N-Puzzle

Practical analysis of the problem

Luigi Seminara

Dipartimento di
Matematica e Informatica
Università degli studi di Catania, Italia

Academic Year 2022/2023

Problem

Description of the problem

The N-Puzzle, also known as the sliding tile puzzle, is a classic problem in artificial intelligence and cognitive psychology. The problem consists of a square board with $N^2 - 1$ numbered tiles and one blank space, where N is the dimension of the board.

Example with $N = 3$:

1	2	3
4	5	6
7	8	

Goal

Description of the goal

The goal of the problem is to rearrange the numbered tiles in ascending order using a series of sliding moves on the blank space.

Example with $N = 3$:

1	2	3
4		6
7	5	8

In this example we have to move the tile 5 and then the tile 8 to reach the goal.

1	2	3
4	5	6
7	8	

Table of contents

- **A* search**
- **BA* search**
- **Heuristics**:
  - Manhattan distance
  - Euclidean distance
  - Misplaced tiles
  - Linear conflict
  - Linear conflict + Manhattan distance
- **Experiments**:
  - **8-Puzzle** $\Rightarrow 3 \times 3$
  - **15-Puzzle** $\Rightarrow 4 \times 4$
  - **24-Puzzle** $\Rightarrow 5 \times 5$
  - **100-Puzzle** $\Rightarrow 10 \times 10$
- **Conclusion**

Informed Research

Definition

Informed research is a problem solving strategy that uses additional information about the environment and the problem. This information, called heuristics, is used to guide the search towards the regions of the search space most likely to contain a solution.

A* search

A* search is an informed search strategy that uses an evaluation function to determine the distance of the current node from the goal and the estimated cost to reach the goal through that node. The evaluation function of A* is defined as:

$$f(n) = g(n) + h(n)$$

where:

- $g(n)$ is the actual cost to reach node n (i.e. the cost of the solution so far);
- $h(n)$ is the estimated cost to reach the goal from node n (i.e. the heuristic).

A* search expands the node with the lowest $f(n)$ value first. This means that A* aims to find the optimal solution, i.e. the solution with the lowest cost. A* uses an admissible heuristic $h(n)$ that never overestimates the actual distance.

A* search

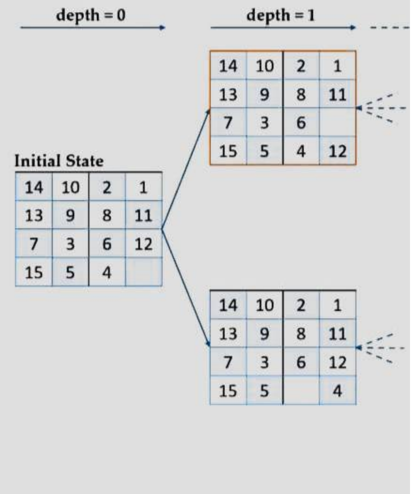

Figure: Operation of A*.

BA* search

Bidirectional A* Search (BA*) differs from the standard A* algorithm in that it searches in both directions, starting from both the initial state and the target state. This type of search is useful for solving search problems in graphs where the optimal solution is difficult to find using only one direction. BA* Search uses pre-existing information on both ends of the problem to find the solution more efficiently.

