PATHOLOGY IN SINGLE-AGENT SEARCH

Mitja Luštrek
Department of Intelligent Systems
Jožef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
Phone: +386 1 4773380; fax: +386 1 4773131
E-mail: mitja.lustrek@ijs.si

ABSTRACT

In incomplete single-agent search, it is generally accepted that deeper searches produces better results. It has recently been discovered, though, that this is not always the case – such behavior has been termed pathological. This paper identifies two properties of search trees that cause pathological behavior and explains how they produce the pathology. A number of different heuristic functions were also investigated, focusing on admissibility and consistency. Consistency was most effective at preventing the pathology, while admissibility helped only in some cases.

1 INTRODUCTION

Search pathology occurs when the quality of a shallower heuristic search exceeds the quality of a deeper one. It is called pathology [9] because it is the opposite of what 'should' happen, namely that the more effort one puts into searching, the better results he gets. This phenomenon has been known in minimax search since 1979 [1, 9], but it has only been discovered in single-agent (or minimin) search in 2003 [5]. It does not occur in complete search methods such as A* [6], because these, given appropriate heuristics, compute the optimal path to the goal, leaving no room for the pathology. Where it has been observed is in incomplete search methods such as real-time A* [7]. Incomplete search methods perform minimin searches to a chosen depth, heuristically evaluate the search tree nodes at that depth and back their values up to a level below the root. These backed-up values are then used to choose the action leading from the root to its descendant with the optimal (typically minimal) value (typically representing the cost of reaching a goal node). We speak of the pathology when the choice of the action is more likely to be erroneous after a deeper search than after a shallower one.

There are two possible reasons for pathological behavior of single-agent search: the domain (or the nature of the problem) and the heuristic evaluation function directing the search. The domain is reflected in the distribution of true values in search trees. We identify two properties of search trees that lead to pathological behavior. The more interesting reason for the pathology, however, is the heuristic function, because unlike the domain, which is given, it can be controlled. The properties of heuristic functions that usually concern their designers are admissibility and consistency. The pathology turned out to be affected by both, by consistency more than by admissibility.

The paper is organized as follows. Section 2 briefly touches on the minimax pathology and then reviews the existing work on the pathology in single-agent search. Section 3 describes the properties of the domain that cause the pathology. Section 4 deals with heuristic functions. Section 5 concludes the paper and points out where further research is needed.

2 RELATED WORK

The minimax pathology was discovered independently by Nau [9] and Beal [1]. They set out to determine why minimaxing reduces the error of the heuristic evaluation function used to evaluate the leaves of the search tree, only to find out that on their seemingly reasonable models, it did exactly the opposite, i.e. amplify the error. Different explanations of this paradox were proposed, the most common [2, 3, 10] being that the pathological models ignored the similarity of positions close to each other, a characteristic of real games. More recent explanations [8, 11] featured reduced error at lower levels of search trees, which was previously dismissed.

It seems that the pathology in single-agent search was investigated only by Bulitko et al. They first demonstrated it [5] on a two-level search tree shown in Figure 1; the numbers in the figure are the nodes' true values and the letters the nodes' names. The tree was designed specifically to be pathological, but the heuristic function was 'fair'.

