Towards Understanding Conceptual Differences Between Minimaxing and Product Propagation

Helmut Horacek[1]

Abstract. Advantages and disadvantages of minimaxing versus product propagation back-up rules for game tree searching have been intensively discussed in the literature. So far, examinations have almost exclusively been carried out through experiments, demonstrating slight superiorities for one or the other back-up rule. In contrast to these purely quantitative investigations, we aim at elaborating differences in strength of these back-up rules by characterizing properties of critical situations in which these differences prove relevant. Evidence from the examinations carried out shows that minimaxing is better for a uniform error distribution under pathologically high and very low error rates, while high frequencies of critical cases favoring product propagation lead to a dominance of this back-up rule for realistic error distributions in depth 2 searches. The results provide insights for assessing degrees of competence of the two back-up rules, suggesting combined uses when facing move decisions.[1]

1 INTRODUCTION

Advantages and disadvantages of minimaxing versus product propagation back-up rules for game tree searching have been intensively discussed in the literature. So far, theoretical models have barely explained the dramatic success in some games, and the weak performance in others. Theoretical examinations have almost exclusively been carried out through experiments, demonstrating slight superiorities for one or the other back-up rule, depending on the particular game chosen and on the assumptions underlying simulations on abstract game trees.

In contrast to these purely quantitative investigations, we aim at elaborating differences in strength of these back-up rules by characterizing properties of critical situations in which these differences prove relevant. Evidence from the examinations carried out shows that minimaxing is better for a uniform error distribution under pathologically high and very low error rates, while high frequencies of critical cases favoring product propagation lead to a dominance of this back-up rule for realistic error distributions in depth 2 searches. The results provide insights for assessing degrees of competence of minimaxing and product propagation, suggesting combinations of the two back-up rules based on value distributions and board evaluator competences.

This paper is organized as follows. First, we provide some background information. Then we summarize previous work on comparing minimaxing with product propagation. Next, we describe and analyse the basic situation in backing-up values in a game tree, followed by evidence from a case study with a concrete game tree model. Finally, we discuss impacts on move decisions.

1 Universität des Saarlandes, FB 14 Informatik, Postfach 1150, D-66041 Saarbrücken, B.R.D., email: horacek@ags.uni-sb.de

2 BACKGROUND

In order to act adequately in interesting games, it is necessary to resort to *heuristic* estimates in terms of an *evaluation function* $f(n)$ that assesses each node that represents a position with some error. We assume that $f(n)$ assigns a value to a node n from the viewpoint of the moving side at n. This function is used when searching deeper and backing up the heuristic values towards the given nodes' parents. In *two-person games* with perfect information, the most successful[2] approach for backing up is *minimaxing* (see [14] for an analysis), which is defined as follows:

Definition 1 *A* minimax value *$MM_f(n)$ of a node n can be computed recursively as follows (in the negamax formulation):*

1. *If n is considered* terminal*: $MM_f(n) \leftarrow f(n)$*
2. *else: $MM_f(n) \leftarrow max_i(-MM_f(n_i))$ for all child nodes n_i of n.*

However, the dramatic benefits of using minimaxing with deeper searches in practice have not been explained theoretically. Even to the contrary, Nau [9] showed that for certain classes of game trees the decision quality is degraded by searching deeper and backing up heuristic values using minimaxing. He called such behavior *pathological*. Essentially the same findings were reported independently by Beal [2]. Several subsequent studies like [13] provided insights into minimax pathology. Therefore, different back-up rules have been proposed, such as *product propagation* [13] (this rule was already used earlier by Slagle and Bursky [18]). It requires that an evaluation function $f'(n)$ returns values between 0 and 1 that are estimates of the probability that the position represented by node n is a forced win.

Definition 2 *A* probability estimate *$PP_{f'}(n)$ of a node n can be computed recursively as follows (in the negamax formulation):*

1. *If n is considered* terminal*: $PP_{f'}(n) \leftarrow f'(n)$*
2. *else: $PP_{f'}(n) \leftarrow (1- \Pi_i PP_{f'}(n_i))$ for all child nodes n_i of n.*

Product propagation is theoretically sound for *independent* probabilities, but this is generally *not* the case in practice. Similarly, minimaxing is not theoretically justified, too [13] – while product propagation ignores commitments to be made, minimaxing ignores uncertainty. Different interpretations of what these back-up rules model in terms of playing against omnipotent or fallible opponents can be found in [5, 13].