BA* search

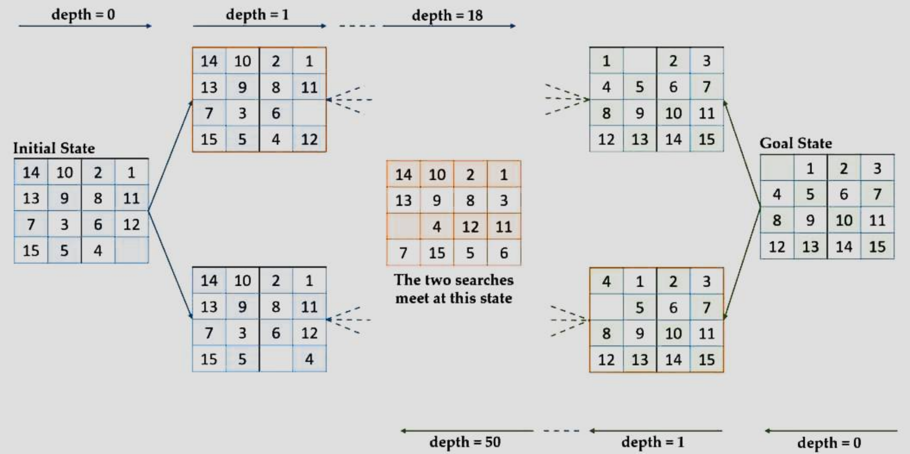

Figure: Operation of BA*.

A* vs BA*

Criterion	**A***	**BA***
Time Complexity	$O(b^d)$	$O(b^{d/2})$
Space Complexity	$O(b^d)$	$O(b^{d/2})$
Complete?	Yes	Yes
Optimal?	No	No

Heuristics

Manhattan distance

This heuristic calculates the total Manhattan distance between each tile's current position and its goal position. This is done by summing the absolute differences of the current and goal row and column indices of each tile.

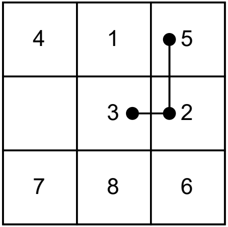

Figure: In this example the Manhattan distance of the tile 3 is 2..

Heuristics

Euclidean distance

This heuristic calculates the total Euclidean distance between each tile's current position and its goal position. This is done by summing the square root of the sum of the squared differences of the current and goal row and column indices of each tile.

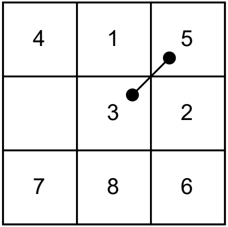

Figure: In this example the Euclidean distance of the tile 3 is 1.

Heuristics

Misplaced tiles

This heuristic simply counts the number of tiles that are not in their goal position on the puzzle.

4	1	5
	3	2
7	8	6

Figure: In this example there are 6 tiles out of place.

Heuristics

Linear conflict

This heuristic counts the number of pairs of tiles that are in the same row or column as their goal position but are in the wrong order.

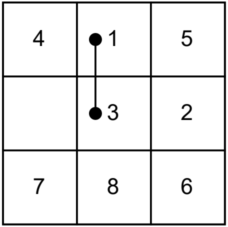

Figure: In this example is possible to observe a linear conflict between 1 and 3.

Heuristics

Linear conflict + Manhattan distance

This heuristic is a combination of the Linear conflict and Manhattan distance heuristics. It calculates the sum of both the Manhattan distance and the Linear conflict for the puzzle.

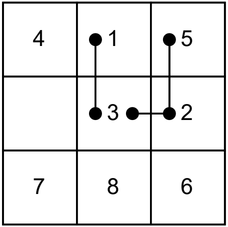

Figure: In this example is possible to observe that the sum of the two heuristics for tile 3 is equals to 3.

8-Puzzle

8	6	5
7	4	
2	1	3

Table: 8-Puzzle generated with 39 steps.

Heuristic	**CPR**	**CP**	**#NE**	**PR**	**MP**	**T**
MD	39	27	8360	27	27	3.9s
LC + MD	39	27	**4460**	27	27	1.5s

Table: Execution of algorithm A*.

Heuristic	**CPR**	**CP**	**#NE**	**PR**	**MP**	**T**
MD	39	27	960	27	20	0.22s
LC + MD	39	33	**346**	33	19	0.078s

Table: Execution of algorithm BA*.

15-Puzzle

1	5	2	7
10	14	11	6
15	12	9	3
13		8	4

Table: 15-Puzzle generated with 34 steps.

