Thinking Too Much: Pathology in Pathfinding

Mitja Luštrek [1] and **Vadim Bulitko** [2]

Abstract. Large real-time search problems, such as pathfinding in computer games and robotics, limit the applicability of complete search methods, such as A*. Real-time heuristic search methods are better suited to these problems. They typically conduct a limited-depth lookahead search and evaluate the states at the frontier using a heuristic. Actions selected by such methods can be suboptimal due to the incompleteness of their search and inaccuracies in the heuristic. Lookahead pathology occurs when deeper search results in worse actions. Over the last three decades the research on the pathology has focused on minimax search and small synthetic examples in single-agent search. This paper conducts a large-scale investigation of the pathology in real-time pathfinding on maps from commercial computer games, finding it quite common. Four explanations for the pathology are provided and supported empirically.

1 INTRODUCTION

Pathfinding tasks commonly require real-time response, which on large problems precludes the use of complete search methods, such as A*. Imagine a real-time strategy computer game. The player commands dozens or even hundreds of units to move towards a distant goal. If each of the units were to compute the whole path before moving, this could easily incur a noticeable delay. Incomplete single-agent search methods [10, 20, 8, 5, 7] work similarly to minimax-based methods used in two-player games. They conduct a limited-depth lookahead search, i.e., expand a part of the space centered on the agent, and heuristically evaluate the distances from the frontier of the expanded space to the goal. By interleaving planning and plan execution, they can guarantee a constant upper bound on the amount of planning per move and compare favorably to complete methods [9].

Actions selected based on heuristic lookahead search are not necessarily optimal, but it is generally believed that deeper lookahead increases the quality of decisions. However, in two-player games it has long been known that this is not always the case [15, 1]. The phenomenon has been termed minimax *pathology*. It has attracted considerable interest over the years and turned out to be rather difficult to explain. Pathological behavior in single-agent search was discovered much later [6, 3] and very little has been known about it so far. This paper endeavors to remedy the situation.

In the paper we investigate lookahead pathology in real-time pathfinding on maps from commercial computer games. Computer games are a domain where the pathology is particularly undesirable: they require fast response, so it is crucial that more expensive, deeper lookahead is not conducted when it is not beneficial or, worse yet, when it is harmful. Game company Bioware Corp., for example, limits its planning time to 1–3 ms for all pathfinding units (many units can be planning simultaneously) [7]. Our paper makes two main contributions. First, it presents an empirical study of the pathology in pathfinding. In this study, a degree of pathology was found in *over 90%* of the problems considered. Second, it explains how both learning and planning are responsible for such wide-spread pathological behavior. Finally, it discusses how the pathology can be prevented.

[1] Jožef Stefan Institute, Department of Intelligent Systems, Jamova cesta 39, 1000 Ljubljana, Slovenia, email: mitja.lustrek@ijs.si

[2] University of Alberta, Department of Computing Science, Edmonton, Alberta T6G 2E8, Canada, email: bulitko@ualberta.ca

2 RELATED WORK

Most game-playing programs successfully use minimax to back-up the heuristic values from the leaves of the game tree to the root. Early attempts to explain why backed-up values are more reliable than static values, however, led to a surprising discovery: minimaxing amplifies the error of the heuristic evaluations [15, 1]. Several explanations for such pathological behavior were proposed, the most common being that the dependence between nearby positions is what eliminates the pathology in real games [2, 16, 19, 14]. It was recently shown that modeling the error realistically might be enough to eliminate it [12]. Common to all the work on the minimax pathology is that it tries to explain why the pathology appears in theoretical analyses but not in practice, whereas in pathfinding, the pathology is a practical problem.