2 The best known examples are the special chess machine Deep Blue, which defeated the highest-rated human chess player in a match under tournament conditions, and the checkers program Chinook, which is even the official man-machine world champion [15].

3 PREVIOUS WORK

Several approaches comparing performance differences among minimaxing and product propagation were carried out by Nau. He investigated product propagation as an alternative back-up rule [11] and found no pathology in so-called P-games, where minimaxing is pathological. In fact, the values of the real leaf nodes of P-games directly correspond to the values of the squares in the initial board configuration, which are randomly assigned one of two values independently of the values of the other squares. Under these conditions as given in P-games, Nau's experiments resulted in a higher probability of correct move decisions using product propagation compared to minimaxing. In so-called N-games (with incremental dependencies of true game-theoretic values), the results showed about the same probability of correct move decision for both back-up rules. In a P-game contest, a program based on product propagation scored marginally better than an otherwise identical program based on minimaxing.

In later work, Nau *et al.* [12] reported that product propagation scored better than minimaxing in a P-game contest for "critical" games. For each initial game board, one game was played with one player moving first and another with his opponent moving first. For some game boards, one player was able to win both games of the pair. These are called critical games. Further experiments showed that minimaxing was better than product propagation (for search depths 3 and 4) in an N-game contest.

Nau [10] also used so-called G-games (with dependencies of true game-theoretic values in graphs where sibling nodes have many children in common) for comparing these propagation rules, which indicated some influence of the evaluation function used. G-game contests revealed that product propagation performed better than minimaxing if some evaluation function was used and worse than minimaxing if another function was used that is more accurate for these games. Results by Chi and Nau [3] confirmed this relationship of the respective advantages of these rules to the strength of an evaluation function used: the stronger the evaluation function the better for minimaxing.

Additionally, Chi and Nau compared these back-up rules on several games, including a small variant of kalah. Most interestingly, in this real game a program based on product propagation performed better than its opponent based on minimaxing.

Since both programs searched to the *same* depth, however, these comparisons were unfair for minimaxing, which could have utilized well-known pruning procedures for searching much deeper with the same number of nodes generated (for a comparison of pruning procedures see [7]). Still, there was some indication that product propagation may be the better rule for backing up heuristic values. Finally, Nau *et al.* [12] as well as Baum [1] investigated combinations of minimaxing and product propagation. Their results suggested that the respective advantages could be utilized by a combination of these back-up rules.

Only recently, results about the effects of deeper searches using minimaxing were achieved through an investigation of more realistic game tree conditions when using multivalued evaluations functions [17]. While these results are more general than those from experiments using concrete game-playing programs, they have a close relation to, e.g., computer chess and checkers practice. In addition, this model has been used for a comparison of minimaxing and product propagation, demonstrating slight advantages in favor of product propagation for depth 2, while minimaxing proved better for deeper searches [5].

4 THE BASIC BACK-UP CONSTELLATION

We carried out an analysis of the differences between the back-up rules according to systematic combinations of evaluation errors. The basic situation is illustrated in Figure 1, for branching factor 2 and depth 2. Two conditions make such a case a *critical case*:

1. The back-up rules must select different moves.
2. One move leads to a won position and the other to a lost one.

Whether or not the first condition holds can be derived from the definitions of the competing back-up rules. We assume that Max is on move at the root position R, M the node preferred by minimaxing, P the one preferrred by product propagation, and that $f'(M_1) \geq f'(M_2)$ and $f'(P_1) \geq f'(P_2)$ hold. According to minimaxing, $f'(M_1)$ and $f'(M_2)$ must both be greater than $f'(P_2)$, the smaller value of P's successors. Conversely, product propagation demands that $f'(P_1)$ must be greater than both $f'(M_1)$ and $f'(M_2)$, otherwise $f'(P_1)*f'(P_2) > f'(M_1)*f'(M_2)$ would not hold. Consequently, there exists a partial ordering between the values of these four nodes:

