Sokoban Search Algorithm[*]

An Khanh Vo
19520007@gm.uit.edu.vn

March 2021

Abstract

Sokoban is a simple transport puzzle game. In the game, the agent is reponsible for pushing boxes to target points. The rule are quite simple, but it is very difficult for humans when the computations become larger. In this report, we try to find the solution for this game using various uniformed search algorithm such as Depth First Search (DFS), Breadth First Search (BFS) and Uniform Cost Search (UCS). We have implemented these three algorithms and compare their performance.

1 States and Modeling

To be able to solve Sokoban, we need to reduce attention to the location of the boxes and the target. We consider each state as a vertex in the graph. The start state is the state at the beginning of the level whereas when all boxes transfered to suitable targets. The actions can be moved in four directions(up, down, left, right) resulting in four successor state at a cost of -1. We can consider each action as an edge in the graph with a weight of -1.

The input for modeling consist of following characters in 2D map: "#" is a wall, " " is a free space, "B" is a box, "." is a goal location, "&" is our agent.

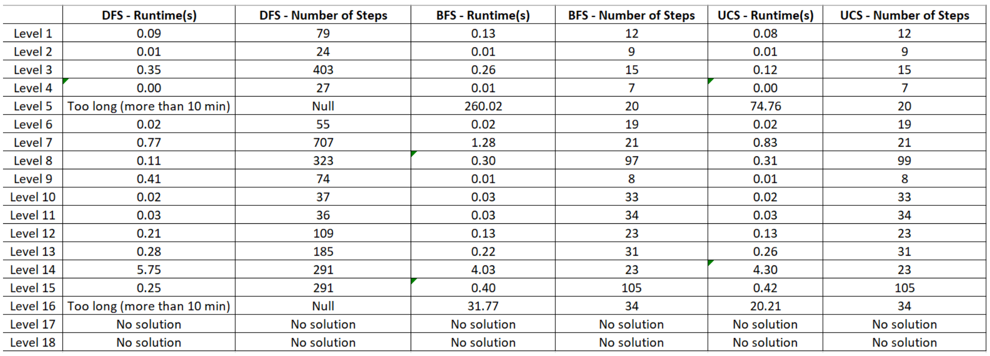
In Sokoban game, we have some constraints that if not satisfied will not solve problem:

1. The box must not be in the dead state
2. The agent is not strong enough to be able to push two or three at the same time.

Some testcaes in this problem do not satisfy the above conditions. Therefore, we have no way to solve these testcases.

2 Result Analysis

In order to illustrate the differences between various search algorithms, we are going to show the result of the same map and compare the results with respect to the number of steps and runtime.



	DFS - Runtime(s)	DFS - Number of Steps	BFS - Runtime(s)	BFS - Number of Steps	UCS - Runtime(s)	UCS - Number of Steps
Level 1	0.09	79	0.13	12	0.08	12
Level 2	0.01	24	0.01	9	0.01	9
Level 3	0.35	403	0.26	15	0.12	15
Level 4	0.00	27	0.01	7	0.00	7
Level 5	Too long (more than 10 min)	Null	260.02	20	74.76	20
Level 6	0.02	55	0.02	19	0.02	19
Level 7	0.77	707	1.28	21	0.83	21
Level 8	0.11	323	0.30	97	0.31	99
Level 9	0.41	74	0.01	8	0.01	8
Level 10	0.02	37	0.03	33	0.02	33
Level 11	0.03	36	0.03	34	0.03	34
Level 12	0.21	109	0.13	23	0.13	23
Level 13	0.28	185	0.22	31	0.26	31
Level 14	5.75	291	4.03	23	4.30	23
Level 15	0.25	291	0.40	105	0.42	105
Level 16	Too long (more than 10 min)	Null	31.77	34	20.21	34
Level 17	No solution	No solution	No solution	No solution	No solution	No solution
Level 18	No solution	No solution	No solution	No solution	No solution	No solution

Figure 1: Performance comparison between DFS, BFS and UCS

[*] This is my report for CS106 - Artificial Intelligence (Spring 2021) at University of Information Technology - Vietnam National University HCMC

In DFS, we obtain with largest number of steps in algorithm. It will always continue to expend nodes in the current path until can not expand anymore, and then change the path. For this reason, it took a long time for DFS to find a solution. It can go very deep into a branch of tree and take a long time to get out.

We can see that BFS always returns the result with smallest number of steps to the goal. BFS always returns the first node of queue and will return the results immediately when a solution have been found. In other words, BFS returns the optimal path. It also takes very little time to find the solutions.

In Sokoban, UCS is quite efficient becase environment has a lot of empty state, each action costs minus one. In this algorithm, it uses cost to make the key to get the node out of the priority queue. Since each action has a cost of minus one, it is highly adaptable to this environment and it is the best algorithm. The UCS solution are almost optimal in this solution.

Most of testcases in this problem are very easy to solve. Expect for test 5 and test 16 which took a lot of time, UCS dominated in these tests and DFS became too slow. Test 17, 18 are in "sokobanLevels" folder but are ignored in the template code. I ran and checked, there is no solution for these tests.

3 Summary

BFS gives the best solution at runtime quite quickly but still quite slow in test 5 and test 16. DFS gives a very bad solution and is also very slow. UCS is the best algorithm in Sokoban both in terms of time and number of steps.