The pathology in single-agent search was discovered by Bulitko et al. [6] and was demonstrated on a three-level synthetic search tree. It was also observed in solving the eight-puzzle [3]. Luštrek [11] used small synthetic search trees to explain how certain properties of the trees affect the pathology. He also showed that consistent and admissible heuristics prevent the pathology. In contrast, Sadikov and Bratko [17] observed in eight-puzzle that the pathology is prevented by pessimistic heuristics (i.e., the opposite of admissible). None of this research translates well to pathfinding, though. The only study of lookahead pathology in pathfinding that we are aware of is our earlier workshop paper [13]. This paper offers several new insights into the problem.

Some work [13, 4] was done on dynamic selection of optimal lookahead depth, which is a mechanism that – if successful – eliminates the pathology. The best performance was achieved by precomputing the optimal depths for pairs of (start, goal) states. Such an approach is not always practical, though, and if the pathology were better understood, alternatives might be found.

3 PROBLEM FORMULATION

In this paper we study the problem of an agent trying to find a path from a start state to a goal state in a two-dimensional grid world. The agent's state is defined by its location in one of the grid's squares. Some squares are blocked by obstacles and cannot be traversed. The traversable squares form the set of states S the agent can occupy;

$s_g \in S$ is the goal state. From each square the agent can move to the eight immediate neighbor squares. The *travel cost* of each of the four straight moves (north, west, south and east) is 1 and the travel cost of the diagonal moves is $\sqrt{2}$. A search problem is defined by a map, a start state and a goal state.

The agent plans its path using the Learning Real-Time Search (LRTS) algorithm [5] configured as straightforwardly as possible (heuristic weight of 1, no backtracking). LRTS conducts a lookahead search centered on the current state $s_c \in S$ and generates all the states within its lookahead area (i.e., up to d moves away from s_c; d is called the lookahead depth). The term *generate* refers to 'looking at' a state, as opposed to physically visiting it. Next, LRTS evaluates the states at the frontier of the lookahead area using the heuristic function h, which estimates the length of the shortest path from the frontier states to s_g. The agent then moves along the shortest path to the most promising frontier state $s_{f\,opt} \in S$. State $s_{f\,opt}$ is the state on the path to s_g with the lowest projected travel cost $f(s_{f\,opt}) = g(s_{f\,opt}) + h(s_{f\,opt})$, where $g(s_{f\,opt})$ is the travel cost from s_c to $s_{f\,opt}$ and $h(s_{f\,opt})$ the estimated travel cost from $s_{f\,opt}$ to s_g. When the agent reaches $s_{f\,opt}$, another lookahead search is performed. This process continues until s_g is reached. The initial heuristic is the *octile distance* to s_g, i.e., the actual shortest distance assuming a map without obstacles (the term octile refers to the eight squares or tiles the agent can move to). After each lookahead search, $h(s_c)$ is updated to $f(s_{f\,opt})$ if the latter is higher, which raises the heuristic in the areas where it was initially too optimistic, making it possible for the agent to eventually find the way around the obstacles even when the lookahead area is too small to see the way directly. This constitutes the process of learning. The updated heuristic is still guaranteed to be admissible (i.e., optimistic), although unlike the initial one, it may not be consistent (i.e., the difference in the heuristic values of two states may exceed the shortest distance between them).

The LRTS algorithm is similar to the classic LRTA* [10]. The main difference is that LRTS conducts a lookahead search only after reaching the frontier of the previous lookahead area, whereas LRTA* searches after every move. LRTS typically finds longer solutions than LRTA*, but generates fewer states to do so. LRTA* was selected because it is more prone to the pathology and as such more interesting for our research. The two algorithms will be further compared in Sections 5 and 6.

Definition 1 For a state $s \in S$, the *length of the solution* (i.e., the travel cost of the path from s to s_g) produced by LRTS with lookahead depth d is denoted by $l(s, d)$.

Definition 2 For a state $s \in S$, the *solution-length vector* $\vec{l}(s)$ is $(l(s,1), l(s,2), \ldots, l(s,d_{\max}))$, where $d_{\max}$ is the maximum lookahead depth.

Definition 3 The degree of *solution-length pathology* of a problem with the start state s_0 is k iff $\vec{l}(s_0)$ has exactly k increases in it, i.e., $l(s_0, i+1) > l(s_0, i)$ for k different i.