$$f'(P_2) \leq f'(M_2) \leq f'(M_1) \leq f'(P_1)$$

The second property that makes a case critical depends on the true values associated with the competing nodes, M and P, which in turn depend on the true values of their successor positions. Based on this association, we introduce the notion of an evaluation error, adopted from [17]: if a heuristic value $f(n)$ is positive ($f'(n) > 0.5$), and the true value of the position is won for the side to move, then we say that f is *correct* for n, and it is not otherwise. Hence, we have to distinguish which of the positions M_1, M_2, P_1, and P_2 are evaluated correctly in the above sense and which ones are not, which yields 16 cases to be considered. A further distinction concerns the relation of these four values to the transition point between loss and win (the "draw value", 0 in terms of f, and 0.5 in terms of f'), which yields 5 cases (see Figure 2). Hence there is a total of 80 cases. Among these cases, those have to be extracted where the move decision matters, that is, where M is won and P lost for the side to move, or vice versa. For illustration purposes, let us discuss two representative cases:

1. If there is no evaluation error, then only the case where the draw value falls in interval I_4 matters, when M is won, and P lost. If the draw value falls in interval I_5, then both M and P are won. For the remaining intervals, both M and P are lost.
2. Similarly, consider the draw value lying in interval I_3, and exactly one of the heuristic values as erroneous. Then this value must be $f'(M_2)$ in order to make minimaxing superior, and it must be $f'(P_2)$ to favor product propagation. In other cases, the decision is irrelevant, since both M and P are lost.

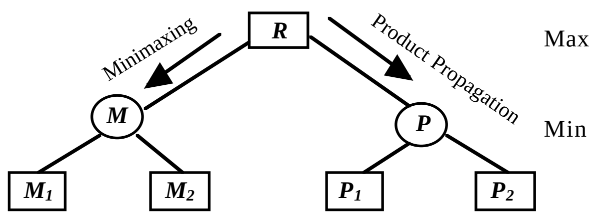

Figure 1. The basic decision situation

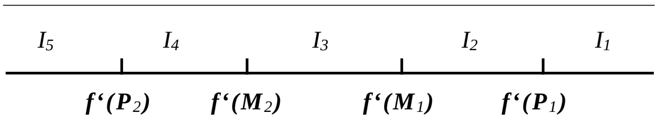

Figure 2. Intervals of the draw value

When pursuing the analysis of all 80 cases along similar lines, the differences between minimaxing and product propagation appear as listed in Table 1 (we have verified this analysis computationally for a limited set of value combinations). The left part contains the constellations favoring minimaxing, and the right one those favoring product propagation. The lines are ordered according to the number of evaluation errors associated with the four positions, where the erroneous positions and the interval in which the draw value falls are indicated for each case.

5 ANALYSIS OF THE BASIC CASE

Table 1 demonstrates that advantages and disadvantages of minimaxing and product propagation are balanced – 15 cases appear in each column. There is one extra case in favor of minimaxing with no evaluation error, and another case with four errors, compensated by two additional cases with two evaluation errors on the side of product propagation. Moreover, the cases in favor of minimaxing with one resp. three errors where the draw value falls in one of the intervals I_2 or I_4 have counterparts for product propagation with the other intervals, I_4 and I_2, respectively. Whereas the same number of intervals appear on both sides of Table 1, the errors of individual positions are distributed rather unevenly: M_1 and P_2 appear 6 times and M_2 and P_1 9 times on the side of minimaxing, while P_1 appears only 3 times, M_2 7 and M_2 8 times, and P_2 even 12 times on the side of product propagation.

In order to assess the quality of the competing back-up rules based on this analysis, we assume that an error function e is associated with f' that indicates for each value of f' how likely it is that the value is correct or not. Depending on assumptions about e, different results emerge.

For the special case that the probability of error is uniformly distributed, the large number of symmetries make the analysis simple and lead to an interesting result. Under this constellation,

