SEARCH PATHOLOGY OF 8-PUZZLE

Rok Piltaver, Mitja Luštrek, Matjaž Gams
Department of Intelligent Systems
Jozef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
e-mail: {rok.piltaver | mitja.lustrek | matjaz.gams}@ijs.si

ABSTRACT

8-puzzle is typically solved by heuristic search. Real-time heuristic search usually gives better results when searching deeper. But sometimes deeper search leads to worse results than shallower which is a phenomenon called search pathology. In this paper we present the results of our investigation of the causes for pathology in 8-puzzle and some of its variations.

1 INTRODUCTION

8-puzzle (or 8-tiles sliding game) is a simple game in which one has to rearrange 8 tiles on 3 by 3 grid by sliding one tile at a time into an empty slot. The objective of the game is to rearrange the tiles into given order in as few moves as possible. Figure 1 shows how to solve start position (a) in 4 moves to obtain goal position (e).

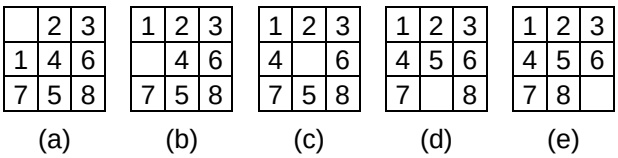

Figure 1: *An example of solving 8-puzzle.*

A program that is trying to solve a given start position in the (8-puzzle) game generates a tree representing all possible sequences of moves of a certain length. Nodes of the tree represent positions in the game. Two nodes are connected with directed edge pointing from position p_1 to position p_2 if a move that transforms p_1 into p_2 exists. The program evaluates the minimal number of moves needed to solve positions in the leaves of that tree by using a certain heuristic function [4]. The move that leads to the sub tree that has the minimal value of the heuristic function in a leaf is chosen as the best move. Usually the deeper the tree is (the longer the sequence of moves) the more likely it is to choose the optimal move. But sometimes deeper trees mislead the algorithm into choosing wrong moves, whereas shallower trees would suggest correct moves. That is called search pathology [1, 3 and 7] and we want to avoid it if we can. In this paper we investigate when and why pathology happens by varying number of factors that may influence pathology and evaluating the correlation between them and measured pathology of a certain variation of 8-puzzle solved using a certain heuristic function.

The pathology of minimax search was independently discovered by D. S. Nau in 1979 [10] and D. F. Beal in 1980 [11]. Pathology of single-agent search was discovered much later in 2003 by V. Bulitko [7]. The causes for pathology of minimax search and their influences on pathology are described in [1, 3 and 6] so we studied the influence of the same factors and some additional ones in the domain of 8-puzzle (and its variations) which is known to be pathological [9, 2]. We investigate the pathology of single-agent search with the assumption that it behaves similarly as pathology of minimax search.

The rest of the paper is structured as follows. In Section 2 we explain how we modeled heuristic function, how we evaluated percent of correct decisions and pathology and influence of granularity of heuristic function on pathology. In Section 3 we present variations of 8-puzzle and some statistic about them. We also describe influence of similarity of sibling nodes in the search tree and branching factor of the search tree on pathology. In Section 4 we present results obtained using some other heuristic functions. Section 6 gives the conclusions.

2 THE USUAL 8-PUZZLE

It is known that 8-puzzle is pathological [2] so for the first part of our paper we use similar heuristic function as Sadikov and Bratko did in their paper. To obtain it we first calculated the optimal (minimal) numbers of moves needed to solve each solvable start position $h^*(n)$ with the use of retrograde analysis, a technique known from computer chess, where it is used to generate endgame databases [8]. We started from goal position and expanded the search tree in reverse order until depth of 31 where we found all solvable positions of 8-puzzle [5]. A solvable position is a position that can be solved using the allowed moves of the empty slot. There are 9!/2 solvable positions in the usual 8-puzzle [5].