The length of the solution is the most natural way to measure the performance of a search algorithm. It can be used in *on-policy* experiments, where the agent follows a path from the start state to the goal state as directed by the algorithm. However, we also conducted *off-policy* experiments, in which the agent spontaneously appears in (randomly selected) states and selects the first move towards the goal state. Such experiments require a different measure of performance.

Definition 4 In a state $s \in S'$, where $S' \subseteq S$ is the set of states visited by the agent, the agent can take an optimal or a suboptimal action. An optimal action moves the agent into a state lying on a lowest-cost path from s to s_g. The *error* $e(S', d)$ is the fraction of suboptimal actions taken in the set of states S' using lookahead depth d.

Definition 5 $\mathcal{S} = (S_1, S_2, \ldots S_{d_{\max}})$, where $S_i \subseteq S$ for all i and $d_{\max}$ is the maximum lookahead depth, is the sequence of sets of states visited by the agent using successive lookahead depths. The *error vector* $\vec{e}(\mathcal{S})$ is $(e(S_1,1), e(S_2,2), \ldots, e(S_{d_{\max}}, d_{\max}))$.

Definition 6 The degree of *error pathology* of a sequence of sets of states $\mathcal{S} = (S_1, S_2, \ldots S_{d_{\max}})$ is k iff $\vec{e}(\mathcal{S})$ has exactly k increases in it, i.e., $e(S_{i+1}, i+1) > e(S_i, i)$ for k different i. We can also speak of the 'pathologicalness' of a single state, but in this case the notion of the degree of pathology makes little sense. A state $s \in S$ is pathological iff there are lookahead depths i and $i+1$ such that the action selected in s using lookahead depth i is optimal and the action selected using depth $i+1$ is not.

Finally, we will need ways to measure the amount of work done by the LRTS algorithm.

Definition 7 $\alpha(d)$ is the average number of states generated per single move during lookahead search with depth d. $\alpha'(d)$ is the average number of states generated per problem.

Definition 8 $\beta(d)$ is the average volume of updates to the heuristic encountered per state generated during lookahead search with depth d. The volume of updates is the difference between the updated and the initial heuristic.

4 PATHOLOGY EXPERIMENTALLY OBSERVED

We experimented on five maps from a commercial role-playing game. The maps were loaded into Hierarchical Open Graph [7], an open-source research testbed. Their sizes ranged from 214×192 (2,765 states) to 235×204 (16,142 states). On these maps, we randomly generated 1,000 problems. In order for the problems to be sufficiently and uniformly difficult, the true distance between the start and the goal state was always between 90 and 100. The lookahead depth ranged from 1 to 10. The start state was only used in on-policy experiments. In off-policy experiments, the states to visit were selected randomly and were the same for all depths. The effect of learning in off-policy experiments depends on the order of visitation and since no meaningful order is apparent, learning was omitted. Since it is not possible to measure solution-length pathology in off-policy experiments, the experiments are compared with respect to error pathology. It is in all cases similar to solution-length pathology. In order to measure the error, all the problems were solved optimally.

First we conducted the basic on-policy experiment: the agent solved the 1,000 problems, we measured the degree of solution-length and error pathology for each problem and counted the number of problems with each of the possible degrees (from 0, which means no pathology, to 9, which means that deeper lookahead is always worse). The results in Table 1 show that *over 90%* of the problems are pathological. Even though the degree of pathology is in many cases low, this still means that one cannot simply rely on the conventional wisdom that deeper search is better.

The first possible explanation of the results in Table 1 is that the maps contain a lot of states where deeper lookaheads lead to suboptimal decisions, whereas shallower ones do not. This turned out *not* to

Table 1. Percentages of pathological problems with respect to solution-length and error pathology in the basic on-policy experiment.