Table 1. Evaluation error constellations in favor of either back-up rule

advantage of	*minimaxing*	*product propagation*
no error	I_4	
one error (position)	$I_4(P_1)$	$I_2(P_2)$
	$I_3(M_2)$	$I_3(P_2)$
	$I_5(P_1)$, $I_5(P_2)$	$I_5(M_1)$, $I_5(M_2)$
two errors (positions)	$I_1(M_1,M_2)$	$I_1(P_1,P_2)$
	$I_2(M_1,M_2)$	$I_2(M_1,P_2)$, $I_2(M_2,P_2)$
	$I_3(P_1,M_2)$	$I_3(M_1,P_2)$
	$I_4(P_1,P_2)$	$I_4(M_1,P_2)$, $I_4(M_2,P_2)$
	$I_5(P_1,P_2)$	$I_5(M_1,M_2)$
three errors (positions)	$I_2(M_1,M_2,P_1)$	$I_4(M_1,M_2,P_2)$
	$I_3(M_2,P_1,P_2)$	$I_3(M_1,M_2,P_2)$
	$I_1(M_1,M_2,P_2)$, $I_1(M_1,M_2,P_1)$	$I_1(M_2,P_1,P_2)$, $I_1(M_1,P_1,P_2)$
four errors	I_2	

all those cases facing one another in Table 1 (one favoring minimaxing, and the other product propagation) with the same number of errors and identical intervals for the draw value occur with equal frequency. Consequently, only the two extreme cases favoring minimaxing and the two average cases favoring product propagation, as well as the difference between the number of cases falling in either of the intervals I_2 and I_4 remain as decisive factors. For a board evaluator competence of exactly 50% error rate, also these remaining cases are leveled out completely, hence the performance of the back-up rules is the same, per average. For a constant error rate other than 50%, the fact that I_2 must be larger than I_4 – otherwise $f'(P_1)*f'(P_2) > f'(M_1)*f'(M_2)$ would not hold – proves to be a decisive factor. Since the draw value falls more frequently in I_2 than in I_4, "pathological" board evaluators with more than 50% error rate always favor minimaxing, while better board evaluators favor product propagation, at least for error rates that are not extremely low. In fact, there is some point in decreasing the constant error rate where the contribution of the cases with only correct evaluations overcompensates the contribution of the cases with errors from the larger interval I_2. For the variants of the game tree model used in the case study reported about in the next section, this score lies between 88% and 95% correct evaluations.

For more realistic error functions, where the degree of error continuously decreases with increasing probabilities to win, we cannot provide clear-cut qualitative results as for uniform error distributions. Nevertheless, we can derive some general tendencies for which the case study in the next section provides a good deal of evidence. When building corresponding pairs or double pairs out of the entries in Table 1, half of them favoring minimaxing and the other half favoring product propagation, it turns out that most of the resulting compensative effects favor product propagation, provided the degree of error associated with the positions mirrors the actual probabilities to win. For example, there are four cases where the draw value falls in I_5 with exactly one position in error – those where the succesors of P are erroneously evaluated favor minimaxing, and the others product propagation. Since $f'(P_1)*f'(P_2) > f'(M_1)*f'(M_2)$ holds, the difference between $f'(P_1)$ and $f'(M_1)$ is larger than that between $f'(P_2)$ and $f'(M_2)$, which means that minimaxing, the side where P_1 is in error, is worse off. Consider, as another example, the cases in favor of minimaxing with the draw value falling in I_4 and I_2, and only P_1 resp. all positions other than P_1 are erroneously evaluated. In the corresponding constellation for product propagation I_2 and I_4 are interchanged, with P_2 instead of P_1 being the position evaluated differently from the others. As before, minimaxing is the side where P_1 is evaluated incorrectly, and this effect is even augmented because I_2 is larger than I_4 since $f'(P_1)$ must be extremer than $f'(P_2)$. Similar dominances hold between most pairs built in this way, except to the case with no error.

We conjecture that several effects reduce or even overcompensate this dominance of product propagation. For example, the estimated degrees of error may deviate significantly from the actual probabilities to win. More importantly, there are several effects of deeper searches which tend to favor minimaxing (see also [5, 12]): the propagated values tend to become extremer, increasing the frequency of cases with the draw value in I_4, the propagated errors for the case with no error tend to increase as well, and the constellations with exactly two position evaluated incorrectly place most combinations where sibling nodes are both evaluated correctly or not on the side favoring minimaxing.

6 A CASE STUDY

In order to get a sense for the differences between minimaxing and product-propagation in quantitative terms, we have computed the outcome for a small version of the game tree model introduced by Scheucher and Kaindl [16, 17], the most recent and probably most realistic model proposed in the literature so far. It is a multi-valued model, with a specific sort of dependencies among related game tree nodes. For the reader's convenience, the detailed assumptions are repeated here in the appendix.

