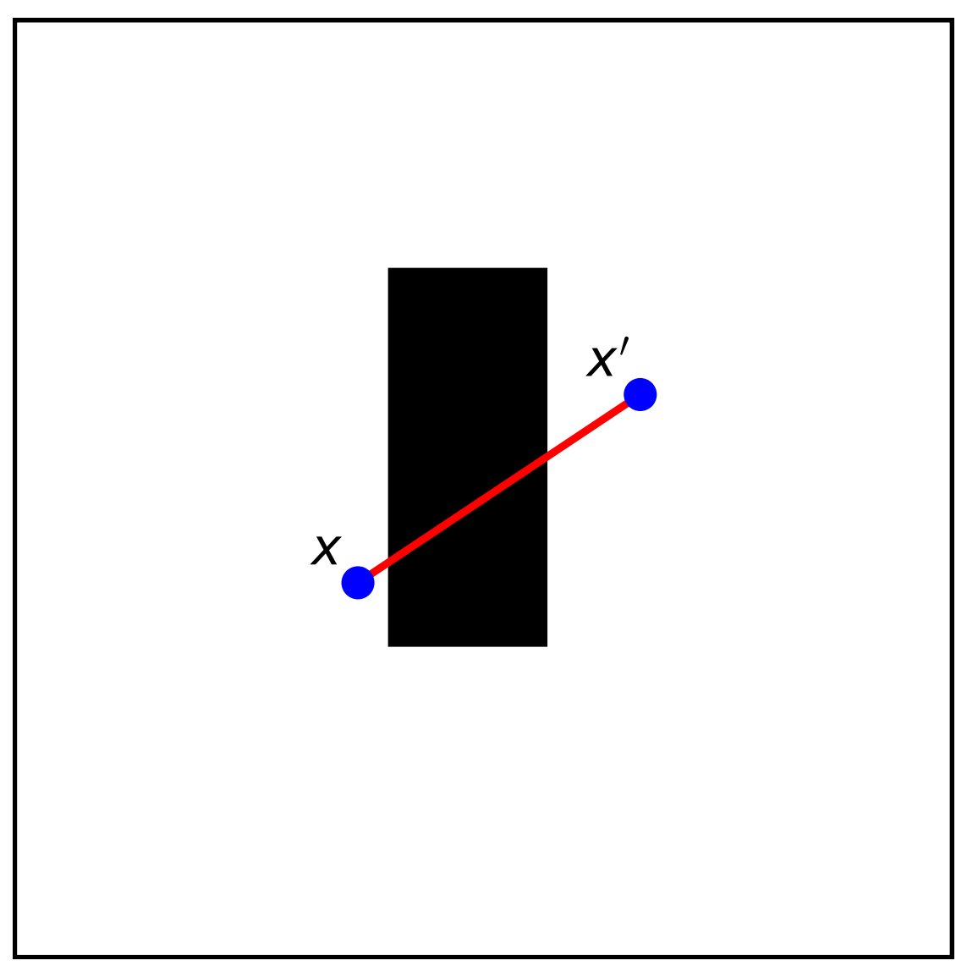
(a) Lazy Bi-HS (b) Insufficient Samples

(— —) backward lazy tree, (— —) forward lazy tree.

Key Inonvations:

Key Innovations:

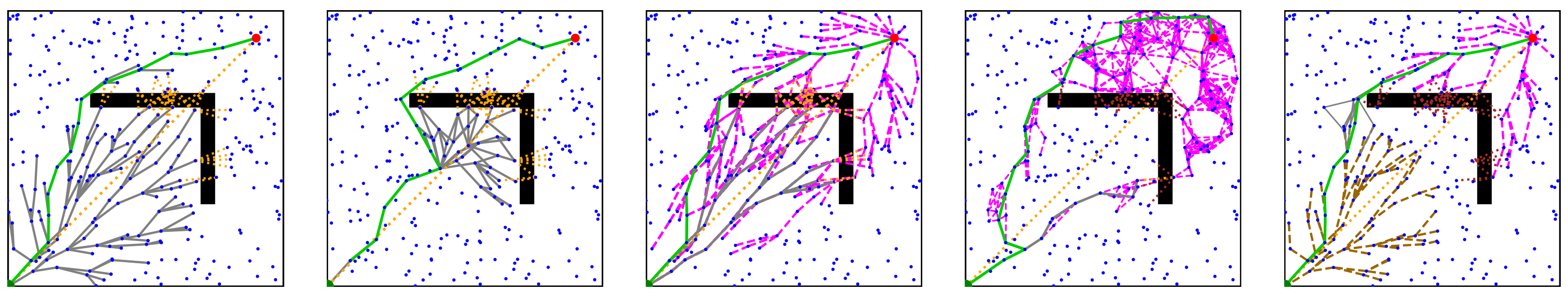
1. Refining heuristic *on-the-fly*: If a state $s \notin \mathcal{T}_{\mathcal{B}}$, $\widehat{h}_{\mathcal{F}}(s) := \widehat{g}_{\mathcal{F}}(s)$. If $s \in \mathcal{T}_{\mathcal{B}}$, $\widehat{h}_{\mathcal{F}}(\mathbf{s}) \leftarrow \min(\max(\widehat{h}_{\mathcal{F}}(\mathbf{s}), \widehat{g}_{\mathcal{F}}(\mathbf{s})), \widehat{g}_{\mathcal{B}}(\mathbf{s}))$
2. Focusing computational cost on states near obstacles



Complete Collision Detection (C-CD): (i) An edge on π, or (ii) A state labeled as a child on an invalid edge.
Sparse collision detection (S-CD): (i) A promising edge before finding the first valid π, (ii) After finding such π, only for states labeled near obstacles.

Illustration of Planners

Performances of BIT* , ABIT*, AIT*, EIT*, and BLIT* operating on a simulated $\mathbb{R}^2$ domain. Each planner, sourced from OMPL, runs to find its initial solution with 300 random states per batch (●). BLIT* finds the current best solution faster ($t = 1.1$ ms) than BIT* ($t = 2.2$ ms), ABIT* ($t = 1.2$ ms), AIT* ($t = 25.9$ ms), and EIT* ($t = 5.3$ ms).



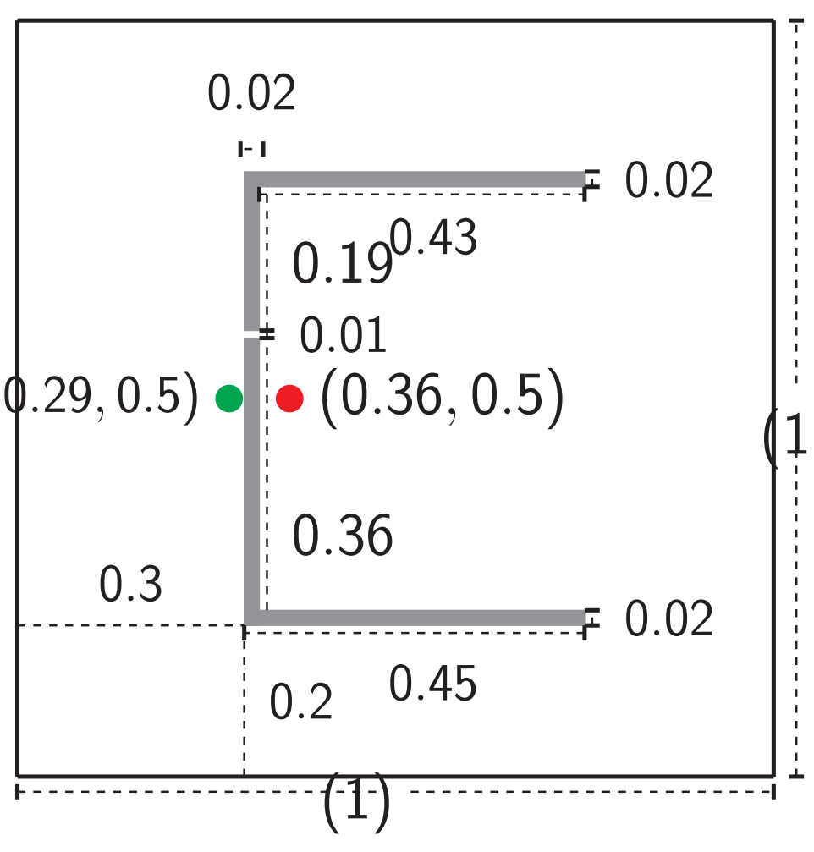
(c) BIT* (d) ABIT* (e) AIT* (f) EIT* (g) BLIT*

