Wikipedia
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

Languages
Add links

Jump point search

From Wikipedia, the free encyclopedia

In computer science, **jump point search** is an optimization to the A* search algorithm pathfinding algorithm for uniform-cost grids. It reduces symmetries in the search procedure by means of graph pruning,[1] eliminating certain nodes in the grid based on assumptions that can be made about the current node's neighbors, as long as certain conditions relating to the grid are satisfied. As a result, the algorithm can consider long "jumps" along straight (horizontal, vertical and diagonal) lines in the grid, rather than the small steps from one grid position to the next that ordinary A* considers.[2]

Jump point search preserves A*'s optimality, while potentially reducing its running time by an order of magnitude.[1]

Graph and tree search algorithms

α–β · A* · B* · Backtracking · Beam · Bellman–Ford · Best-first · Bidirectional · Borůvka · Branch & bound · BFS · British Museum · D* · DFS · Depth-limited · Dijkstra · Edmonds · Floyd–Warshall · Fringe search · Hill climbing · IDA* · Iterative deepening · Johnson · **Jump point** · Kruskal · Lexicographic BFS · Prim · SMA*

Listings

Graph algorithms · *Search algorithms* · *List of graph algorithms*

Related topics

Dynamic programming · Graph traversal · Tree traversal · Search games

v · t · e

History [edit]

Harabor and Grastien's original publication provides algorithms for neighbour pruning and identifying successors.[1] The original algorithm for neighbour pruning allowed corner-cutting to occur, which meant the algorithm could only be used for moving agents with zero width; limiting its application to either real-life agents (e.g. robotics) or simulations (e.g. many games).

The authors presented modified pruning rules for applications where corner-cutting is not allowed the following year.[3] This paper also presents an algorithm for pre-processing a grid in order to minimise online search times.

A number of further optimisations were published by the authors in 2014.[4]

All the published modifications and optimisations preserve A* optimality.

References [edit]

1. ^ *a b c* D. Harabor; A. Grastien (2011). *Online Graph Pruning for Pathfinding on Grid Maps* (PDF). 25th National Conference on Artificial Intelligence. AAAI.
2. ^ Witmer, Nathan (5 May 2013). "Jump Point Search Explained". *zerowidth positive lookahead*. Retrieved 9 March 2014.
3. ^ D. Harabor; A. Grastien (2012). *The JPS Pathfinding System*. 26th National Conference on Artificial Intelligence. AAAI.
4. ^ Harabor, Daniel; Grastien, Alban. "Improving Jump Point Search" (PDF). *Australian National University College of Engineering and Computer Science*. Association for the Advancement of Arti- ficial Intelligence (www.aaai.org). Retrieved 11 July 2015.

This algorithms or data structures-related article is a stub. You can help Wikipedia by expanding it.

Categories: Game artificial intelligence | Graph algorithms | Search algorithms | Algorithms and data structures stubs | Computer science stubs