ISSN: 2230-9926

Available online at http://www.journalijdr.com

IJDR

International Journal of Development Research
Vol. 10, Issue, 11, pp. 42345-42347, November, 2020
https://doi.org/10.37118/ijdr.20323.11.2020

RESEARCH ARTICLE OPEN ACCESS

SOLVING 2048-A DETAILED COMPARISON OF STRATEGIES

*Chakor Pandkar, Atharva Pandkar and Niket Doke

Maharashtra Institute of Technology, Pune, India

ARTICLE INFO

Article History:
Received 04th August, 2020
Received in revised form
19th September, 2020
Accepted 26th October, 2020
Published online 30th November, 2020

Key Words:
2048, Mathematical Puzzles, Solving 2048, 2048 Game, 2048 strategies, 2048 Patterns.

Corresponding author: Chakor Pandkar

ABSTRACT

2048 is a mathematical puzzle game which although looks relatively simple with primitive and easy to understand rules. It has been proved that 2048 is Pspace Hard and also computationally np-hard in determining the end tile from a given starting position. Although there have been several attempts in solving 2048 using machine learning algorithms , none of them are easily human understand- able and some of them are also compromised in their consistency. We have thus created our own algorithm that is easier to understand and comprehend by human players. We have also performed a detailed comparison in the pattern strategies explored by most human players.We have concluded on the impor- tance of flexibility in solving the mathematical game. These strategies could be combined in further research to solve the np-hard game.

Citation: ***Chakor Pandkar, Atharva Pandkar and Niket Doke. 2020.*** "Solving 2048-A detailed Comparison of Strategies*", International Journal of Development Research*, 10, (11), 42345-42347.

INTRODUCTION

2048 is a mathematical puzzle game created by Gabriele Circulli [Gabrielecirulli]. The game has a simple agenda , to get to the tile 2048. It is a single person game that according to the author has gone viral over the internet and rightly so. The game progressively gets tougher as the player gets closer to the final tile. The simple sets of rules often elude the player who soon realizes the complexity and difficulty of the game. It has been proved that it is NP hard to predict if a starting position in a m*n board would reach a specific tile value [Langerman, 2018]. The setup of the most popular and original version is as follows. 2048 is played on a 4x4 grid by a single player. The game begins with all the 16 cells, empty and progresses by inserting a 2 or a 4 randomly in one of the empty cells after a valid move. A valid move consists of UP, DOWN, LEFT and RIGHT where the direction define the direction in which all the non-empty tiles are pushed in. If while pushing in any direction a non-empty tile hits another non empty tile of the same value they merge together with the tile value being the sums of the 2 merging tiles. It is hence not hard to see that all the numbers are always multiples of 2. The goal of the player is to reach the tile 2048. The game is declared over when there are no valid moves left i.e there are no empty tiles left and moves in any direction do not create any new merging tiles. The game score is a sum of all the merges occurred until then but we exclude this factor while judging out program. We judge our program solely on the highest tile acquired by the iteration.

Related Work: There have been a couple of papers that have evaluated the time and space complexity of the game. In [http://www.sciencedirect.com/science/article/pii/ S0304397 51830 1798] the authors have proved that the game 2048 over a board of 4*4 is PSPACE Hard. The authors of [Langerman, 2018] also conclusively proved that it is NP hard to predict that whether a starting position in a m*n board of 2048 would be able to reach a specific tile value.

Research has also been done to solve 2048 using artificial intelligence and heuris- tic functions. In [Mehta, 2014] the authors have solved 2048 using an expictimax search on the state space. The use of expictimax in this paper made the approach luck based and dependent on the new tiles. The authors tried a Q learning method to solve 2048 but it being computationally worse, performed poorly as compared to the expicitmax search using a heuristic function did. This method although might be easily computable but is very hard to be human understandable. The authors of [Chowdhury] solved the game

2048 using various machine learning agents like Reflex Agent, Depth Limited Expectimax and Markov Decision process. Although the use of these 3 different approaches was very innovative the results acquired by the authors are primitive at best with the highest tile being 256 in 40 Reflex and 1024(very rarely) in Depth Limited Expectimax. We have thus tried to tackle these issues by creating a human understandable algorithm that can be adopted by human players and one which gives a consistent performance.

Proposed Approach: Weproposetosolve2048usingabetter heuristic function with only a depth of 4 moves in the future. Searching the state space the function finds the best possible outcome and makes a move towards that outcome. After every move a new state space of depth 4 would be created. We specifically have chosen a small depth of 4 to provide evidence that any human player could adapt this heuristic in their playing style to achieve the tile 2048. The heuristic function is based on 2 factors only 1. Trying to maintain a pattern 2. The number of blank spaces available We have experimented with a few patterns including the most common s pattern equaling the results of various patterns to provide a comprehensive comparison in the strategies. The more the number of bank spaces, further the player is from a game over. These two factors are combined to create the heuristic function.

Psuedo Code

.SUDO for heuristics

- .Set num to pattern
- .Set score to 0
- .Set b to number of blanks
- .Set constant to 16

.for i in range 0 to 15

.if game[row][column] is empty
.add score with constant*(constant**(MaxDepth-CurrentLevel))
.END if
.else

add score with num[i] * constant * (value at [row][column] ** chk(game))

. END else
. END for
.SUDO for chk
.Set cnt to 0

.for row in range 0 to 4

. for column in range 0 to 4
. if game[row][column] is empty
. increment cnt
. END if

END for

.END for

Patterns: We have explored the following 4 patterns and ran and logged the maximum tiles for a 100 test cases each.