BLIT* ($cost = 2.9$), BIT* ($cost = 2.9$), ABIT* ($cost = 3.2$), AIT* ($cost = 2.9$), and EIT* ($cost = 3.5$).

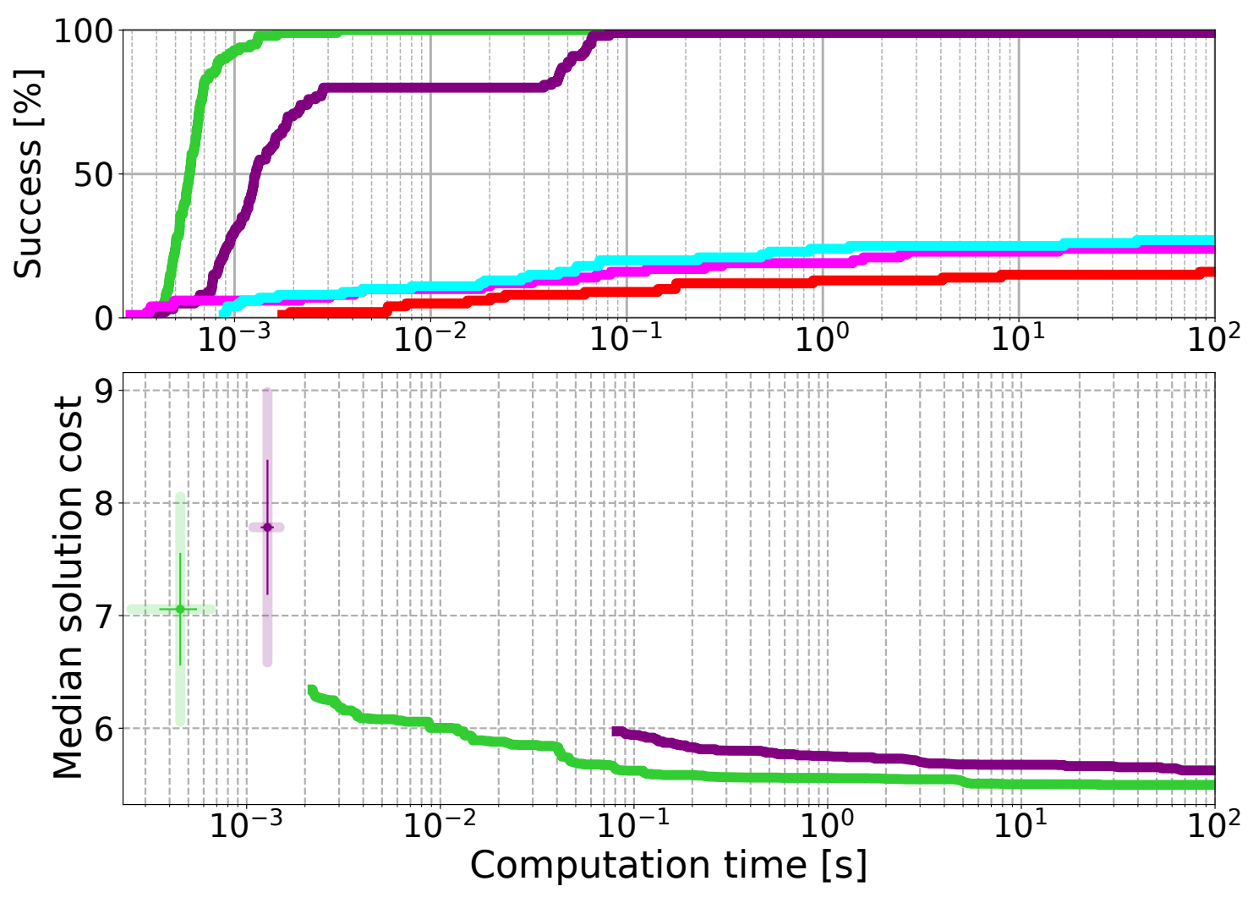
(—) valid path, (—) valid edge, (▪▪▪) invalid edge (C-CD), (▪▪▪) invalid edge (S-CD), (— —) forward lazy tree, (— —) backward lazy tree, ● start state, ● start state.