In making use of this game tree model, we have varied the value of the root node and the maximal value increment per move, a, in order to examine dependencies of values distributions. Moreover, we have tried several functions expressing probabilities to win and combinations with several distributions of the probability of error. For probabilities to win (used for mapping evaluator scores f onto probabilities to win f') we have used variants for w_c (see the appendix) with c = 1, 4, and 16. For the error function we have used functions of the same shape, with coefficient choices overestimating or underestimating the probability to win (function e_2), as well as a proportional error rate function (function e_1), which starts by 0% at the extreme values h_{min} and h_{max}, going down to 50% at the draw value.

Altogether, the results prove to be quite robust. Almost all parameter combinations yield expectations clearly in favor of product propagation, which is consistent with the simulation results obtained by Kaindl and Scheucher. There was no influence of the value increment, and changes of the value distributions due to varying the root node value were marginal. Only for gross underestimations of the probabilities to win, minimaxing was competitive. Table 2 demonstrates two representative cases, each for a total of 1,000,000 positions, with root value 1 and maximal value increment 9, including repetitions of constellations due to symmetries in adding or substracting increments. The table contains two variants of error functions, the proportional one (e_1), and a function of the form w_c with c = 4 (probabilities to win are distributed according to w_c with c = 1, labeled e_2). Numbers in the table cells express percentages of the critical cases. When using the error function e_1, there are slightly more critical cases than when using e_2 (86,614 out of 1,000,000 total cases for e_1, as opposed to 81,986 for e_2).

Table 2. Contributions of evaluation error constellations in favor of either back-up rule (in percentage of critical cases)

advantage of		*minimaxing*		*product propagation*		
error function		e_1	e_2		e_1	e_2
no error	I_4:	0.39%	0.24%			
one error (position)	$I_4(P_1)$:	0.26%	0.16%	$I_2(P_2)$:	2.62%	2.74%
	$I_3(M_2)$:	1.73%	1.65%	$I_3(P_2)$:	2.06%	1.62%
	$I_5(P_1)$:	1.64%	1.65%	$I_5(M_1)$:	1.97%	1.99%
	$I_5(P_2)$:	2.24%	2.25%	$I_5(M_2)$:	2.17%	2.18%
two errors (positions)	$I_1(M_1,M_2)$:	0.51%	0.54%	$I_1(P_1,P_2)$:	0.58%	0.61%
	$I_2(M_1,M_2)$:	2.38%	2.49%	$I_2(M_1,P_2)$:	2.31%	2.41%
				$I_2(M_2,P_2)$:	2.05%	2.15%
	$I_3(P_1,M_2)$:	1.27%	1.22%	$I_3(M_1,P_2)$:	1.50%	1.44%
	$I_4(P_1,P_2)$:	0.24%	0.15%	$I_4(M_1,P_2)$:	0.31%	0.20%
				$I_4(M_2,P_2)$:	0.33%	0.21%
	$I_5(P_1,P_2)$:	1.44%	1.45%	$I_5(M_1,M_2)$:	1.69%	1.71%
three errors (positions)	$I_2(M_1,M_2,P_1)$:	2.03%	2.12%	$I_4(M_1,M_2,P_2)$:	0.29%	0.19%
	$I_3(M_2,P_1,P_2)$:	1.10%	1.05%	$I_3(M_1,M_2,P_2)$:	1.32%	1.27%
	$I_1(M_1,M_2,P_2)$:	0.35%	0.37%	$I_1(M_2,P_1,P_2)$:	0.40%	0.43%
	$I_1(M_1,M_2,P_1)$:	0.45%	0.48%	$I_1(M_1,P_1,P_2)$:	0.45%	0.47%
four errors	I_2:	1.54%	1.62%			

Table 3. Distribution of critical cases and errors over intervals

intervals	I_1	I_2	I_3	I_4	I_5
		number of critical cases			
error function e_1	*47,280*	*196,956*	*117,216*	*23,208*	*135,024*
error function e_2	*7,904*	*32,522*	*17,738*	*2,342*	*21,480*
		percentage of critical cases favoring product propagation			
error function e_1	*0.13%*	*1.02%*	*0.41%*	*0.05%*	*0.51%*
error function e_2	*0.13%*	*1.07%*	*0.59%*	*0.05%*	*0.53%*