Heuristic	CPR	CP	#NE	PR	MP	T
MD	34	34	**5505**	34	34	2.8s
LC + MD	34	40	46191	40	40	337.9s

Table: Execution of algorithm A*.

Heuristic	CPR	CP	#NE	PR	MP	T
MD	34	36	**1467**	36	30	0.8s
LC + MD	34	40	2728	40	37	2.8s

Table: Execution of algorithm BA*.

15-Puzzle

1	11	6	2
10	13	15	5
3	12		4
9	7	14	8

Table: 15-Puzzle generated with 40 steps.

Heuristic	**CPR**	**CP**	**#NE**	**PR**	**MP**	**T**
MD	40	??	+46508	??	??	+3600s
LC + MD	40	??	+46191	??	??	+3600s

Table: Execution of algorithm A* not complete.

Heuristic	**CPR**	**CP**	**#NE**	**PR**	**MP**	**T**
MD	40	42	**640**	42	30	0.88s
LC + MD	40	48	1054	48	37	1.8s

Table: Execution of algorithm BA*.

24-Puzzle

1	3	4	14	5
6	7		8	9
11	18	2	23	13
16	12	19	24	17
21	22	20	15	10

Table: 24-Puzzle generated with 39 steps.

24-Puzzle

Heuristic	CPR	CP	#NE	PR	MP	T
MD	39	39	40950	39	39	232s
LC + MD	39	41	**11755**	41	42	23s

Table: Execution of algorithm A*.

Heuristic	CPR	CP	#NE	PR	MP	T
MD	39	39	**1223**	39	30	2.6s
LC + MD	39	45	6626	45	40	34s

Table: Execution of algorithm BA*.

24-Puzzle

	1	3	14	5
6	7	4	8	9
11	18	2	23	13
16	12	19	24	17
21	20	12	15	10

Table: 24-Puzzle generated with 46 steps.

24-Puzzle

Heuristic	**CPR**	**CP**	**#NE**	**PR**	**MP**	**T**
MD	46	46	83773	46	46	1117.6s
LC + MD	46	48	**17277**	48	49	50s

Table: Execution of algorithm A*.

Heuristic	**CPR**	**CP**	**#NE**	**PR**	**MP**	**T**
MD	46	48	**1896**	48	25	1.49s
LC + MD	46	48	16695	48	37	108s

Table: Execution of algorithm BA*.

1	12	2	14	4	6	7	8	9	10
11	22	3	13	5	15	17	18	19	20
21	32	23	24	25	16	27	28	29	30
31	42	34	35	36	26	37	38	39	40
41	33		54	44	46	47	48	49	50
51	52	43	53	45	56	57	58	59	60
61	62	63	64	55	76	66	68	69	70
71	72	73	74	65	75	67	88	78	80
81	82	83	84	85	86	77	87	79	90
91	92	93	94	95	96	97	98	89	99

Table: 100-Puzzle generated with 34 steps.

100-Puzzle

Heuristic	CPR	CP	#NE	PR	MP	T
MD	34	36	**87304**	36	39	1570.78s
LC + MD	34	??	+159808	??	??	+7200s

Table: Execution of algorithm A*, not complete for the LC + MD heuristic.

Heuristic	CPR	CP	#NE	PR	MP	T
MD	34	38	**176**	38	36	1.27s
LC + MD	34	??	+110378	??	??	+7200s

Table: Execution of algorithm BA*, not complete for the LC + MD heuristic..

Conclusion

From the results obtained from the experiments it has been deduced that the BA* algorithm allows to solve in less time and with a lower number of expanded nodes most of the puzzles that need about 40 steps to be solved. Furthermore, it has been observed that as the size and difficulty of the puzzle increases, a simple heuristic, such as the Manhattan distance, makes informed search algorithms more efficient.

N-Puzzle

Practical analysis of the problem

Luigi Seminara

Dipartimento di
Matematica e Informatica
Università degli studi di Catania, Italia

Academic Year 2022/2023