Pessimistic Heuristics Beat Optimistic Ones in Real-Time Search.

Conference Paper · January 2006

Source: DBLP

CITATIONS
9

READS
566

2 authors:

Aleksander Sadikov
University of Ljubljana
82 PUBLICATIONS **713** CITATIONS
SEE PROFILE

Ivan Bratko
University of Ljubljana
213 PUBLICATIONS **5,060** CITATIONS
SEE PROFILE

Pessimistic Heuristics Beat Optimistic Ones in Real-Time Search

Aleksander Sadikov and Ivan Bratko[1]

Abstract. Admissibility is a desired property of heuristic evaluation functions, because when these heuristics are used with complete search methods, such as A* and RBFS, they guarantee that an optimal solution will be found. Since every optimistic heuristic function is admissible, optimistic functions are widely used. We show, however, that with incomplete, real-time search, optimistic functions lose their appeal, and in fact they may hinder the search under quite reasonable conditions. Under these conditions the exact opposite is to be preferred, i.e. pessimistic heuristic functions that never *underestimate* the difficulty of the problem. We demonstrate that such heuristics behave better than optimistic ones of equal quality on a standard testbed using RTA* search method.

1 INTRODUCTION

Admissibility is a desired property of heuristic evaluation functions, because when these heuristics are used with complete search methods, such as A* [6] and RBFS [8], they guarantee that an optimal solution will be found. A known theorem about the admissibility states that a heuristic function, which always gives an optimistic assessment of the position, is admissible [6]. This theorem made optimistic heuristic functions popular and widely used. The main problems with complete search methods, though, are their exponential running time and the necessity to wait until the search completes before the first step towards the goal can be taken. To alleviate these problems, incomplete search methods have been proposed, such as RTA* [7]. Since these methods can tackle much larger problems, they can often be the only practical option.

Incomplete search methods do not guarantee finding an optimal solution even when used in conjunction with admissible heuristics. Thus, the main reason for using admissible and consequently optimistic heuristics is void. Nevertheless, people use optimistic heuristic functions with incomplete search methods, because: (a) they are often readily available, since they were developed for complete search algorithms, (b) common sense saying that since they proved useful with complete search methods, perhaps they are useful with incomplete search methods as well, and (c) it was never shown that they could be counterproductive.

In this paper we show that under very reasonable conditions, optimistic functions are counterproductive in real-time (incomplete) search. Our results indicate that pessimistic functions should be used instead.

Section 2 of the paper derives the condition under which optimistic heuristics harm real-time search. It also shows that pessimistic heuristics behave much better under these same conditions. An intuitive explanation for this phenomenon is given. Section 3 compares pairs of heuristic functions of equal quality, one optimistic, the other pessimistic, and demonstrates that the latter give much better results when used with RTA* search in the 8-puzzle domain. Section 4 discusses some consequences of this finding. Section 5 concludes the paper and gives some pointers for further work.

[1] University of Ljubljana, Faculty of Computer and Information Science, Artificial Intelligence Laboratory, Tržaška 25, 1000 Ljubljana, Slovenia, e-mail: {aleksander.sadikov;ivan.bratko}@fri.uni-lj.si

2 OPTIMISTIC AND PESSIMISTIC HEURISTICS

Let RTA* evaluate node n with an evaluation function f of the common form $f(n) = g(n) + h(n)$. Here $g(n)$ is the cost of the path from the current node to node n, and $h(n)$ is a heuristic estimate of the cost of an optimal path from node n to the nearest goal node. Let us denote the true cost of such an optimal path from node n with $h^*(n)$. The relationship between the true and heuristic value of node n is governed by the following equation:

$$h(n) = h^*(n) + e(n) \tag{1}$$

where $e(n)$ is an error the heuristic makes at node n.