Degree	0	1	2	3	4	≥ 5
Length pathology [%]	7.2	15.6	32.3	26.9	13.8	4.2
Error pathology [%]	6.3	13.1	24.8	29.0	18.1	8.7

be the case: for each of the problems we measured the 'pathological-ness' of every state on that map in the off-policy mode. Surprisingly, only 3.9% of the states turned out to be pathological. This is disconcerting: if the nature of the pathfinding problems is not pathological, yet there is a lot of pathology in our solutions, then perhaps the search algorithm is to blame.

Comparing the percentage of pathological states to the results in Table 1 is not entirely fair, because in the first case the pathology is considered per state and in the second case it is computed from the error averaged over all the states visited per problem. The average number of states visited per problem during the basic on-policy experiment is 485. To make the comparison fairer, we randomly selected the same number of states per problem in the off-policy mode and computed the average error over them. Pathology measurements from the resulting error vector are shown in Table 2. We will call this experiment the basic off-policy experiment.

Table 2. Percentages of pathological problems in the basic off-policy experiment.

Degree	0	1	2	≥ 3
Error pathology [%]	83.1	14.9	2.0	0.0

Measuring the off-policy pathology per problem reduces the discrepancy between on-policy and off-policy experiments from 92.8% vs. 3.9% to 92.8% vs. 16.9%, which is still a lot. This suggests that pathological states are not the main reason for on-policy pathology, because if they were, their effect in on-policy and off-policy experiments should be comparable. In the rest of the paper, we will investigate why there is so much more pathology on-policy than off-policy.

5 EXPLANATIONS OF THE PATHOLOGY

The simplest explanation for the pathology in the basic on-policy experiment is as follows:

Explanation 1 The LRTS algorithm's behavioral policy tends to steer the search to pathological states.

The explanation can be verified by computing off-policy pathology from the error in the states visited during the basic on-policy experiment instead of randomly selected 485 states. This experiment differs from the basic on-policy experiment in that the error is measured in the same states at all lookahead depths (in on-policy experiments, different states may be visited at different depths) and there is no learning. The results in Table 3 do show a larger percentage of pathological problems compared to the basic off-policy experiment in Table 2 (23.2% vs. 16.9%), but they are still far from the basic on-policy experiment in Table 1 (23.2% vs. 93.7%). So while the percentage of pathological states visited on-policy is somewhat above average, this cannot account for the large fraction (93.7%) of pathological problems in the basic on-policy experiment.

Table 3. Percentages of pathological problems measured off-policy in the states visited on-policy.

Degree	0	1	2	3	4	≥ 5
Error pathology [%]	76.8	13.8	5.7	2.3	1.0	0.4

We know that the basic on-policy experiment involves learning (i.e., updating the heuristic function), whereas the basic off-policy experiment does not. The agent performs a search with lookahead depth d every d moves. If the map were empty, the agent would move in a straight line and would encounter exactly one updated state during each search (the one updated during the previous search). Each search generates $(2d+1)^2$ distinct states, so $1/(2d+1)^2$ of them would have been updated: a fraction that is larger for smaller d. This leads us to:

Explanation 2 Smaller lookahead depths benefit more from the updates to the heuristic. This can be expected to make their decisions better than the mere depth would suggest and thus closer to larger depths. If they are closer to larger depths, cases where a deeper lookahead actually performs worse than a shallower one should be more common.

A first test of Explanation 2 is to perform an on-policy experiment where the agent is still directed by the LRTS algorithm that uses learning (as without learning it would often get caught in an infinite loop), but the measurement of the error is performed using only the initial, non-updated heuristic. To do this, two moves are selected in each state: one that uses learning, which the agent actually takes, and another that ignores learning, which is used for error measurement. The results in Table 4 suggest that learning is indeed responsible for the pathology, because the pathology in the new experiment is markedly smaller than in the basic on-policy experiment shown in Table 1: 70.4% vs. 93.7%.

Table 4. Percentages of pathological problems on-policy with error measured without learning.