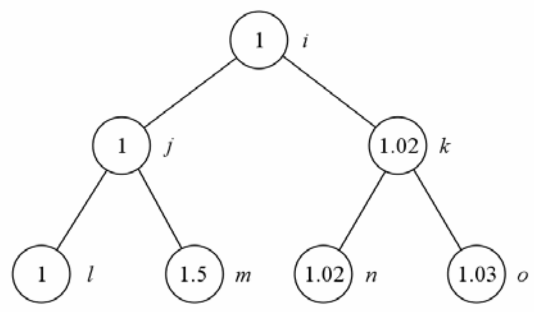

Figure 1. *Pathological search tree by Bulitko et al.*

The heuristic value or cost of a node is usually presented as $f = g + h$, where g is the known part and h the heuristic part. We will only discuss the cost as a whole: the heuristic cost of a node x will be denoted $c(x)$ and the true cost $c^*(x)$. In the example in Figure 1, $c(x)$ is distributed uniformly between $c(y)$ and $c^*(x)$, where y is the parent of x. This makes is both monotonically non-decreasing (which is equivalent to its heuristic part being consistent) and admissible. Let $Err(x, d)$ be the probability that among the descendants of a node x, one that does not have the lowest true value has the lowest heuristic value as returned by a search to depth d. In the example in Figure 1, $Err(i, 1) = 0.461 < 0.486 = Err(i, 2)$. This shows that the pathology is possible even with a heuristic function having both desirable properties, consistency and admissibility.

Bulitko later [4] discovered that the pathology can also occur outside synthetic search trees. He observed pathological behavior when solving the eight-puzzle using a heuristic function represented by an artificial neural network.

An issue Bulitko et al. did not address, though, is why the pathology occurs. This is what this paper focuses on.

3 DOMAIN AS A REASON FOR THE PATHOLOGY

The reasons for the pathology were mostly investigated on synthetic search trees with depth 2 and branching factor 2. Monte Carlo experiments were used to confirm that the conclusions also apply to larger branching factors and depths.

3.1 The First Reason

The first property of search trees that affects the pathology, *property 1*, is the difference in true value between the level-1 node with the lowest value and its descendants, compared to the difference in true value between other level-1 nodes and their descendants. It can be observed in the example from Figure 1, using Bulitko et al.'s heuristic function. To do that, we will treat heuristic values as random variables. We will write static $c(x)$ as X_0, backed-up $c(x)$ when searching d levels below x as X_d and $c^*(x)$ as X^*. $f_{X\,d}(y)$ is the probability density function of X_d. Figure 2 shows the probability density functions of J and K from the example in Figure 1: on the left side for search to $d = 1$ and on the right side for search to $d = 2$. The functions are integrated over all the possible values of I.

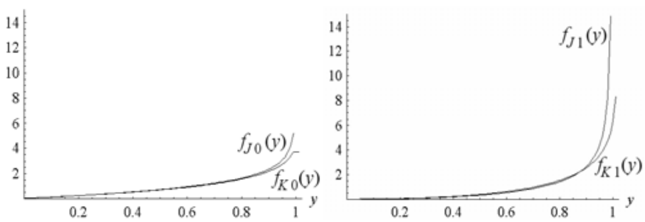

Figure 2. *Explanation of property 1.*

Since $J^* < K^*$, an error occurs when $J > K$, so we must determine why $P(J_1 > K_1) > P(J_0 > K_0)$; the left side of the inequality is shown in the right side of Figure 2 and the right side of the inequality in the left side of Figure 2. A node's backed-up heuristic value is higher than its static heuristic value because the static heuristic values of the node's descendants must be higher than the node's static value (due to consistency of the heuristic function) and the node's backed-up value is made up of the descendants' values. This can be seen in Figure 2, where the probability density functions on the right have on average a higher value than those on the left. If this increase were equal for both J and K, there would be no pathology. However, the increase is larger for J, which means that J_1 is closer to K_1 than J_0 to K_0, resulting in the pathological $P(J_1 > K_1) > P(J_0 > K_0)$. To see why J_1 is closer to K_1 than J_0 to K_0, we must consider that $J_1 = \min(L_0, M_0)$ and $K_1 = \min(N_0, O_0)$. Let us take two random variables X and Y. If $Z = \min(X, Y)$, Z tends to be lower than both X and Y, because for some values of $X = x$, there is a non-zero probability that Y (and therefore Z) is lower than x, and vice versa, for some values of $Y = y$, there is a non-zero probability that X (and therefore again Z) is lower than y. If Y is much higher than X, than Z will be very similar to X, because Y will have little effect on Z. The closer X and Y get, the more effect Y will have; its effect can only be to make Z lower, so the closer X and Y get, the lower Z will be. Since N_0 and O_0 are much closer to each other than L_0 and M_0 and since K_0 puts an upper limit on N_0 and O_0 in the same fashion as J_0 on L_0 and N_0, $K_1 = \min(N_0, O_0)$ is lower relatively to K_0 than $J_1 = \min(L_0, M_0)$ relatively to J_0. This explains why J_1 and K_1, the descendants of the lower level-1 node, are closer to each other than J_0 and K_0 and hence how property 1 produces the pathology in the example in Figure 1.