Suppose the current state has two successors, nodes a and b, a being better than b. The immediate task of the search is to choose between node a and node b. RTA* will choose a if $f(a) < f(b)$. We will simplify the analysis slightly by assuming without loss that all the edges have unit cost. Then the condition $f(a) < f(b)$ for choosing a is equivalent to the condition $h(a) < h(b)$. If this condition holds then RTA* will make the correct decision (i.e. choose a), otherwise it will commit a decision error by choosing b. So a decision error occurs when

$$h(a) > h(b). \tag{2}$$

This can be rewritten in terms of true costs h^* and heuristic errors e:

$$h^*(a) + e(a) > h^*(b) + e(b). \tag{3}$$

Since a is better than b, $h^*(a) < h^*(b)$, and the difference $\Delta h^* = h^*(b) - h^*(a)$ is positive. The condition for decision error can be stated as:

$$e(a) - e(b) > \Delta h^* > 0. \tag{4}$$

Now let us consider what happens in two special cases: (1) when the algorithm uses an *optimistic* heuristic function, and (2) when it uses a *pessimistic* heuristic function. By definition, for optimistic heuristics, heuristic errors e are always negative because heuristic approximations underestimate true costs. So for optimistic heuristics, for all nodes n, $e(n) = -|e(n)|$. The decision error condition then becomes:

$$|e(b)| - |e(a)| > \Delta h^* > 0. \tag{5}$$

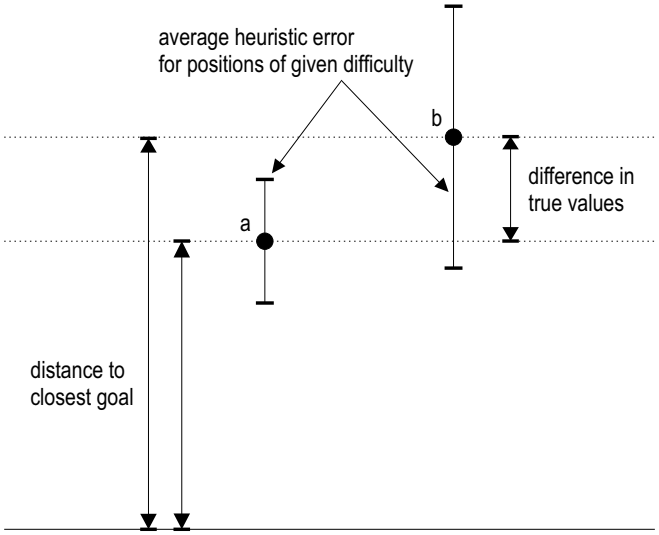

Figure 1. The difference in true values

Equation 5 has an interesting consequence. It basically says that if the heuristic function's absolute error increases with increasing difficulty of the problems (nodes requiring more costly paths to reach the goal), then for optimistic heuristics the chance to make a decision error *increases*. In our example, node b is more difficult than a, so under this assumption, absolute heuristic error at b will tend to be greater than at a. This indicates a problem with optimistic heuristics when heuristic errors (in absolute terms) increase with the difficulty of the problems. Conversely, if heuristic errors decrease with the difficulty of the problems, then the optimistic heuristic will have better chance of making correct decisions. However, as also discussed later in the paper, this second case (heuristic error decreasing with the cost) seems to be rather unrealistic in practice.

For pessimistic heuristics, by definition heuristic errors e are always positive, and the condition for RTA* making a decision error is:

$$|e(a)| - |e(b)| > \Delta h^* > 0. \tag{6}$$

If we compare Equations 5 and 6, we can see that just the opposite of what is true for optimistic heuristics holds for pessimistic heuristics. Pessimistic heuristics have better chances to produce correct decision if the heuristic error increases with increasing difficulty of the nodes, and worse if the heuristic error decreases with increasing difficulty of the nodes. According to this, in the case of absolute error increasing with difficulty, i.e. the case to be expected more likely in practice, optimistic heuristics will tend to make decision errors more frequently than pessimistic heuristics.

