

CSE 486 A

Emil Sayahi

1st September 2023

Lecture notes from the 2023 undergraduate course ‘Introduction to Artificial Intelligence’, given by Professor Khodakhast Bibak at Miami University at Benton Hall in the academic year 2023-2024. This course covers introductory artificial intelligence concepts. Credit for the material in these notes is due to Professor Khodakhast Bibak, while the structure is loosely taken from the in-class lectures. The credit for the typesetting is my own.

Disclaimer: This document will inevitably contain some mistakes—both simple typos and legitimate errors. Keep in mind that these are the notes of an undergraduate student in the process of learning the material, so take what you read with a grain of salt. If you find mistakes and feel like telling me, I will be grateful and happy to hear from you, even for the most trivial of errors. You can reach me by email, in English, at sayahie@miamioh.edu.

This work is licensed under a [Creative Commons](https://creativecommons.org/licenses/by-nc-sa/4.0/) “Attribution-NonCommercial-ShareAlike 4.0 International” license.



For more notes like this, visit [my GitHub profile](#).

Emil Sayahi,
Fall Term: 2023,
Last Update: 1st September 2023,
Miami University

Contents

| | |
|--|----------|
| Lecture 1: Week 1, Wednesday | 1 |
| 1.1 Uninformed Search | 1 |
| Lecture 2: Week 1, Friday | 2 |
| 2.1 Breadth-first search (BFS) | 2 |
| 2.2 Uniform-cost search (UCS) | 3 |

Wed, 30 August 2023, 11:40am – 1:00pm

Lecture 1: Week 1, Wednesday

1.1 Uninformed Search

Many AI tasks can be formulated as search problems; the goal is to find a *sequence of actions*.

- Puzzles
- Games
- Navigation
- Assignment
- Motion planning
- Scheduling
- Routing

Fri, 1 September 2023, 11:40am – 1:00pm

Lecture 2: Week 1, Friday

Definition 2.1

Uninformed search is a search strategy that uses no problem-specific knowledge. Only the goal test and the successor function are used; the **successor function** generates all possible states. It is not known which non-goal states are better than others. Strategies that know whether one non-goal state is better than another are referred to as **informed search** or **heuristic search** strategies.

There are five major types of uninformed search strategies:

- Breadth-first search (BFS)
- Uniform-cost search (UCS)
- Depth-first search (DFS)
- Depth-limited search (DLS)
- Iterative deepening search (IDS)

All of these uninformed search strategies are distinguished by the *order* in which nodes are expanded.

2.1 Breadth-first search (BFS)

Breadth-first search operates level-by-level, expanding all nodes at a given level before expanding any nodes at the next level. On a given level, nodes are expanded from left to right by convention.

Definition 2.2

Breadth-first search is implemented using a **first in, first out (FIFO) queue**. The FIFO queue is a data structure that supports two operations: **enqueue** and **dequeue**. The enqueue operation adds an element to the end of the queue, and the dequeue operation removes an element from the front of the queue.

Definition 2.3

If a solution exists, breadth-first search will find it in finite time, provided that the branching factor is finite and the depth of the solution is finite; this means that breadth-first search is **complete**. Breadth-first search is not always **optimal**, however, as the solution found may not have the minimum cost. It is optimal when all edges have the same cost, no cost, or when the cost is a non-decreasing function of the depth of the node.

Definition 2.4

The **time complexity** of an algorithm is the number of steps required to solve a problem of size n , where n is the size of the input; in the worst-case of breadth-first search, the goal node would be the very last node explored (ie, every vertex and edge is explored; $O(|V| + |E|)$). The **space complexity** of an algorithm is the maximum amount of memory required to solve a problem of size n ; in the worst-case of breadth-first search, the goal node is discovered after all vertices are explored & stored in memory $O(|V|)$.

Definition 2.5

Complexity is expressed in terms of three quantities:

- b is the **branching factor**, or the maximum number of children, or 'successors', of any node.
- d is the **depth** of the shallowest (ie, closest to the root) goal node.
- m is the **maximum length** of any path in the state space.

The time and space complexity of breadth-first search is exponential, $O(b^d)$, where b is the branching factor and d is the depth of the shallowest goal node. This is because the number of nodes expands exponentially with the depth of the tree.

2.2 Uniform-cost search (UCS)

Uniform-cost search expands the node n with the *lowest* path cost $g(n)$ instead of expanding the shallowest node, where $g(n)$ returns the cost of the path from the starting node, s , to the current node, n . This is also referred to as 'Dijkstra's algorithm'. This algorithm uses a priority queue to order nodes in the frontier list by path cost, with the lowest cost node at the front of the queue.

Notes