To further demonstrate the effect of property 1, let us strengthen it (cause it to produce a stronger pathology) by increasing M^* to 3. $Err(i, 1) = 0.461$ remains unchanged, while $Err(i, 2)$ increases from 0.486 to 0.507. If O^* is reduced to 1.02, $Err(i, 2)$ increases as well, this time to 0.492. Property 1 can also be weakened, for example by reducing M^* to 1 or by increasing O^* to 3. This eliminates the pathology by reducing $Err(i, 2)$ to 0.426 and 0.391 respectively.

3.2 The Second Reason

The second property of search trees that affects the pathology, *property 2*, is the difference in true value between the level-1 node with the lowest value and other level-1 nodes relative to the true value of the root. It can be explained on the search tree shown in Figure 3, which is somewhat related to the tree in Figure 1, but is not pathological. Bulitko et al.'s heuristic function is used again.

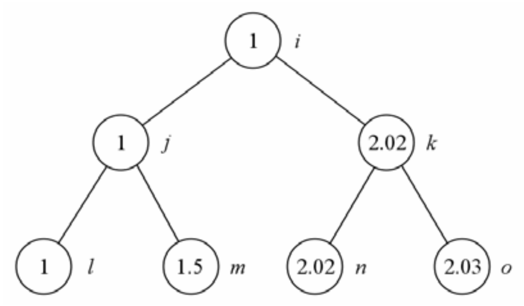

Figure 3. *Search tree to explain property 2.*

Figure 4 shows the probability density functions of *J* and *K* from the example in Figure 3: on the left side for search to $d = 1$ and on the right side for search to $d = 2$. The functions are integrated over all the possible values of *I*.

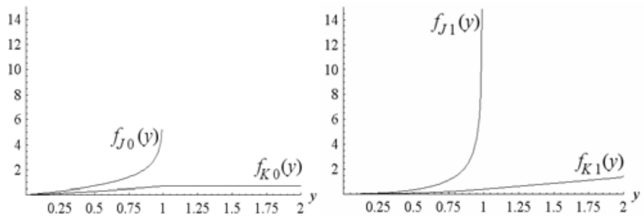

Figure 4. *Explanation of property 2.*

As can be seen in Figure 4, $f_{J\,1}$ (*y*) has larger values for *y* close to $J^* = 1$ than $f_{J\,0}$ (*y*); $f_{K\,1}$ (*y*) also has larger values for *y* close to $K^* = 2.02$ than $f_{K\,0}$ (*y*). This phenomenon was already explained in subsection 3.1 when discussing property 1. However, the example in Figure 3 has a much larger difference between $I^* = J^*$ and K^* than the example described in subsection 3.1. This gives $f_{K\,1}$ much more room to distance itself from $f_{J\,1}$ and thus eliminates the pathology. Property 2 does not only require level-1 nodes to be far away from each other to eliminate the pathology, it requires them to be far away from each other relatively to the value of one of them (or the root). Why this is necessary can easily be seen by proportionately increasing the values of all the nodes: this certainly increases the distance among level-1 nodes, but does not affect the error, so it cannot affect the pathology.

To further demonstrate the effect of property 2, let us strengthen it (cause it to produce a stronger pathology) on the tree from Figure 1. This can be done by reducing K^* and N^* to 1.01 and O^* to 1.02. On the unmodified tree, the increase in error when increasing the depth of search from 1 to 2 is 5.4 %. On the modified tree, it is 8.4 %, which is more pathological. Property 2 can also be weakened, for example by increasing K^* and N^* to 1.03 and O^* to 1.04. This reduces the increase in error to 2.9 %, which is less pathological.