2.1 Intuitive explanation

Figure 1 shows a decision task we discussed. The tails stemming from nodes a and b represent the average heuristic error for the problems of given difficulty. To commit a decision error, the heuristic values have to overcome the difference in true values between the two nodes. Suppose the heuristic's errors increase with increasing difficulty of the problems. Since node b represents a more difficult problem, the heuristic will on the average make a larger error in it. This is represented in the figure with a longer tail coming out of node b than the one coming out of node a. Furthermore, if the heuristic is optimistic, only downward part of the tail is possible, and vice versa, if the heuristic is pessimistic only the upward part of the tail is possible. If the heuristic is optimistic, the longer tail, coming out of node b has to overcome the difference in true values to give a chance for a decision error to be committed. However, if the heuristic is pessimistic, it is the shorter tail coming out of node a that has to overcome the difference in true values. Since the longer tail can more easily overcome this difference than the shorter tail, it follows that the optimistic heuristic can more easily commit a decision error when the heuristic's errors increase with increasing difficulty of the problems.

3 EXPERIMENTS

We have tested our theoretical results on a classical testbed for single-agent search methods, the 8-puzzle sliding tiles problem, described e.g. in [2]. We have chosen this small puzzle, because for complete evaluation of success of search we needed to know the true value of every state in the domain — the reason for this will become apparent in the next subsection.

3.1 The two heuristic functions

To empirically test our theoretical findings, we compared two heuristic functions, one optimistic, the other pessimistic. The functions needed to be of equal quality (in terms of their relative errors), so that neither one of them has an unfair advantage. To get such a pair of functions, we had to artificially construct them.

We first calculated the true values of all legal 8-puzzle positions with the use of retrograde analysis, a technique known from computer chess, where it is used to generate endgame databases [13]. An indexed array of distances to nearest goal defines the perfect evaluation function h^*. Then we proceeded to generate the two heuristic functions by appropriately corrupting our perfect heuristics.

We modelled the distribution of our optimistic heuristic after the distribution of Manhattan distance heuristic — a well-known optimistic heuristic for the 8-puzzle domain. On average, Manhattan distance heuristic's error increases with increasing difficulty of the problems (the average error over all positions of a given difficulty level). We measure the difficulty of the problem as the number of steps needed to reach the goal assuming optimal play, i.e. with the problem's true value. The dispersion of heuristic values around the average evaluation for a given difficulty level is more or less constant.

We created our artificial heuristics by corrupting the perfect evaluations in a controlled manner. Our method of doing this is as follows. We take a position's true value $h^*(n)$ and add to it a certain amount of Gaussian noise, described with the formula:

$$\mathrm{P}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/(2\sigma^2)}. \tag{7}$$

The formula gives the probability $\mathrm{P}(x)\mathrm{d}x$ that given the correct evaluation $\mu = h^*(n)$ and standard deviation σ, the heuristic evaluation, $h(n) = x \in \mathbf{R}$, will take on a value in the range $[x, x + \mathrm{d}x]$. The error of heuristic evaluation $h(n)$ is $e(n) = x - \mu$. We do this for all legal positions. A more detailed description of this process is given in [11]. Parameter σ controls the level of corruption. Since we modelled our heuristics after the Manhattan heuristic, we chose $\sigma = 2.5$ steps to equal the standard deviation of Manhattan heuristic's evaluations.

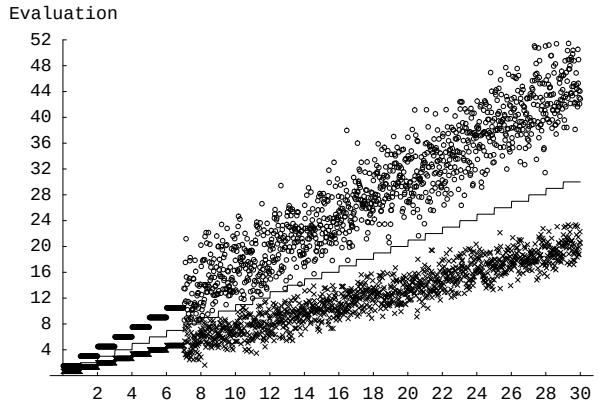

Figure 2. The evaluations given by our heuristic functions

To get the optimistic heuristic, we multiplied the obtained corrupted heuristic by a constant c:

$$h(n) = c \cdot (h^*(n) + e(n)) \,. \tag{8}$$

We chose $c = 2/3$ to emulate the level of errors Manhattan heuristic commits. The pessimistic heuristic of comparable relative error was obtained by multiplying the corrupted heuristic by inverted constant $1/c$, that is with $3/2$. The random process of corrupting the true evaluations was of course repeated for the pessimistic heuristic (we did not use the same errors as with the optimistic heuristic). If some evaluations in either heuristic were not pessimistic or optimistic as intended, their values were corrupted again.

