

Pathfinding Algorithms for Grid & NavMesh

Author: **Jiaming Liu** — 301626723

Introduction

I built an interactive pathfinding demo to compare **grid-based** search and a **simplified NavMesh** approach. Users can place obstacles (grid mode), set start/goal, pick algorithms, and watch agents animate their path. Metrics measure speed, nodes explored, and path quality. The goal is to show how different planners trade off optimality, efficiency, and movement realism.

Demo: <https://766.jiamingliu.com/>

Repo: <https://github.com/Jiaaming/Pathfinding>

Related Work

- **A***: classic optimal graph search with heuristics.
- **Dijkstra**: uniform-cost baseline.
- **Greedy Best-First**: heuristic-only, fast but not always optimal.
- **Jump Point Search (JPS)**: A* pruning for uniform grids.
- **Theta***: any-angle extension of A*.
- **Rapidly-exploring Random Trees (RRT)**: sampling-based planning.
- **NavMesh**: industry standard to abstract walkable space into a sparse graph for efficient search and funnel smoothing.

Algorithm / Method

Grid mode

- Representation: 2D grid, 4/8-neighbor moves, obstacles block cells.
- Algorithms: A*, Dijkstra, Greedy, JPS, Theta*, RRT.
- Costs: unit step (or Euclidean for any-angle), heuristic is Manhattan for A* and Greedy.

NavMesh mode (simplified)

- Representation: merge walkable cells into rectangles (polygons); shared edges are portals; polygon centers are nodes.
- Algorithms: A*, Dijkstra, Greedy on the polygon adjacency graph; NavMesh RRT samples polygon centers/goal.
- Funnel: portal sequence → funnel/string pulling → straightened path; then densified for animation.
- Costs: edge cost = distance from polygon center to portal midpoint; heuristic = center-to-goal distance; path cost = Euclidean length of the funnel path.

Implementation

- Frontend: HTML5 canvas for grid/NavMesh visualization and agent animation.
- Grid tools: add obstacles, set agents, pick algorithms, start/pause/reset.

- NavMesh: obstacle editing disabled; shows polygons and centers; start/goal selection only.
- Visualization: paths as dashed lines; explored nodes shaded in light agent color; polygon centers always visible (green).
- Metrics: computation time (ms), nodes explored, path cost/length.

Evaluation

Example comparison:

Algorithm Performance Comparison					
Agent 1		A* Search	Agent 2		JPS (Jump Point)
Computation Time		1.600 ms	Computation Time		0.800 ms
Nodes Explored		424	Nodes Explored		26
Path Length		48 steps	Path Length		38.6 steps
Agent 3		Theta*	Agent 4		Dijkstra
Computation Time		1.200 ms	Computation Time		1.700 ms
Nodes Explored		188	Nodes Explored		998
Path Length		38.1 steps	Path Length		48 steps
Agent 5		Greedy Best-First	Agent 6		RRT
Computation Time		0.200 ms	Computation Time		0.400 ms
Nodes Explored		112	Nodes Explored		56
Path Length		48 steps	Path Length		51.8 steps

- **A*** is a solid baseline: optimal but expands many nodes.
- **JPS** is the **fastest optimal** algorithm on grids — huge reduction in node expansions (only 26 vs A*'s 424).
- **Theta*** yields more **natural and shorter** any-angle paths.
- **Dijkstra** searches blindly → **largest** node expansions and slowest runtime.
- **Greedy** is extremely fast but risks lower-quality solutions.
- **RRT** finds feasible paths in tricky maps but is **not optimal** and path cost varies.

Summary:
JPS for optimal + fast grid search
Theta* for smoother, realistic grid movement

Limitations

- NavMesh uses rectangle merging instead of full polygon decomposition.
- All geometry still derived from a grid → not continuous collision.
- RRT lacks bias/rewiring → not optimal like RRT*.

Acknowledgements

Thanks to open literature on A*, Dijkstra, JPS, Theta*, RRT, and NavMesh funnel methods, and course staff for guidance.

Key References (≤ 5)

1. Hart, Nilsson, Raphael. "A Formal Basis for the Heuristic Determination of Minimum Cost Paths." IEEE TSSC, 1968.
2. Dijkstra. "A Note on Two Problems in Connexion with Graphs." Numerische Mathematik, 1959.
3. Harabor & Grastien. "Online Graph Pruning for Pathfinding on Grid Maps (Jump Point Search)." AAAI, 2011.
4. Nash, Koenig, Tovey. "Any-Angle Path Planning." AAAI, 2009.
5. LaValle. "Rapidly-Exploring Random Trees: A New Tool for Path Planning." Technical Report, 1998.