Then we simulated the heuristic values $h(n)$ by corrupting the optimal values in two steps. In the first step we took position's true value $h^*(n)$ and added to it a certain amount of Gaussian noise. The added noise caused that some of the heuristic values were grater and the others were smaller than the true values. Sadikov and Bratko used two different heuristic functions one that was pessimistic and the other optimistic which means that in the first case the positive noise was added to the true values and in the second it was

subtracted. Results of using pessimistic and optimistic heuristic functions are presented in Section 4.

For standard deviation of added Gaussian noise we choose $\sigma = 2.5$ to equal the standard deviation of Manhattan distance heuristic function [2, 4], which is well known optimistic heuristic for the 8-puzzle domain. We did not corrupt the optimal evaluations for the first 7 levels of difficulty ($h^*(n) \leq 7$), because few positions belong to these levels and it is therefore practically impossible to corrupt them so that they would maintain more or less constant dispersion [2].

In the second step we limited the number of possible heuristic values as follows. We limited maximal and minimal heuristic value so that $\forall n$: $h(n) \in [0, M]$ where $M = \max_n\{h^*(n)\} + \lfloor\sigma\rfloor + 1$. We choose M so that it was close to maximal heuristic value and that only few heuristic values of certain positions were greater than M. If $h(n) > M$ we set it to M and if $h(n) < 0$ we set it to 0. Then we multiplied all heuristic values by a certain factor to scale the interval of possible heuristic values to $[0, g]$ and rounded them to the closest integer value. We call g granularity of heuristic function because it denotes the number of possible values of the heuristic function.

In the next step of our experiment we calculated average percentage of wrong decisions in the case of 1 and 5 levels of lookahead among all solvable positions. The percentage of wrong decisions using d levels of lookahead for given position m was calculated using the following formula:

$$\text{wrong}_d(m) = \frac{\#(\text{mintree}_d\, h(n) \wedge \neg \min h^*(n))}{\#\,\text{mintree}_d\, h(n)} \qquad (1)$$

Denominator $\#\,\text{mintree}_d\, h(n)$ means the number of nodes n reachable from m in one move that have the smallest value of $h(p)$ where p is a leaf of the subtree rooted in n with depth $d - 1$. In other words, it is the number of sibling nodes that have the smallest backed-up value of heuristic function. In case of $d = 1$ the only node of subtree is its root so the backed-up value is equal to the value of heuristic function in the root. Numerator $\#(\text{mintree}_d\, h(n) \wedge \neg \min h^*(n))$ is the number of positions n reachable from m by one move that have smallest value of backed-up heuristic function among sibling nodes and do not have the smallest value of $h^*(n)$ among all their siblings. In other words, the numerator represents the number of moves that are the best according to the heuristic function but are not optimal. The formula 1 gives the probability that the search algorithm will choose the wrong move in a position m if we let it look d levels deep and if it randomly decides which move to make when more than one sibling node has the smallest value of backed-up heuristic function. An example of calculating $\text{wrong}_1(m)$ and $\text{wrong}_5(m)$ is shown in Figure 2 and Figure 3.