Both heuristics are plotted in Figure 2. The x-axis gives the difficulty (true value) of the position, the y-axis gives the heuristic evaluation of the position. The crosses represent the optimistic heuristic, the circles the pessimistic heuristic, and the solid line represents the true evaluations. A random sample of 50 positions of each difficulty is displayed. The figure clearly shows that the average heuristic error grows close to linear with increasing difficulty of positions for both heuristics.

The first seven levels of difficulty deserve an explanation. For these levels we did not corrupt the true evaluations, we just multiplied them with the appropriate constant. The reason for this is that 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. Thus, we once again decided to model after the Manhattan distance heuristic, which also without exception gives correct estimates for the first seven levels of difficulty.

3.2 The search engine

We varied the depth of RTA* lookahead from 1 to 30, thirty being the difficulty of the hardest 8-puzzle problems. We were able to reach these very high depths of lookahead by employing transposition tables — yet another technique known from computer chess. We took advantage of the comparatively small number of possible positions in the 8-puzzle domain. We calculated depth 1 lookahead evaluations for all positions and stored these values in an array. Then we calculated depth 2 evaluations for all positions by doing one ply of search and using previously stored depth 1 evaluations, again storing the calculated evaluations. We repeated the process for other depths. A more detailed description of the procedure is given in [11].

3.3 Results

We were interested in two characteristics: the percentage of correct decisions each of our heuristics makes when used in conjunction with RTA* search and the actual solution length such a search produces.

3.3.1 Percentage of correct decisions

When measuring the percentage of correct decisions we varied the difficulty of the problems to be solved and the depth of lookahead. For a given difficulty level of problems and a given depth of lookahead, we measured the average percentage of correct decisions on *all* possible puzzles of this level where one path is clearly better from the others (otherwise there is nothing to decide between).

The results of the experiments are presented in Figure 3. The x-axis represents the depth of lookahead, and the y-axis represents the percentage of correct decisions made. Figure 3 shows a representative subset of the results. The chart on the left represents moderately difficult puzzles, the middle one hard puzzles, and the right chart represents a random mixture of 1,000 puzzles of various difficulties — this way of testing is quite common and was used for example in [3, 7]. The first two charts do not represent a single puzzle, but rather all puzzles of the given difficulty.

It is obvious from the charts that the pessimistic heuristic, represented by a dashed line, gives rise to a higher average percentage of correct decisions than the optimistic heuristic for all difficulty levels of the puzzles. This is especially so when the lookahead depth is close to the difficulty level of puzzles.

One may argue that perhaps the constant $c = 3/2$ used to get the pessimistic heuristic is misguided, and that $c = 4/3$ should be used instead. The latter gives the same level of errors in absolute terms, while the first one gives the same relative errors. We indeed constructed such a pessimistic heuristic as well, and the results were, as expected, even more in favour of the pessimistic heuristic.

3.3.2 Solution length

When measuring the solution length we varied the depth of lookahead and the quality of heuristics used (by varying the parameter σ for the pair of heuristic functions). The results are presented in Figure 4. The x-axis again represents the depth of lookahead, and the y-axis represents the solution length. The results are averaged over all legal puzzles. The dotted line at 21.50 represents the length of optimum solution averaged over all legal puzzles. The left chart represents the case for $\sigma = 2.5$ (the pair of heuristics modelled after the Manhattan heuristic), while the other two charts represent the cases with larger heuristic errors, $\sigma = 3$ and $\sigma = 4$, respectively.

We can see that pessimistic heuristics (dashed line) clearly outperform their optimistic counterparts (solid line) by consistently finding shorter solutions. It is interesting that after the depth of lookahead reaches 5 moves, the gain in solution length (thick line) is quite constant. At very high search depths the gain of course decreases, eventually reaching zero, because more and more puzzles are solved optimally since the solution is seen directly from the starting state.