4 HEURISTIC FUNCTION AS A REASON FOR THE PATHOLOGY

The two properties of the heuristic function that will be considered in this section are admissibility and consistency. If a heuristic function is admissible, the corresponding cost function will be called optimistic, because admissibility of *c* means $c(x) \le c^*(x)$, i.e. that the true value of *x* is always underestimated. If a heuristic function is consistent, the cost function is monotonically non-decreasing, which means $c(x) \ge c(y)$, where *y* is the parent of *x*. In complete search, the former implies the latter, but we are investigating incomplete search, where this is not the case, so we can deal with each property separately.

We will investigate uniformly and normally distributed cost functions. The cost function can be plain (the heuristic value of a node is simply distributed around its true value), optimistic, pessimistic (the opposite of optimistic, i.e. $c(x) \le c(y)$, where *y* is the parent of *x*), monotonic (monotonically non-decreasing), Bulitko (distributed as described in section 2) or a combination thereof.

The cost functions were compared in Monte Carlo experiments. 100,000 search trees with depth 2 and branching factor 2 were built and 100 sets of heuristic values were generated for each tree. Some experiments with larger depths and branching factors were also performed, yielding similar results. The percentage of pathological search trees is denoted *Pat*. However, different cost functions cause errors with different probabilities, so *Pat* might not be best suited to comparing them. For example, if one function causes very few errors, it will also be pathological on very few trees, even though it might be pathological on every tree where it causes an error. Therefore another measure is needed: relative pathology *RPat*. *RPat* is defined as *Pat* divided by the probability of a deeper search being beneficial compared to a shallower one. Both *Pat* and *RPat* for different cost functions are shown in Figure 5 and Figure 6.

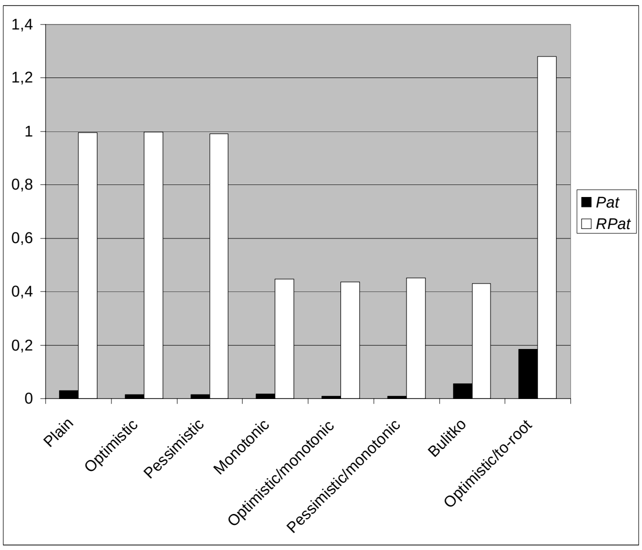

Figure 5. *Pathology of cost functions with uniform probability distributions.*

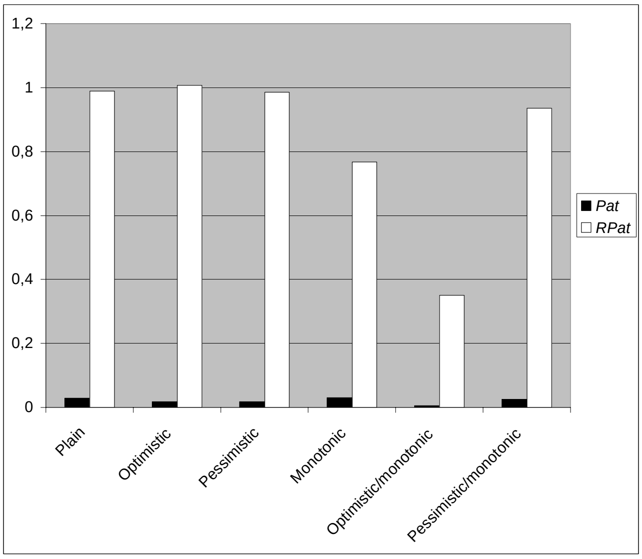

Figure 6. *Pathology of cost functions with normal-based probability distributions.*