Degree	0	1	2	3	4	≥ 5
Error pathology [%]	29.6	20.4	19.3	18.2	8.5	4.0

During the basic off-policy experiment, all lookahead depths are on an equal footing with respect to learning as there are no updates to the heuristic. Since learning is inevitable in on-policy experiments, the best one can do to put all the depths on an equal footing during an on-policy experiment is to have all the states within the lookahead area updated as uniformly as possible. We implement such uniform learning as follows. Let s_c be the current state, S_i the set of all the interior states of the lookahead area and S_f the frontier of the lookahead area. For every interior state $s \in S_\mathrm{i}$, an LRTS search originating in s and extending to the frontier S_f is performed. The heuristic value of s is then updated to the travel cost of the shortest path to the goal found during the search, just like the heuristic value of s_c is with the regular update method. The results for an on-policy experiment with uniform learning are found in Table 5. Unfortunately they seem to contradict Explanation 2, since they are about as pathological as the results of the basic on-policy experiment shown in Table 1: 93.4% vs. 92.8% for solution-length pathology and 93.3% vs. 93.7% for error pathology.

Table 5. Percentages of pathological problems on-policy with uniform learning.

Degree	0	1	2	3	4	≥ 5
Length pathology [%]	6.6	16.6	30.9	27.4	15.4	3.1
Error pathology [%]	6.7	17.9	33.9	28.7	10.5	2.3

In our next attempt to confirm Explanation 2, which was also intended to clear the apparent contradiction of uniform learning, we measured the volume of heuristic updates encountered during search. Figure 1 shows the results for the regular and uniform learning, as well as for the basic off-policy experiment, where no learning takes place. We see that the volume of updates decreases with increased

lookahead depth in both on-policy experiments (unlike in the basic off-policy experiment). This indicates that the impact of learning is larger at smaller depths, which is exactly what Explanation 2 claims, even though the reasoning that led to the explanation may not be entirely correct. Figure 1 also explains why uniform learning is no less pathological than the regular one: the volume of updates with uniform learning decreases more steeply than with the regular learning, so if anything, uniform learning should be more pathological. The comparably small volume of updates with uniform learning is probably caused by the agent finding the goal state more quickly: the length of the solution averaged over all the problems is 2.5–4.3 times shorter than with the regular learning (for depths > 1, since both types of learning are the same at depth 1). Shorter solutions mean that the agent returns to the areas already visited (where the heuristic is updated) less often.

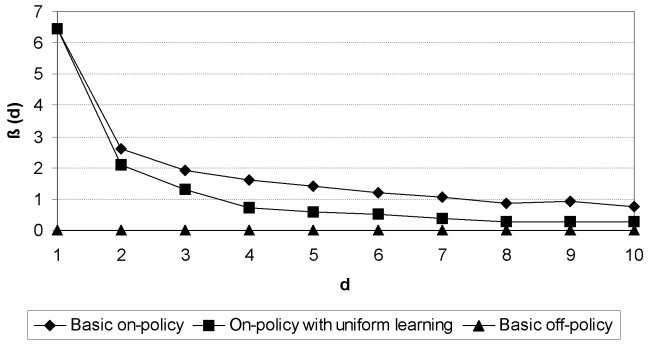

Figure 1. The volume of heuristic updates encountered per move with respect to the lookahead depth in different experiments.

Unlike the regular learning, uniform learning preserves the consistency of the heuristic. Experiments on synthetic search trees suggested that inconsistency increases the pathology [11], but since the consistent uniform learning is as pathological as the inconsistent regular learning, this appears not to be the case in pathfinding.

Explanation 3 Let $\alpha_{\text{off}}(d)$ and $\alpha_{\text{on}}(d)$ be the average number of states generated per move in the basic off-policy and on-policy experiments respectively. In off-policy experiments a search is performed after every move, whereas in on-policy experiments a search is performed every d moves. Therefore $\alpha_{\text{on}}(d) = \alpha_{\text{off}}(d)/d$ (assuming the same number of non-traversable squares). This means that in the basic on-policy experiment fewer states are generated at larger lookahead depths than in the basic off-policy experiment. Consequently the depths in the basic on-policy experiment are closer to each other with respect to the number of states generated. Since the number of states generated can be expected to correspond to the quality of decisions, cases where a deeper lookahead actually performs worse than a shallower one should be more common.