The gain in solution length for the pessimistic heuristic over its optimistic counterpart, modelled by Manhattan heuristic, is about 5% to 10% (on the interesting interval with lookahead depth over 5 and before too many solutions are directly found). The decreasing quality

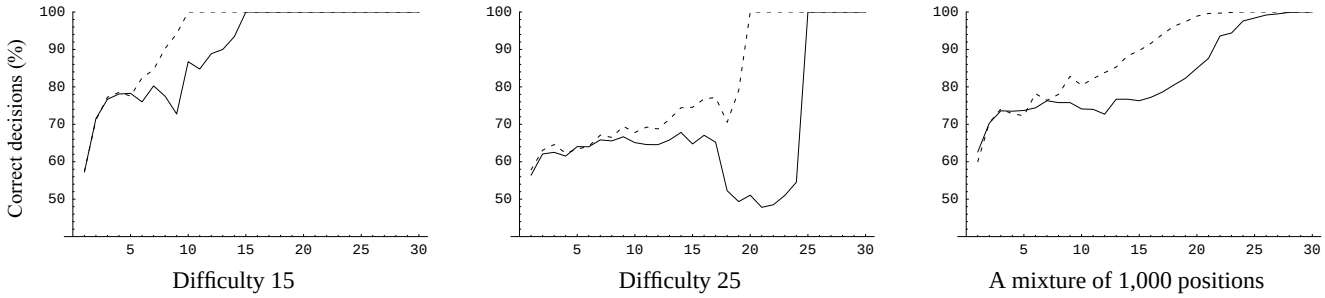

Figure 3. The comparison of correct decisions made between optimistic (solid line) and pessimistic (dashed line) heuristic

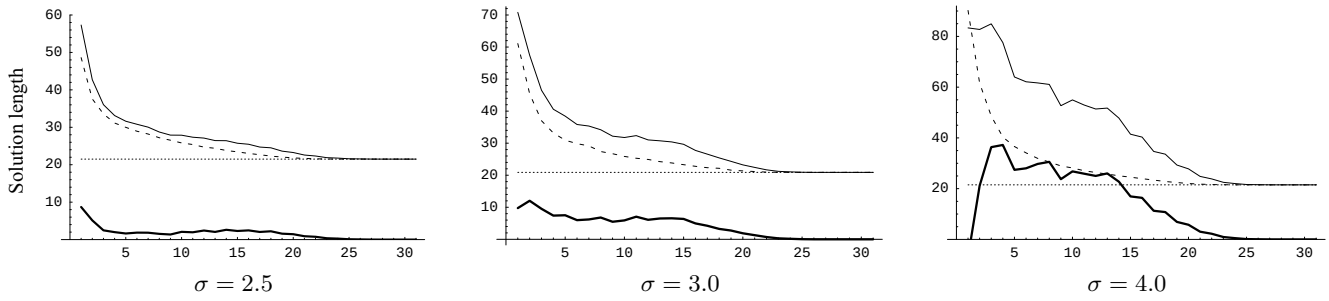

Figure 4. The comparison of solution length between optimistic (solid line) and pessimistic (dashed line) heuristic; the thick line represents the difference between them (the gain by pessimistic heuristic)

of heuristics, however, causes a sharp increase in the gain. For $\sigma = 3$ the gain is slightly below 20% and for $\sigma = 4$ it is already about 50%.

3.4 Search depth pathology

A search depth pathology is a phenomenon when deeper lookahead results in more decision errors committed or worse solutions found. This pathology was first discovered in two-player games in the late 1970s [1, 9]. An overview is given in [10, 11]. Recently, such pathology has also been detected in single-agent search [3, 5]. As we can see from the charts in Figures 3 and 4 our artificial heuristics also display such pathological behaviour. However, it is interesting that the pessimistic heuristic displays lesser inclination towards such behaviour than its optimistic counterpart. As we can see the pathological behaviour is not limited only to decision quality but also manifests itself in solution length. This was also observed by Korf [7]. With lower quality of heuristics the optimistic ones become more and more pathological while the pessimistic ones behave normally. This deserves further study.