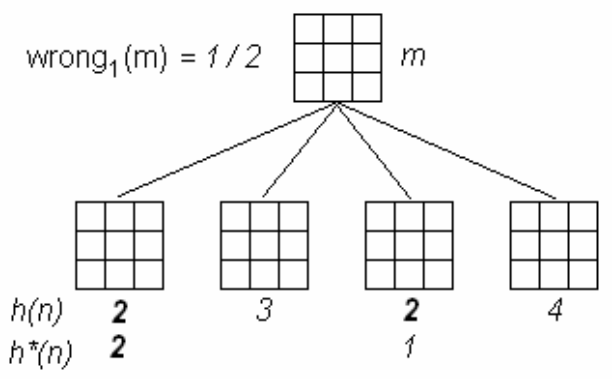

Figure 2: *Example of calculating $wrong_1(m)$*

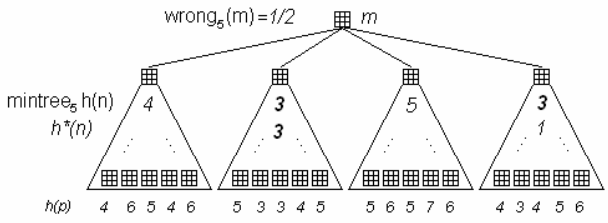

Figure 3: *Example of calculating $wrong_5(m)$.*

In last step of our expermint we calculated pathology using the followig formula:

$$\text{pat}\ j/1 = \frac{arv(j)}{avr(1)},\ avr(k) = \frac{\sum_{i \in SolvPos} wrong_k(i)}{|SolvPos|} \qquad (2)$$

All solvable positions in 8-puzzle compose a set denoted by *SolvPos*.

Graph of *pat5/1* with respect to g (the granularity of the heuristic function) is shown in Figure 4. We see that *pat5/1* decreases with increasing g and that solving the puzzle is pathological for $g < 10$ and is not pathological for $g \geq 10$. Result is qualitatively the same as in min-max search model described in [6].

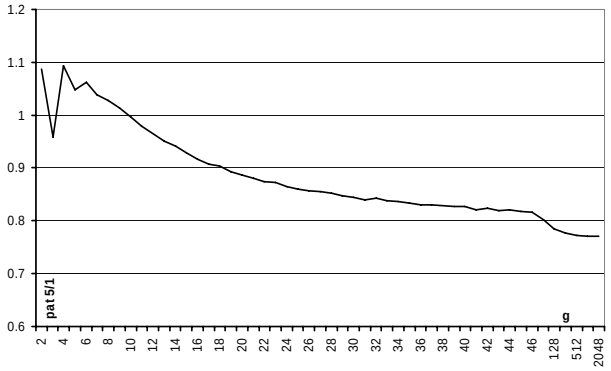

Figure 4: *Pat5/1 of usaual 8-puzzle.*

3 VARIATIOS OF 8-PUZZLE

After experiments on the usual 8-puzzle we tried to vary the branching factor and similarity of sibling nodes in the search tree. In order to do so we had to introduce additional moves. Besides the 4 usual moves we considered 4 additional ones. All possible moves are shown in Figure 5.

By selecting all possible subsets of the 8 moves we came up with

$$\sum_{i=1}^{8}\binom{8}{i} = 255 \qquad (3)$$

different games. There were 129 games that had only a few solvable start positions (less than 202), 31 games with 9!/2 = 181440 solvable start positions and 95 games with 9! = 362880 solvable positions. We decided to study only the games with many ($\geq$ 9!/2) solvable positions because the results from games with less than 202 solvable start positions are statistically much less significant and those games are not very playable either.

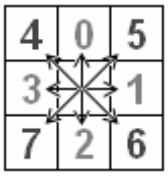

Figure 5: *Possible moves in variations of 8-puzzle.*

We found out that even some games with only 3 moves produce 9! solvable start positions. In the worst case 46 moves are needed to solve the most difficult start position (in a game with 3 moves), whereas in the game with all (8) possible moves there are only 20 moves needed to solve the most difficult start position. We measured the average branching factor of the interesting games and came up with 13 groups of games with branching factors: 1.56, 1.78, 2, 2.22, 2.44, 2.67, 2.89, 3.11, 3.33, 3.56, 3.78, 4 and 4.44.