Figure 5 and Figure 6 lead to the following conclusions:

- if the cost function is monotonic, the pathology is less likely;
- if the cost function is non-monotonic, the pathology is hardly affected by whether the function is optimistic or pessimistic;
- if the cost function is monotonic and distributed uniformly, the pathology is also hardly affected by whether the function is optimistic or pessimistic;
- if the cost function is monotonic and its distribution is based on normal, the pathology is less severe if the function is optimistic.

These conclusions do not appear to be very exciting: it is accepted that a cost function should be optimistic and monotonically non-decreasing. However, it should be noted that if a cost function is effective at leading an agent to desirable actions, that does not prevent it from being pathological. A good cost function can still be more effective when used in shallower searches than when used in deeper ones. So it is reassuring to *know* that the two properties already considered desirable also help with the pathology. Perhaps somewhat surprising is the discovery that it is more important that the function is monotonically non-decreasing than that it is optimistic, since usually more attention is paid to the latter. This is probably the legacy of complete search, where an optimistic cost function guarantees an optimal solution.

5 CONCLUSION

Unlike the minimax pathology, the pathology in single-agent is not very well understood. This paper sheds some light on why it occurs by identifying two properties of search trees that cause pathological behavior and explaining how they cause it. The effect of these two properties is undeniable on synthetic search trees, but it also needs to be verified on a practical example. This is one area where we will direct our future research.

The distribution of true values in the search tree is only one reason for the pathology – the other is the heuristic function. We showed that to avoid the pathology, the heuristic function should first be consistent and then admissible. However, since pathological behavior has been observed even with consistent and admissible heuristic functions, there are probably other characteristics of heuristic function that cause the pathology. They also need to be verified on a practical example. This is the other area where additional research is required.

References

[1] Beal, D. F. (1980). An Analysis of Minimax. In Advances in Computer Chess 2 (ed. M. R. B. Clarke), pp. 103-109

[2] Beal, D. F. (1982). Benefits of Minimax Search. In Advances in Computer Chess 3 (ed. M. R. B. Clarke), pp. 17-24

[3] Bratko, I. and Gams, M. (1982). Error Analysis of the Minimax Principle. In Advances in Computer Chess 3 (ed. M. R. B. Clarke), pp. 1-15

[4] Bulitko, V. (2003). Lookahead Pathologies and Meta-level Control in Real-time Heuristic Search. Proceedings of 15th Euromicro Conference on Real-Time Systems, Porto, Portugal, pp. 13-16

[5] Bulitko, V., Li, L., Greiner, R. and Levner, I. (2003). Lookahead Pathologies for Single Agent Search. Proceedings of International Joint Conference on Artificial Intelligence (IJCAI), Acapulco, Mexico, pp. 1531-1533

[6] Hart, P. E., Nilsson, N. J. and Raphael, B. (1968). A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE Transactions on Systems Science and Cybernetics 4(2), pp. 100-107

[7] Korf, R. E. (1990). Real-Time Heuristic Search. Artificial Intelligence 42(2, 3), pp. 189-211

[8] Luštrek, M., Bratko, I. and Gams, M. (2005). Why Minimax Works: An Alternative Explanation. Proceedings of International Joint Conference on Artificial Intelligence (IJCAI), Edinburgh, UK

[9] Nau, D. S. (1979). Quality of Decision versus Depth of Search on Game Trees. Ph. D. thesis, Duke University

[10] Nau, D. S. (1982). An Investigation of the Causes of Pathology in Games. Artificial Intelligence 19(3), pp. 257-278

[11] Scheucher, A. and Kaindl, H. (1998). Benefits of Using Multivalued Functions for Minimaxing. Artificial Intelligence 99(2), pp. 187-208