4 DISCUSSION

We have shown that if the error committed by the heuristic evaluation increases with the difficulty of the problems, pessimistic heuristics are comparatively more successful than optimistic ones of equal quality, and vice versa, if the heuristic error decreases with the difficulty of the problems, then optimistic heuristics are better. This gives rise to an important question: which is more plausible, increasing or decreasing heuristic errors? We believe that the majority of real-life heuristics belong to the first group, namely that their heuristic errors increase with increasing difficulty of the problems. For example, imagine a task scheduling problem and a heuristic for it. If the optimal solution is that the tasks complete in, say, 10 hours, it is easy to imagine a heuristic estimating this time as somewhere between 8 and 12 hours, that is committing an error of ± 2 hours. However, if the optimal solution is that the tasks complete in 1,000 hours, it is quite unimaginable that the heuristic would estimate this time as somewhere between 998 and 1,002 hours. That would be an incredibly accurate heuristic. It is much more likely that the heuristic would estimate the time needed as something between 800 and 1,200 hours, committing the error of ± 200 hours. We believe it is quite conceivable that the error is usually *relative* to the size/difficulty of the problem; that the heuristic is, say, 20% off the mark. However, the equations in Section 2 only need this error to increase in absolute terms, it does not matter that the heuristic is more or less equally wrong in relative terms. From this we conclude that pessimistic heuristics seem to be preferable. We can give one more example: to which group do the two well-known heuristics for the 8-puzzle belong? Both, Manhattan distance and "Manhattan distance + Sequence score" heuristics, described e.g. in [2], belong to the first group — their error rises with increasing difficulty of the problems.

A possible argument in favour of optimistic heuristics could be that they may be easier to construct than pessimistic ones. But is this really so? Probably it is just that people are more used to opti-

mistic heuristics since they were usually preferred. For example, it is trivial to use air distance as a heuristic when approximating the road distance between two cities. But it is similarly trivial to use as a heuristic the distance based solely on highways, not on all the roads. This heuristic is pessimistic. Another point should be noted here. The equations in Section 2 do not say that it is strictly necessary for a heuristic to be pessimistic for *every* position, just for most of them. This, on the other hand, is not the case with the admissibility theorem — it necessitates that *all* positions are optimistically evaluated for the heuristic to be admissible.

The difference in the decision accuracy between the two heuristic functions in our experiments is a few percent, unless the search reaches the vicinity of goal nodes, where this difference increases. How much are these few percent worth? Not so little, because the search makes a series of decisions and for every single one of them the pessimistic heuristic gives it an extra few percent. For example, in the 8-puzzle domain, each mistake means the search will have to make at least two additional steps — one going back to the position where it came from, the other taking the right path (which it could have taken in the first place if it would not make a mistake).

Suppose we have an optimistic heuristic for some problem, e.g., the Manhattan distance for the 8-puzzle. We could make it pessimistic by adding a constant to it. Unfortunately this approach does not work, because by doing so the new heuristic's error would actually increase with decreasing problem difficulty, so all we would do is transform one disadvantageous case into another. How about multiplying an optimistic heuristic by a constant to get a pessimistic one?[2] This approach might actually work, though not for every heuristic. In some cases, like with Manhattan distance, we could again get a pessimistic heuristic whose error again increases with decreasing difficulty of the problems. But for some heuristics we could succeed. But even in this case we would have to experiment with finding the right constant, since if the constant is too big, we degrade the heuristic too much, and if the constant is too small, the resulting heuristic might not be pessimistic.