We run the same tests as for the usual 8-puzzle on all the 126 interesting games (including the usual 8-puzzle) and draw the graph of *pat5/1* with respect to the branching factor. We expected to see the influence of branching factor on pathology [6], but that did not happen. A likely explanation is that the games differ in many aspects besides the branching factor: in the **number of possible moves**, the **number of solvable start positions** (two groups with 9! and 9!/2 solvable start positions), the **length of optimal solution for worst-case start position** denoted *maxOD* (21 groups of games with *maxOD* ranging from 20 to 46) and the **percentage of the positions in which all possible moves are optimal** (percentage is ranging from 12% to 70.7% but 61% of the games have less then 25% of positions in which all moves are optimal and only 6% of games have more than 45% of such positions). We were unable to determine the exact influence of these factors on pathology. But we know that games with 3, 7 and 8 possible moves are highly pathological and that pathology of games with the same branching factor decreases with increasing *maxOD*.

According to [6] the similarity of sibling nodes is also an important factor that causes or reduces the pathology so we calculated the similarity of sibling nodes for all the games using correlation (which indicates the strength and direction of a linear relationship between two random variables). Correlation is calculated for pairs (X, Y), in our case the pairs were the true value of a position and its descendants (node, $decs_1$), (node, $decs_2$), ... , (node, $decs_b$). Similarity of interesting games varied from 0.877 to 0.966. That means that there is not much difference in similarity, so we did not expect a strong influence of similarity on *pat5/1*. Despite this we noticed that the *pat5/1* is slowly decreasing with increasing similarity (Figure 6). Again result is qualitatively the same as in min-max search model described in [6].

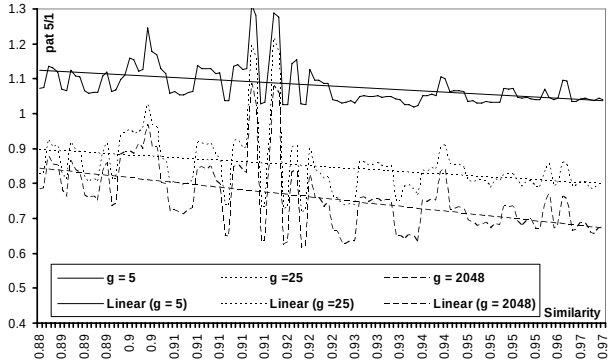

Figure 6: *The influence of similarity on pat5/1 for $g \in \{5,25,2048\}$ in variations of 8-puzzle.*

4 OTHER HEURISTIC FUNCTIONS

We also studied the influence of σ the standard deviation of the heuristic error. We run the tests described above on the usual 8-puzzle (only 4 basic moves of an empty slot allowed) for a number of different values of σ. The results of some of the test are shown in Figure 7. We see that higher values of σ (larger heuristic error) result in higher *pat5/1* for low granularity and lower *pat5/1* for high granularity.

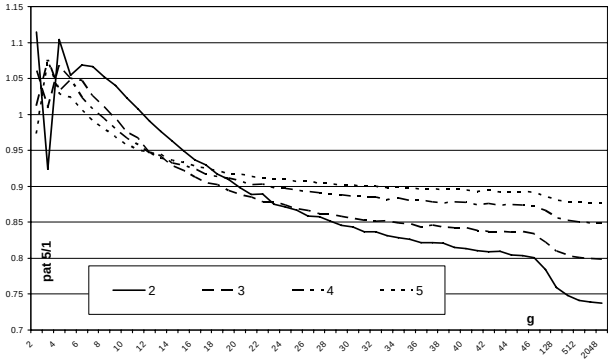

Figure 7: *pat5/1 of usual 8-puzzle for $\sigma \in \{2, 3, 4, 5\}$.*

We repeated the same experiment as in Section 1 for other heuristic functions. We used optimistic and pessimistic heuristic functions obtained by corrupting the true values

with either Gaussian noise (as in [2]) or by uniformly distributed noise (as in [1]). We noticed that pessimistic heuristic functions cause less pathology than optimistic ones. Sadikov and Bratko [2] showed that only for heuristic functions that were obtained by adding Gaussian noise to true values where as we found out that results are the same if we use uniformly distributed noise. The pathology in the case of heuristic function that is neither pessimistic nor optimistic (the one used in Sections 1-3) is greater than the pathology in the case of pessimistic and lower than in the case of optimistic heuristic function. These results can be seen in Figure 8. We also used some other heuristic functions but the main result was always the same: the higher the granularity the lower the pathology.

