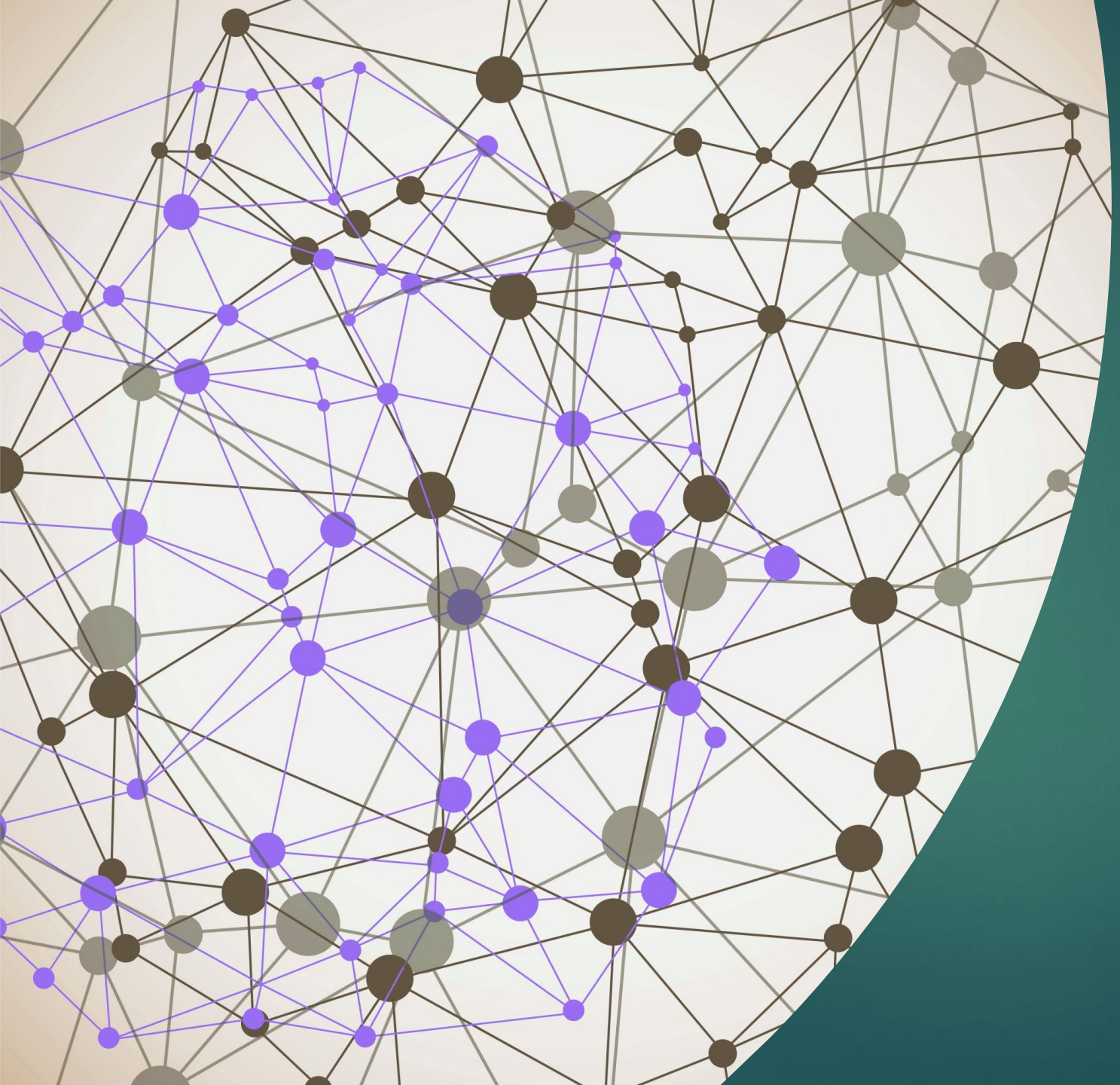


Assignment 2: Analysis

BY ELIJAH MON - 40078229

The Problem: S-Puzzle

- The state space for this puzzle scale greatly as S increases {(S+1)!}
- The modified version of the S-Puzzle that we are using for this assignment has no blank space. Because any of the tiles can swap places with any other adjacent tiles, The branching factor of the search is significantly larger than a standard S-Puzzle. Many of these branches/paths, however, are redundant.

Puzzle Solving Algorithms

Uninformed Searches

- Depth First Search
- Iterative Deepening Depth Limited Search

Informed Searches

- A* (Hamming Distance)
- A* (Manhattan Distance)

Depth First Search

- Depth first search is not a good algorithm for this problem
- Depending on the implementation, the entire search space might have to be explored to return a solution
- Depending on the implementation, it is possible for the algorithm to enter infinite cycles or traverse an infinitely long path
- DFS is complete in finite search spaces but incomplete in infinite search spaces

Report

AVERAGE EXECUTION TIME: 51.0000500202179

TOTAL EXECUTION TIME: 1020.0010004043579

AVERAGE COST: 4359.3

TOTAL COST: 87186

AVERAGE PATH LENGTH: 4359.3

TOTAL PATH LENGTH: 87186

AVERAGE NO SOLUTION: 85.0

TOTAL NO SOLUTION: 17

Iterative Deepening Depth Limited Search

- A step in the right direction
- This algorithm combines elements of DFS and BFS
- Complete on finite acyclic state spaces or finite state spaces where cycles are checked

