



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  
Čeština  
فارسی  
Français  
Հայերեն  
Српски / srpski  
ไทย Edit links

[Create account](#) [Log in](#)

Article [Talk](#)

[Read](#) [Edit](#) [View history](#)

Search Q

# Beam search

From Wikipedia, the free encyclopedia

In [computer science](#), **beam search** is a [heuristic search](#) algorithm that explores a graph by expanding the most promising node in a limited set. Beam search is an optimization of [best-first search](#) that reduces its memory requirements. Best-first search is a graph search which orders all partial solutions (states) according to some heuristic which attempts to predict how close a partial solution is to a complete solution (goal state). But in beam search, only a predetermined number of best partial solutions are kept as candidates.<sup>[1]</sup>

Beam search uses [breadth-first search](#) to build its [search tree](#). At each level of the tree, it generates all successors of the states at the current level, sorting them in increasing order of heuristic cost.<sup>[2]</sup> However, it only stores a predetermined number of best states at each level (called the beam width). Only those states are expanded next. The greater the beam width, the fewer states are pruned. With an infinite beam width, no states are pruned and beam search is identical to [breadth-first search](#). The beam width bounds the memory required to perform the search. Since a goal state could potentially be pruned, beam search sacrifices completeness (the guarantee that an algorithm will terminate with a solution, if one exists). Beam search is not optimal (that is, there is no guarantee that it will find the best solution). It returns the first solution found.

The beam width can either be fixed or variable. One approach that uses a variable beam width starts with the width at a minimum. If no solution is found, the beam is widened and the procedure is repeated.<sup>[3]</sup>

**Contents** [[hide](#)]

- [1 Name](#)
- [2 Uses](#)
- [3 Extensions](#)
- [4 References](#)

## Name [[edit](#)]

The term "beam search" was coined by [Raj Reddy](#), [Carnegie Mellon University](#), 1976.

## Uses [[edit](#)]

A beam search is most often used to maintain tractability in large systems with insufficient amount of memory to store the entire search tree.<sup>[4]</sup> For example, it is used in many [machine translation](#) systems.<sup>[5]</sup> To select the best translation, each part is processed, and many different ways of translating the words appear. The top best translations according to their sentence structures are kept, and the rest are discarded. The translator then evaluates the translations according to a given criterion, choosing the translation which best keeps the goals. The first use of a beam search was in the Harpy Speech Recognition System, CMU 1976.<sup>[6]</sup>

## Extensions [[edit](#)]

Beam search has been made [complete](#) by combining it with [depth-first search](#), resulting in [beam stack search](#)<sup>[7]</sup> and *depth-first beam search*,<sup>[4]</sup> and with limited discrepancy search,<sup>[8]</sup> resulting in *beam search using limited discrepancy backtracking*<sup>[4]</sup> (BULB). The resulting search algorithms are [anytime algorithms](#) that find good but likely sub-optimal solutions quickly, like beam search, then backtrack and continue to find improved solutions until convergence to an optimal solution.

## References [[edit](#)]

### 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

1. <sup>^</sup> [beam search from FOLDOC](#)
2. <sup>^</sup> [British Museum Search](#)
3. <sup>^</sup> Norvig, Peter. *Paradigms of Artificial Intelligence*. [http://books.google.com/books?id=X4mhySvjqUAC&pg=PA196&lpg=PA196&dq=beam+search+in+paradigms+of+artificial+intelligence+programming&source=web&ots=20BH2IB3sc&sig=udBIO2Qeg8awaySgwCu7gDEObY4&hl=en&sa=X&oi=book\\_result&resnum=1&ct=result#PPA196,M1](http://books.google.com/books?id=X4mhySvjqUAC&pg=PA196&lpg=PA196&dq=beam+search+in+paradigms+of+artificial+intelligence+programming&source=web&ots=20BH2IB3sc&sig=udBIO2Qeg8awaySgwCu7gDEObY4&hl=en&sa=X&oi=book_result&resnum=1&ct=result#PPA196,M1) [Programming](#)
4. <sup>^</sup> <sup>a</sup> <sup>b</sup> <sup>c</sup> Furcy, David. Koenig, Sven. "Limited Discrepancy Beam Search". 2005. <http://www.ijcai.org/papers/0596.pdf> 
5. <sup>^</sup> Tillmann, Christoph. Ney, Hermann. "Word Reordering and a Dynamic Programming Beam Search Algorithm for Statistical Machine Translation". <http://acl.ldc.upenn.edu/J/J03/J03-1005.pdf> 
6. <sup>^</sup> Lowerre, Bruce. "The Harpy Speech Recognition System", Ph.D. thesis, Carnegie Mellon University, 1976
7. <sup>^</sup> Zhou, Rong. Hansen, Eric. "Beam-Stack Search: Integrating Backtracking with Beam Search". 2005. <http://www.aaai.org/Library/ICAPS/2005/icaps05-010.php>
8. <sup>^</sup> [CiteSeerX: 10.1.1.34.2426](#)

Categories: [Search algorithms](#)

This page was last modified on 13 July 2015, at 22:57.

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.

[Privacy policy](#) [About Wikipedia](#) [Disclaimers](#) [Contact Wikipedia](#) [Developers](#) [Mobile view](#)