The S pattern

16	15	14	13
9	10	11	12
8	7	6	5
1	2	3	4

Figure 1. The S Pattern

This is one of the most popular patterns used by many human players. The values in the respective boxes were multiplied with the numbers displayed in

the pattern.

The Half S Pattern

16	15	14	13
9	10	11	12
12	11	10	9
13	14	15	16

Figure 2: The Half S Pattern

This pattern was explored in order to provide flexibility to the algorithm by allowing it to make an inverted S if an when necessary.

The N-1 Pattern

16	15	14	13
15	14	13	12
14	13	12	11
13	12	11	10

Figure 3. The N-1 Pattern

This pattern was explored to deviate slightly from the S pattern strategy to create local maximas and combining them to solve the game.

The Random Pattern

We used this as a control for the other patterns by randomly shuffling the num- bers around.

14	2	7	11
10	1	16	13
4	8	9	12
3	15	6	5

Figure 4. The Random Pattern

Functions

We used 5 different functions on each pattern to explore the best possible values for those respective patterns. A 100 test cases were run for each of these slightly modified patterns. The five different functions used on the pattern were:

- Power 2 function- Each element in the pattern was raised to the power of 2
- Power 3 function- Each element in the pattern was raised to the power of 3
- Power 4 function- Each element in the pattern was raised to the power of 4
- 16-function-Each element in the pattern was raised to the power of (16-i) where i is the cell number
- Depth 3- The pattern was run with a minimized depth of 3 instead of 4

RESULTS

We ran the various patterns and every modification for a 100 test cases each and tabulated the highest tile reached in every one of them. We choose to tabulate the highest tile instead of the score as that would be a clearer indication of how close the player was in finishing the game. We have added the counts of the games reaching the same highest tile to tabulate the number of games ending in that particular highest tile. We have tabulated the results of all the patters as follows:

Table 1: S Pattern

Highest Tile	Power 4	Power 3	Power 2	16 position	Depth 3
2048	0	0	2	0	1
1024	39	53	50	4	47
512	52	39	42	53	43
256	9	8	6	42	7
128	0	0	0	1	2

Table 2: Half S Pattern

Highest Tile	Power 4	Power 3	Power 2	16 position	Depth 3
2048	3	1	0	3	1
1024	39	40	26	55	56
512	56	50	63	39	41
256	2	9	11	3	1
128	0	0	0	0	1

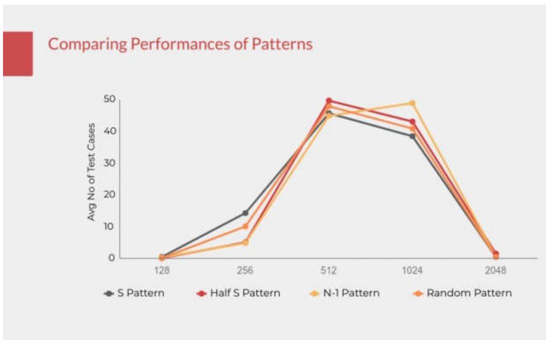

Figure 5. The Comparison between strategies

Analysis: To analyze our results better we have illustrated this graph that lists the totals of highest tiles reaches across all 500 test cases.

As illustrated the different strategies work very similarly with even the random permutation producing comparable results.

Table 3. N-1 Pattern

Highest Tile	Power 4	Power 3	Power 2	16 position	Depth 3
2048	1	0	1	1	0
1024	48	43	50	53	51
512	45	48	43	44	45
256	5	9	5	2	4
128	1	0	1	0	0

Table 4. Random Pattern

Highest Tile	Power 4	Power 3	Power 2	16 position	Depth 3
2048	0	0	3	0	0
1024	46	30	38	42	49
512	40	56	52	47	45
256	14	13	7	11	6
128	0	1	0	0	0

The results of the S pattern corrob- orate its popularity among human players with a highly consistent performance of achieving the tile 1024. The Half S pattern slightly out performs the S pat- tern thus enforcing the importance of flexibility in this game. Since the game is unpredictable because of it being np hard in nature the importance of flexibility is highlighted by the Half S pattern. This flexibility would entail reversing the direction of S if need be. The N-1 pattern that focuses on local maximas sur- prisingly performs better than its counterparts for consistently achieving 1024. This suggests that with a few tweaks this strategy which is the easiest to follow and comprehend would be enough to achieve the tile 2048. The scalability of this strategy is questionable but as far as winning the game is concerned, this strategy would be sufficient. The random pattern was utilized as a control for the other strategies. We have thus proved that a simple strategy like the N-1 would work better than the S pattern at a minimal depth for human players to follow along. A future scope would be to combine the different patterns in dif- ferent scenarios to provide even better algorithms that would outperform their parents.

REFERENCES

Gabrielecirulli, 2018 original game. arXiv:http://gabrielecirulli. github.io/2048/.

Langerman, S. Y. Uno, Threes!, fives, 1024!, and 2048 are hard, The- oretical Computer Science 748 (2018) 17 – 27, fUN with Algorithms. doi:https://doi.org/ 10.1016/ j.tcs.2018.03.018.

URL http://www.sciencedirect.com/science/article/pii/ S0304 397518301798

Mehta, R. 2018 is (pspace) hard, but sometimes easy (2014). arXiv:1408. 6315.

Chowdhury, G. V. Dhamodaran, 2018 using expectimax.arXiv: https://www.cs.uml.edu/ecg/uploads/AIfall14/vignesh_gayas_ 2048_project.pdf.

LUI, M. H. A. PROST, C. W. LEE, Solving the game ‘2048’ using vari- ous machine-learning agents. arXiv:https:// home.cse.ust.hk/~yqsong/ teaching/comp 3211/ projects/ 2017Fall/G13.pdf.

****