Explanation 3 can be verified by an on-policy experiment where a search is performed after every move instead of every d moves. The results in Table 6 confirm the explanation. The percentage of pathological problems is considerably smaller than in the basic on-policy experiment shown in Table 1: 39.8% vs. 92.8% for solution length pathology and 34.7% vs. 93.7% for error pathology. Since LRTS that searches every move is very similar to LRTA* [10], LRTA* can also be expected to be less pathological. We will talk more about the relative merits of the two algorithms in Section 6.

Explanation 3 can be further tested as follows. Figure 2 shows that in the basic off-policy experiment and in the on-policy experiment when searching every move, the number of states generated per move increases more quickly with increased lookahead depth than in the basic on-policy experiment. This means that the depths are less similar than in the basic on-policy experiment, which again confirms Explanation 3.

Table 6. Percentages of pathological problems in an on-policy experiment when searching every move.

Degree	0	1	2	3	4	≥ 5
Length pathology [%]	60.2	18.6	11.1	5.0	3.0	2.1
Error pathology [%]	65.3	14.6	8.6	7.1	3.0	1.4

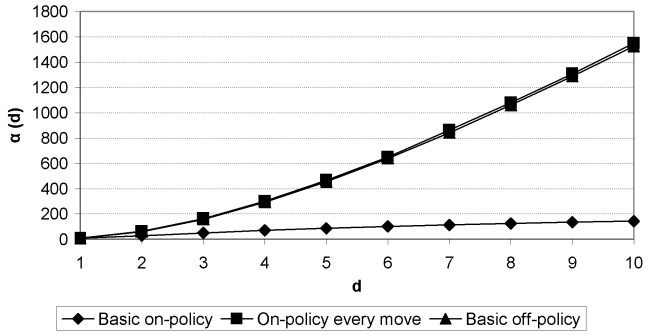

Figure 2. The number of states generated per move with respect to the lookahead depth in different experiments.

Uniform learning already showed that consistency does not prevent the pathology like it does in synthetic search trees [11]. There are conflicting accounts regarding admissibility, though: it decreases the pathology in synthetic search trees [11] and increases it in eight-puzzle [17]. The latter seems to be more in line with what happens in pathfinding:

Explanation 4 During lookahead search, states with low heuristic values are favored. If the heuristic values are admissible (i.e., lower than the true values), the lowest heuristic value is likely to be particularly far from the true value. With deeper lookahead, more states are considered and the chances of selecting a state with an especially inaccurate heuristic value are increased. If the heuristic values are pessimistic (i.e., higher than the true values), the opposite is true: the states with accurate heuristic values are favored and the more states are considered, the more likely a state with a very accurate heuristic value will be selected. So as far as the accuracy of heuristic values is concerned, deeper lookahead is beneficial only with pessimistic heuristics. Therefore pessimistic heuristics can be expected to decrease the pathology compared to optimistic ones.

We verified Explanation 4 in an on-policy experiment with pessimistic heuristic values. If the regular heuristic value of a state s is $h(s) = h^*(s) - e$, where e is the heuristic error, then the pessimistic heuristic value is $h_{\text{p}}(s) = h^*(s) + e$. Such a heuristic is of course unrealistic, but its absolute error is the same as the absolute error of the regular heuristic, so it is suitable for comparing the two. It should give us an idea of what to expect from realistic pessimistic heuristics, should we be able to design them. The results in Table 7 do show a decrease in pathology compared to the basic on-policy experiment shown in Table 1: 86.2% and 86.1% vs. 92.8% and 97.7% for solution-length and error pathology respectively. The results are not entirely conclusive, though, because the pessimistic heuristic causes a greater amount of severe pathology (degree ≥ 4) than the optimistic one.