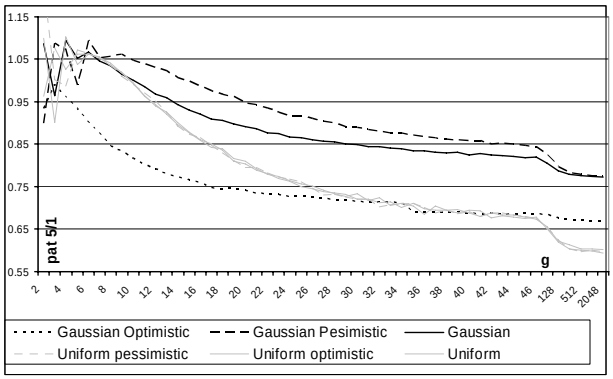

Figure 8: *pat 5/1 for different heuristic functions.*

4. CONCLUSION

We showed that higher granularity of heuristic function causes lower pathology in all interesting variations of 8-puzzle for a number of different heuristic functions (which can be seen in Figures 4, 6, 7 and 8). We also showed that higher similarity of sibling nodes slightly decreases pathology (Figure 6). We were unable to determine the effect of branching factor on pathology due to differences of games with different branching factors. Finally we showed that higher noise produces lower *pat5/1* if there are only a few possible values of heuristic function (granularity) and higher pathology if there are many possible values of heuristic function (Figure 8). The influence of noise is the same for Gaussian and uniformly distributed noise.

The results of our research regarding granularity and similarity are consistent with the results presented in [1 and 6]. According to those sources the branching factor also influences the pathology but as mentioned above we were unable to determine its effect in the domain of 8-puzzle and its variations. The advantage of pessimistic heuristic functions over optimistic ones is consistent with the results presented in [2]. The influence of amount of noise added to true values to obtain values of heuristic function is not studied in related work.

References

[1] M. Luštrek. *Patologija v hevrističnih preiskovalnih algoritmih*. Ph. D. thesis, University of Ljubljana, Faculty of Computer and Information Science, 2007.

[2] A. Sadikov, I. Bratko. *Pessimistic Heuristics Beat Optimistic Ones in Real-Time Search.* ECAI, 2006.

[3] M. Luštrek. *Pathology in Single-Agent Search*. Information Society conference, 2005.

[4] D. R. Kunkle. *Solving the 8 Puzzle in a Minimum Number of Moves: An Application of the A* Algorithm*. http://www.ccs.neu.edu/home/kunkle/docs/EightPuzzle.pdf

[5] A. Reinefeld. *Complete Solution of the Eight-Puzzle and the Benefit of Node Ordering in IDA*.* International Joint Conference on Artificial Intelligence, pp. 248-253, 1993.

[6] B. Kaluža, M. Luštrek, M. Gams, A. Tavčar. *Pathology in Minimax Searching.* International Electrotechnical and Computer Science Conference, 2007.

[7] V. Bulitko, Lihong Li, R. Greiner, I. Levner. *Lookahead pathologies for single agent search.* Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, 2003.

[8] K. Thompson. *Retrograde analysis of certain endgames*. ICCA Journal, 9(3), 131–139, 1986.

[9] V. Bulitko. *Lookahead pathologies and meta-level control in real-time heuristic search*. 15th Euromicro Conference on Real-Time Systems 13-16, 2003.

[10] D. S. Nau. *Quality of decision versus depth of search on game trees*. Ph. D. thesis, Duke University, 1979.

[11] D. F. Beal. *An analysis of minimax*. V Advances in Computer Chess 2. 103-109. Edinburgh University Press, 1980.