УДК 811.111:004.92:004.4

ITERATIVE DEEPENING A* SEARCH ALGORITHM

Kvachenyuk G. V., student
Scientific supervisor – Vanik I. Y., senior lecturer
English language department № 1
Belarusian National University of Technology
Minsk, Republic of Belarus

The IDA* (Iterative Deepening A*) algorithm is a search algorithm that combines the advantages of both depth-first search and A* search. In this paper, we provide a comprehensive overview of the IDA* algorithm, its underlying principles, advantages, and applications. We also discuss its implementation, optimization techniques, and compare it with other search algorithms [1].

Search algorithms are fundamental in solving various computational problems, particularly in artificial intelligence and computer science. One such algorithm is IDA*, which is an extension of both depth-first search (DFS) and A* search algorithms. It was introduced by Richard Korf in 1985 as a memory-efficient alternative to A* search, while still maintaining completeness and optimality.

The IDA* algorithm is an iterative deepening variant of the A* search algorithm. It performs all operations that A Star does, but it takes less memory [1]. The IDA* algorithm works by gradually increasing a threshold value until a solution is found. At each iteration, it performs a depth-first search limited by the current threshold. If a solution is not found within the threshold, the threshold is increased, and the search continues. This process repeats until a solution is found. Next, the implementation of pseudocode is provided.

Pseudocode realization:

```
function IDA*(node, cost, threshold)
f=cost+heuristic(node)
if f>threshold
return f
if node is goal
return FOUND
min_cost=∞
for each successor of node
```

```
new_cost=cost+distance(node, successor)
result=IDA*(successor, new_cost, threshold)
if result==FOUND
return FOUND
if result<min_cost
min_cost=result
return min_cost
```

Several optimizations can be applied to improve the efficiency of IDA*. These include using iterative deepening, memory pruning techniques, and efficient heuristic functions tailored to the problem domain [2]. IDA* offers several advantages over traditional search algorithms. Unlike A* search, IDA* does not require storing all generated nodes, making it memory-efficient. Additionally, it guarantees optimality without the need for extra memory. However, it may explore the same nodes multiple times, leading to potentially higher time complexity compared to A* search [3].

IDA* has been successfully applied in various domains, including puzzle solving, route planning, and optimization problems. Its memory-efficient nature makes it suitable for resource-constrained environments, such as robotics and embedded systems.

In conclusion, the IDA* algorithm provides an efficient and memory-conscious solution to search problems. By iteratively deepening the search space and utilizing heuristics, it strikes a balance between completeness, optimality, and resource utilization. Its versatility and effectiveness make it a valuable tool in the arsenal of search algorithms.

References

1. Iterative deepening A* algorithm (IDA*) – Artificial Intelligence [Electronic resource] – Mode of access: https://www.geeksforgeeks.org/iterative-deepening-a-algorithm-ida-artificial-intelligence/. – Date of access: 03.04.2024.

2. Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths // IEEE Transactions on Systems Science and Cybernetics, 1968. Vol. 4 (2), P. 100–107.

3. Korf, R. E. (1985). Depth-first iterative-deepening: An optimal admissible tree search // Artificial Intelligence, Vol. 27 (1), P. 97–109.