The entries in Table 2 provide insights about the frequency of individual cases out of all critical constellations, which includes the low frequency of the case with no errors. In Table 3, this information is aggregated over the five intervals of the draw value, in absolute numbers and in percentages of critical cases. In particular, this Table shows that the frequency in which the draw value falls in interval I_2 strongly dominates the cases where it falls in I_4. In fact, this is the main reason for product propagation scoring better than minimaxing, at least for depth 2 searches. Hence, the results suggest that the contribution of the error rate is clearly overcompensated by the frequency in which constellations occur that favor one back-up rule over the other.

Finally, aggegating the contributions over errors of individual positions also yields an interesting result (see Table 4). If the "major" argument in favor of either back-up rule ($f'(M_1)$ for minimaxing, and $f'(P_1)$ for product propagation) proves to be wrong, then the contribution to favoring the other back-up rule goes up considerably. For their sibling positions, which in some sense are "secondary" arguments favoring the other back-up rule, correspondingly, the inverse result is obtained.

7 IMPACTS ON MOVE DECISIONS

For usage in practical games, minimaxing is widely preferred to product propagation because of its ability to search much deeper with comparable resources due to the possibility of cut-offs [8]: in computer chess, for example, a product propagation search to depth n would require resources comparable to a minimaxing search to depth n^2, per average. Thus, in order for product propagation to be competitive, the depth of search must be of minor importance, at least in the specific position under consideration. In such a situation, there are several criteria for choosing among the competing back-up rules in a motivated manner:

1. If the minimaxing values of the competing positions are identical, then the additional information available to product propagation in terms of "second best" moves is decisive.

Table 4. Percentage of critical cases with a specific position misevaluated (positive scores favor minimaxing, negative ones product propagation)

positions	M_1	M_2	P_1	P_2
error function e_1	-2.57%	3.11%	8.54%	-6.95%
error function e_2	-2.06%	3.41%	8.40%	-6.85%

2. A further criterion lies in the interval in which the draw value falls. If it is I_4 (or, eventually, I_1), a preference in favor of minimaxing may be assumed, since the advantage of product propagation in this interval is only marginal, even for accurate estimates of the probability to win, and it disappears for less accurate ones, as additional data shows. For the other intervals, preference should be given to product propagation.
3. For board evaluators that express their assessments not only by point values, there exists a further option (e. g., the chess program Merlin has a facility to recognize some kinds of unclear positions [4]). Since Figure 4 shows that the competence of one back-up rule is worse than that of the other if the correctness of major or secondary arguments in its favor (M_1 and P_2 for minimaxing) is in doubt, available uncertainty estimates justify the choice in favor of one of the back-up rules.

8 CONCLUSION

In this paper, we have aimed at elaborating differences in strength of back-up rules by characterizing properties of critical situations in which the differences between these back-up rules prove relevant. Evidence from the examinations carried out demonstrates that minimaxing is superior to product propagation for a uniform error distribution under pathologically high and very low error rates, while high frequencies of critical cases favoring product propagation lead to a dominance of this back-up rule for realistic error distributions and depth 2 searches.

In the future, we intend to elaborate more precisely the conditions under which product propagation is superior to minimaxing for depth 2 searches. Moreover, we want to investigate effects of deeper searches on value distributions and propagated error probabilities, and to analyse typical sorts of situations in a game, such as forced moves. Our results provide some insights for assessing degrees of competence of minimaxing and product propagation. Moreover, motivated combinations of the two back-up rules could be built, at least for specific situations, according to relevant properties observed in a particular game.

ACKNOWLEDGEMENT

I would like to thank Hermann Kaindl and Anton Scheucher for introducing me into details about the game-tree model used in their work and for discussions about ideas underlying this paper.

REFERENCES

[1] E. Baum: 'On Optimal Game Tree. Propagation for Imperfect Players', in Proc. of the *Tenth National Conference on Artificial Intelligence*, 507-512, San Jose, CA, (1992).

[2] D. Beal: 'An Analysis of Minimax', in *Advances in Computer Chess 2*, M. Clarke (ed.), 103-109, Edinburgh University Press, (1980).