In section 2, we showed that RTA* using optimistic heuristics is likely to make more decision errors than RTA* with pessimistic heuristics. A related question is how the fixed-depth lookahead performed by RTA* affects the accuracy of heuristic evaluations. When RTA* decides which node to move to next, it considers the minimin backed-up heuristic values rather than the original, "static" heuristic values. The relevant question is: Are these backed-up heuristic values more accurate than the static heuristic estimates themselves? We will here indicate, without full mathematical proof, the intuition behind the following answer: Minimin backed-up heuristic values have on average smaller absolute error when the terminal nodes of lookahead are evaluated by pessimistic heuristics than when they are evaluated optimistically. Of course in all detail this depends on the distribution of true values and on error distribution. It is however easy to see the advantage of pessimistic heuristics in the cases when there are several lookahead terminal nodes all of which having true values close to the lowest true value. Minimin lookahead will return the minimum heuristic value encountered at the fringe of the lookahead. In the case of pessimistic heuristics, this minimum heuristic value will tend to be the *least distorted* among the true values close to optimal. This is because minimin will tend to return the least overestimated node close to optimal. On the contrary, in the case of optimistic heuristics, this minimum heuristic value will tend to be the *most distorted* among the values close to optimal, because minimin will tend to return the most underestimated node value close to optimal.

[2] Both, adding a constant or multiplying with a constant is used in the weighted LRTA* algorithm [4, 12]. However, while the purpose there is to allow *some* states to be evaluated pessimistically, here we would like *all* states to be evaluated pessimistically.

5 CONCLUSIONS AND FURTHER WORK

We have shown that pessimistic heuristic functions are more effective than their optimistic counterparts of equal quality when used with incomplete search methods under the condition that the heuristic's errors grow with increasing difficulty of the problems. We have argued that such a condition is often met in practice, and that therefore pessimistic heuristics should be preferred.

We have also mentioned that our preconditions do not strictly necessitate that the heuristic is pessimistic for every single node. We believe it would be worthwhile to study how the percentage of such violations of pessimistic evaluation affect the gain.

The experiments also indicated that while both heuristic functions display some pathological behaviour, the pessimistic heuristics seem less inclined to do so. Why is this so is an interesting topic for further work.

ACKNOWLEDGEMENTS

This work was partly funded by ARRS, Slovenian Research Agency.

REFERENCES

[1] Donald F. Beal, 'An analysis of minimax', in *Advances in Computer Chess 2*, ed., M. R. B. Clarke, pp. 103–109. Edinburgh University Press, Edinburgh, UK, (1980).
[2] Ivan Bratko, *Prolog Programming for Artificial Intelligence*, Addison-Wesley Publishing Company, 3rd edn., 2001.
[3] Vadim Bulitko, 'Lookahead pathologies and meta-level control in real-time heuristic search', in *Proceedings of the Fifteenth Euromicro Conference on Real-Time Systems*, pp. 13–16, Porto, Portugal, (July 2003).
[4] Vadim Bulitko and Greg Lee, 'Learning in real-time search: A unifying framework', *Journal of Artificial Intelligence Research*, **25**, 119–157, (2006).
[5] Vadim Bulitko, Lihong Li, Russ Greiner, and Ilya Levner, 'Lookahead pathologies for single agent search', in *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, eds., G. Gottlob and T. Walsh, pp. 1531–1533, Acapulco, Mexico, (August 2003).
[6] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael, 'A formal basis for the heuristic determination of minimum cost paths', *IEEE Transactions on Systems Science and Cybernetics*, **4**(2), 100–107, (1968).
[7] Richard E. Korf, 'Real-time heuristic search', *Artificial Intelligence*, **42**(2-3), 189–211, (1990).
[8] Richard E. Korf, 'Linear-space best-first search', *Artificial Intelligence*, **62**(1), 41–78, (1993).
[9] Dana S. Nau, *Quality of Decision Versus Depth of Search on Game Trees*, Ph.D. dissertation, Duke University, Durham, NC, 1979.
[10] Aleksander Sadikov, *Propagation of heuristic evaluation errors in game graphs*, Ph.D. dissertation, University of Ljubljana, Faculty of Computer and Information Science, Ljubljana, Slovenia, 2005.
[11] Aleksander Sadikov, Ivan Bratko, and Igor Kononenko, 'Bias and pathology in minimax search', *Theoretical Computer Science*, **349**(2), 268–281, (2005).
[12] Masashi Shimbo and Toru Ishida, 'Controlling the learning process of real-time heuristic search', *Artificial Intelligence*, **146**(1), 1–41, (2003).
[13] Ken Thompson, 'Retrograde analysis of certain endgames', *ICCA Journal*, **9**(3), 131–139, (1986).