Report

AVERAGE EXECUTION TIME: 8.4484517216682 44

TOTAL EXECUTION TIME: 168.96903443336487

AVERAGE COST: 3.05

TOTAL COST: 61

AVERAGE PATH LENGTH: 3.05

TOTAL PATH LENGTH: 61

AVERAGE NO SOLUTION: 10.0

TOTAL NO SOLUTION: 2

A* Algorithm

- A* Search is complete
- Total search cost is lower than the uninformed strategies which translates to a faster search in for this puzzle
- There exists a computational overhead to determine h(n),f(n) and consequently f(n)

Heuristics
Cost-optimality is dependent on certain properties of the heuristics used (admissibility, consistency, etc…)

- Hamming Distance
- Manhattan Distance

A* - Hamming Distance

Report

- AVERAGE EXECUTION TIME: 0.015422630310058593
- TOTAL EXECUTION TIME: 0.3084526062011719
- AVERAGE COST: 5.2
- TOTAL COST: 104
- AVERAGE PATH LENGTH: 2.35
- TOTAL PATH LENGTH: 47
- AVERAGE NO SOLUTION: 0.0
- TOTAL NO SOLUTION: 0

A* - Manhattan Distance

Report

- AVERAGE EXECUTION TIME: 0.015446221828460694
- TOTAL EXECUTION TIME: 0.30892443656921387
- AVERAGE COST: 5.2
- TOTAL COST: 104
- AVERAGE PATH LENGTH: 2.35
- TOTAL PATH LENGTH: 47
- AVERAGE NO SOLUTION: 0.0
- TOTAL NO SOLUTION: 0

Comparison

Uninformed Algorithms

DFS vs IDDLS

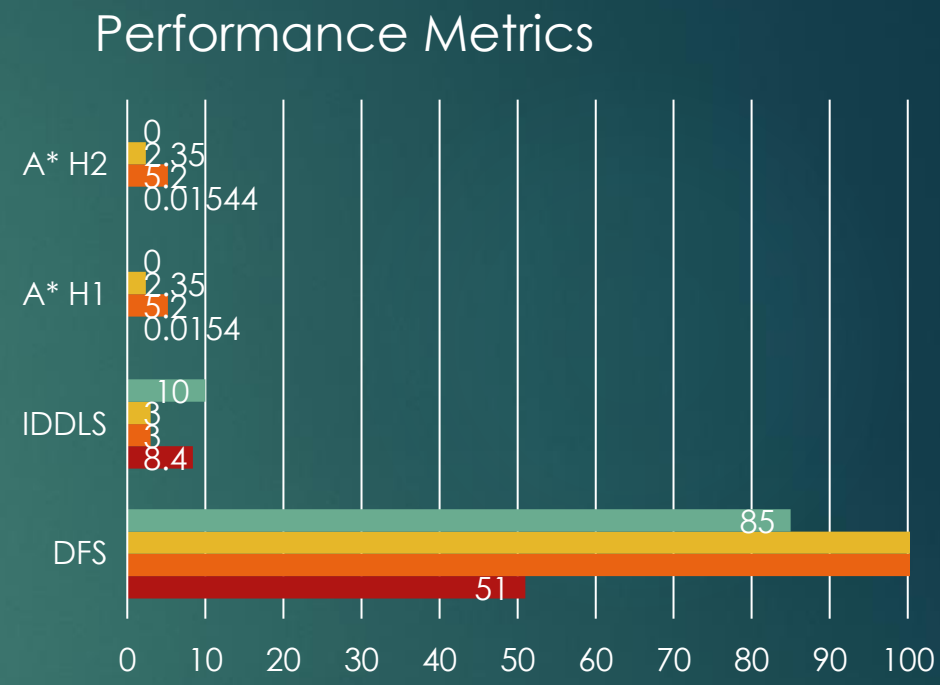
Informed vs Uninformed

A* vs IDDLS

Informed Algorithms

A* Heuristic 1 vs A* Heuristic 2

Note: This data was collected from 20 3x3 puzzle samples with at most 10 random permutations performed on each. The differences between the 3 best performing algorithms can be much better seen when scaling up the size of each puzzle. This is also likely true for more "difficult" sample states.



	DFS	IDDLS	A* H1	A* H2
Failure (%)	85	10	0	0
Average Path Length (depth)	4359	3	2.35	2.35
Average Cost	4359	3	5.2	5.2
Average Execution Time (s)	51	8.4	0.0154	0.01544

Improvements

To further improve the algorithms for this search problem, several changes can be made:

- Improved heuristic functions
- Modified A* algorithms such as IDA* or Bidirectional A*
- Making use or parallelism/multithreading on these modified algorithms

(Basel A. Mahafzah (2011) Parallel multithreaded IDA* heuristic search: algorithm design and performance evaluation, International Journal of Parallel, Emergent and Distributed Systems, 26:1, 61-82, DOI: 10.1080/17445761003604521)