Table 7. Percentages of pathological problems in an on-policy experiment with pessimistic heuristic.

Degree	0	1	2	3	4	≥ 5
Length pathology [%]	13.8	3.5	7.0	20.9	26.4	28.4
Error pathology [%]	13.9	4.1	8.3	22.9	27.7	23.1

6 DISCUSSION

Let us review the four explanations and consider whether they can be used to avoid the pathology in real-time pathfinding.

Explanation 1 tells us that 23.2% of the states the LRTS algorithm visits are pathological instead of the average 16.9%. Unfortunately we do not know why the algorithm favors pathological states, so we cannot propose a remedy. However, since the difference is small, such a remedy would probably not help much.

Explanation 2 states that turning off learning reduces the pathology from 93.7% to 70.4%, which means that learning is an important source of the pathology. This is very difficult to avoid, though. Shallower lookaheads cause the agent to revisit states more often, which by necessity results in greater benefit of learning. Any effective attempt to counter this by increasing the benefit of learning at deeper lookaheads will result in fewer revisits. Fewer revisits will in turn reduce the benefit of learning. Our uniform learning is an example of this.

Explanation 3 probably shows the most promise for practical reduction of the pathology. Searching every move the way LRTA* does brings the pathology from 92.8%–93.7% to 34.7%–39.8%. It also generates 1.5–2.6 times shorter solutions (for lookahead depths > 1). However, it increases the number of states generated per move roughly by a factor of d (in our on-policy experiments by a factor of 10.8 at $d = 10$). Figure 3 illustrates the tradeoff: it shows the number of states generated *per problem* for the basic on-policy experiment (regular LRTS) and for the on-policy experiment when searching every move (LRTA* style). Compared to Figure 2 (which shows the number of states generated per move), the difference between searching every d moves and searching every move is smaller because of the shorter solutions in the second case, but the regular LRTS still generates up to 4.5 times fewer states per problem.

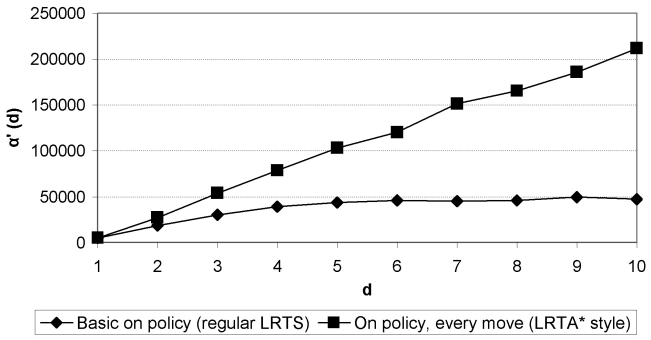

Figure 3. The number of states generated per problem with respect to the lookahead depth in different experiments.

Finally, Explanation 4 suggests that pessimistic heuristics may be less prone to the pathology. In addition, the solutions found using the pessimistic heuristic were nearly optimal (3.8–7.2 times shorter than with the regular heuristic). Since our pessimistic heuristic was not realistic, we cannot draw any firm conclusions from this result, but we can certainly say that pessimistic heuristics warrant further investigation.

7 CONCLUSION AND FUTURE WORK

This paper made two main contributions: it demonstrated that lookahead pathology is a common phenomenon in real-time pathfinding and it provided four complementary empirically supported explanations for it.

There are two practical conclusions to be drawn from the explanations of the pathology. First, it is dangerous to commit to a segment of path of length d after a lookahead search of depth d as the LRTS algorithm does. But it is also wasteful to search after every move as LRTA* does. This suggests that some kind of middle ground should be looked for, ideally a method to determine at what point exactly a new search is needed. And second, pessimist heuristics seem to be well suited to real-time pathfinding, which is a confirmation of recent results for combinatoric puzzles [17, 18].