Experimental Results

Goal Enclosure with Wall Gap in $\mathbb{R}^{16}$: The start and goal are close together, which leads to many invalid edges, a highly inaccurate predefined heuristic, and a narrow gap in the wall that makes it particularly difficult for random sampling-based planners.

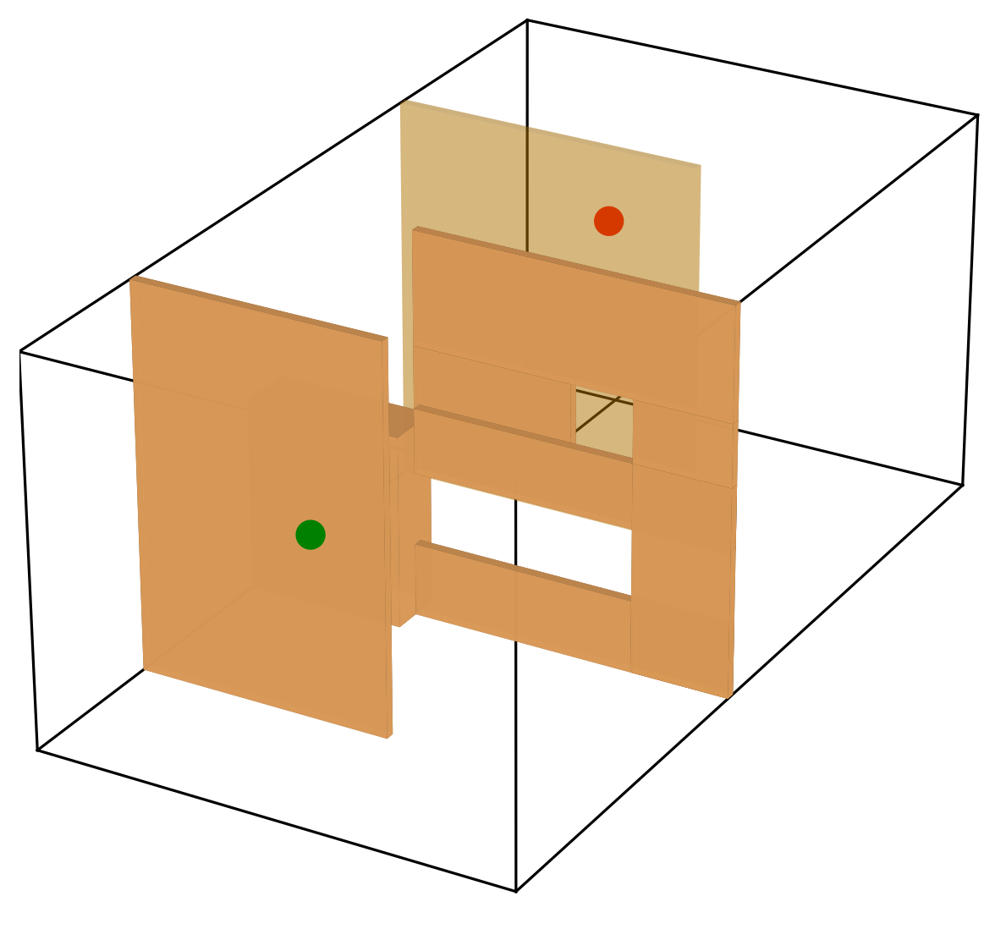


(h) $\mathbb{R}^{16}$

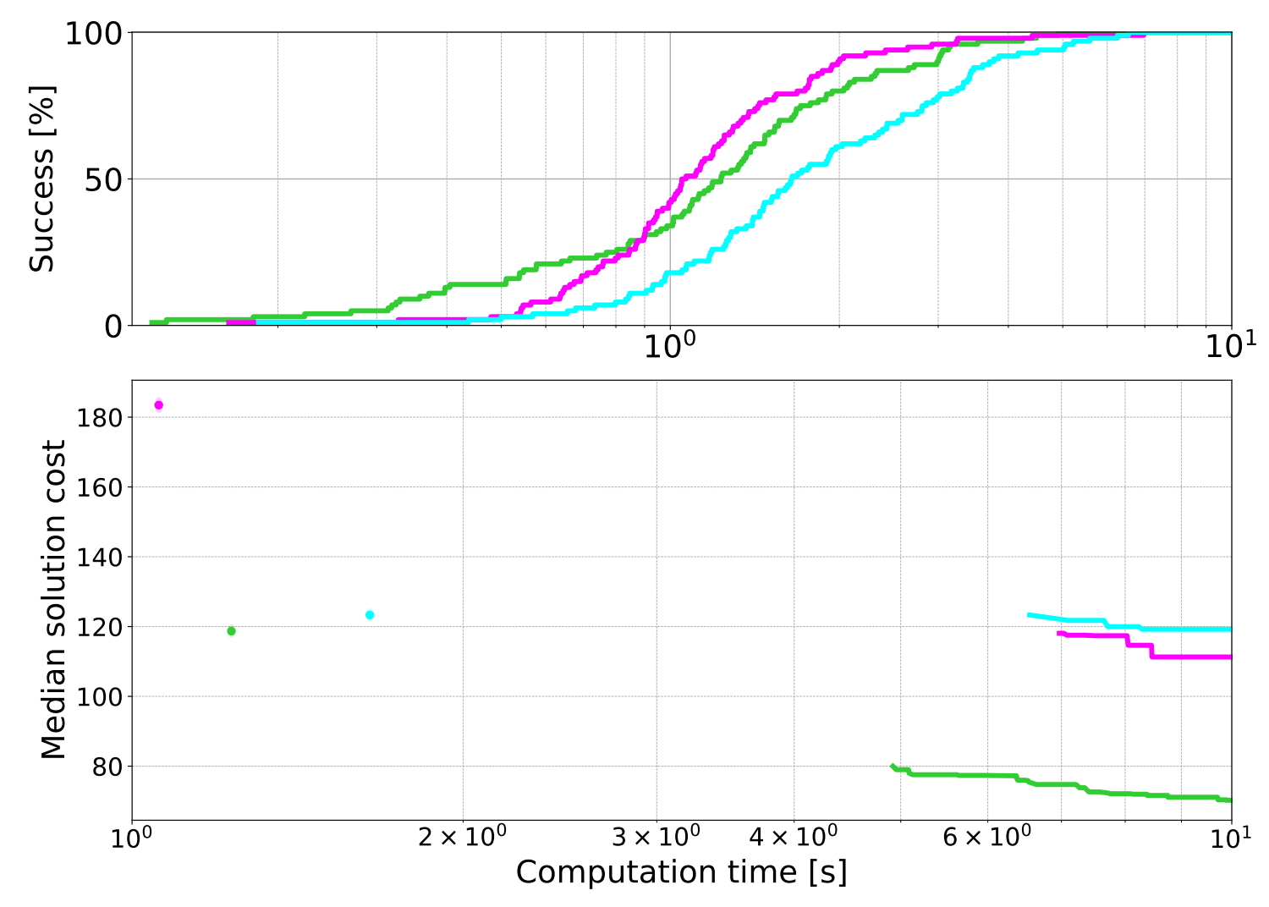


(i) $\mathbb{R}^{16}$

LAB with 10D Linearized Quadrotor (LQ): It features narrow, non-convex passages that create bottlenecks, making sampling more challenging and rendering the heuristic inaccurate. This modification presents a significant computational challenge as RGG becomes increasingly dense over time.



(j) 10D LQ



(k) 10D LQ

— BIT* — ABIT* — AIT* — EIT* — BLIT*