REFERENCES

[1] Donald F. Beal, 'An analysis of minimax', in *Advances in Computer Chess*, volume 2, pp. 103–109, (1980).

[2] Ivan Bratko and Matjaž Gams, 'Error analysis of the minimax principle', in *Advances in Computer Chess*, volume 3, pp. 1–15, (1982).

[3] Vadim Bulitko, 'Lookahead pathologies and meta-level control in real-time heuristic search', in *Proceedings of the 15th Euromicro Conference on Real-Time Systems*, pp. 13–16, Porto, Portugal, (2003).

[4] Vadim Bulitko, Yngvi Bjornsson, Mitja Luštrek, Jonathan Schaeffer, and Sverrir Sigmundarson, 'Dynamic control in path-planning with real-time heuristic search', in *Proceedings of ICAPS*, pp. 49–65, (2007).

[5] Vadim Bulitko and Greg Lee, 'Learning in real time search: A unifying framework', *Journal of Artificial Intelligence Research*, **25**, 119–157, (2006).

[6] Vadim Bulitko, Lihong Li, Russell Greiner, and Ilya Levner, 'Lookahead pathologies for single agent search', in *Proceedings of IJCAI, poster section*, pp. 1531–1533, Acapulco, Mexico, (2003).

[7] Vadim Bulitko, Nathan Sturtevant, Jieshan Lu, and Timothy Yau, 'Graph abstraction in real-time heuristic search', *Journal of Artificial Intelligence Research*, **30**, 51–100, (2007).

[8] Carlos Hernández and Pedro Meseguer, 'LRTA*(k)', in *Proceedings of IJCAI*, pp. 1238–1243, Edinburgh, UK, (2005).

[9] Sven Koenig, 'A comparison of fast search methods for real-time situated agents', in *Proceedings of AAMAS*, volume 2, pp. 864–871, (2004).

[10] Richard E. Korf, 'Real-time heuristic search', *Artificial Intelligence*, **42**(2, 3), 189–211, (1990).

[11] Mitja Luštrek, 'Pathology in single-agent search', in *Proceedings of Information Society Conference*, pp. 345–348, Ljubljana, Slovenia, (2005).

[12] Mitja Luštrek, Ivan Bratko, and Matjaž Gams, 'Why minimax works: An alternative explanation', in *Proceedings of IJCAI*, pp. 212–217, Edinburgh, UK, (2005).

[13] Mitja Luštrek and Vadim Bulitko, 'Lookahead pathology in real-time path-finding', in *Proceedings of AAAI, Learning for Search Workshop*, pp. 108–114, Boston, USA, (2006).

[14] Mitja Luštrek, Matjaž Gams, and Ivan Bratko, 'Is real-valued minimax pathological?', *Artificial Intelligence*, **170**(6–7), 620–642, (2006).

[15] Dana S. Nau, *Quality of Decision versus Depth of Search on Game Trees*, Ph.D. dissertation, Duke University, 1979.

[16] Judea Pearl, 'On the nature of pathology in game searching', *Artificial Intelligence*, **20**(4), 427–453, (1983).

[17] Aleksander Sadikov and Ivan Bratko, 'Pessimistic heuristics beat optimistic ones in real-time search', in *Proceedings of ECAI*, pp. 148–152, Riva del Garda, Italy, (2006).

[18] Aleksander Sadikov and Ivan Bratko, 'Solving 20x20 puzzles', in *Proceedings of Computer Games Workshop*, pp. 157–164, Amsterdam, The Netherlands, (2007).

[19] Anton Scheucher and Hermann Kaindl, 'Benefits of using multivalued functions for minimaxing', *Artificial Intelligence*, **99**(2), 187–208, (1998).

[20] L.-Y. Shue, S.-T. Li, and R. Zamani, 'An intelligent heuristic algorithm for project scheduling problems', in *Proceedings of the 32nd Annual Meeting of the Decision Sciences Institute*, San Francisco, USA, (2001).