[3] P. Chi, D. Nau: 'Comparison of Minimax and Product Back-up Rules in a Variety of Games', in *Search in Artificial Intelligence*, L. Kanal and V. Kumar (eds.), 450-471, Springer, New York, (1988).

[4] H. Horacek: 'Reasoning with Uncertainty in Computer Chess', *Artificial Intelligence* **43**, 37-56, (1990).

[5] H. Horacek, H. Kaindl, M. Wagner: 'Probabilities in Game-Playing: Possible Meanings and Applications', in Proc. of the *Third Austrian Meeting on Artificial Intelligence*, 12-23, Springer, Berlin, (1987).

[6] H. Kaindl, A. Scheucher: 'Back-up of Heuristic Values: Minimaxing vs. Product Propagation', In Proc. of *ECAI-98*, 665-669, (1998).

[7] H. Kaindl, R. Shams, H. Horacek: 'Minimax Search Algorithms with and without Aspiration Windows', *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* **13**, 1225-1235, (1991).

[8] D. Knuth, R. Moore: 'An Analysis of Alpha-Beta Pruning', *Artificial Intelligence* **6**, 293-326, (1975).

[9] D. Nau: 'Pathology on Game Trees: a Summary of Results', in Proc. of the *First National Conference on Artificial Intelligence*, 102-104, Stanford, CA, (1980).

[10] D. Nau: 'On Game Graph Structure and its Influence on Pathology', *International Journal Computer and Information Sciences* **12**, 367-383, (1983).

[11] D. Nau: 'Pathology on Game Trees Revisited, and an Alternative to Minimaxing', *Artificial Intelligence* **21**, 221-244, (1983).

[12] D. Nau, P. Purdom, H. Tzeng: 'Experiments on Alternatives to Minimaxing', *International Journal of Parallel Programming* **15**, 163-183, (1986).

[13] J. Pearl: 'On the Nature of Pathology in Game Searching', *Artificial Intelligence* **20**, 427-453, (1983).

[14] J. Pearl: *Heuristics: Intelligent Search Strategies for Computer Problem Solving*, Addison-Wesley, Reading, MA, (1984).

[15] J. Schaeffer, R. Lake, P. Lu, M. Bryant: 'CHINOOK: The World Man-Machine Checkers Champion', *AI Magazine* **17**, 21-29, (1996).

[16] A. Scheucher: 'Untersuchungen von Minimaxing in einem mehrwertigen Modell', Diplomarbeit, Wirtschaftsuniversität Wien, (1992).

[17] A. Scheucher, H. Kaindl: 'Benefits of Using Multivalued Functions for Minimaxing', *Artificial Intelligence* **99**, 187-208, (1998).

[18] J. Slagle, P. Bursky: 'Experiments with a Multipurpose, Theorem-Proving Heuristic Program', *Journal of the ACM* **15**, 85-99, (1968).

APPENDIX: Game Tree Model

The assumptions of the underlying game-tree model are derived from Pearl's [13] basic *pathological* model (as given in [17]):

1. The tree structure has a uniform branching degree b.
2. True values of nodes (TV) are either *WIN* or *LOSS*.
3. True values have the game-theoretic relationship of two-person zero-sum games with perfect information. A nonterminal node is won if at least one of its child nodes is won.
4. Heuristic values h (assigned to a node n by a static evaluation function $f(n)$) belong to the set $\{-h_{min}, \ldots -1, +1, \ldots +h_{max}\}$.

Two additional conditions are taken from [17]:

1. Non-uniformity of error distribution

Whenever a heuristic value incorrectly estimates the true value, an error occurs. The non-uniform error distribution in this model

$$e_c(h) = \begin{cases} w_c(h) & h < 0 \\ 1 - w_c(h) & h > 0 \end{cases}$$

is based on the “probability to win” (for some c and $h = f(n)$):

$$w_c(h) = \begin{cases} \frac{1}{2} + \frac{1}{2\arctan(c)} \arctan(c\frac{h}{h_{max}}) & \text{if } h = f(n) \in [-h_{max}, h_{max}] \\ 0 & \text{otherwise} \end{cases}$$

2. Dependency of heuristic values

The heuristic values of child nodes depend on the heuristic value of the parent node, with a maximum change of a between node n and its child nodes n_i, depending on the side to move as follows:

MAX: $f(n) \le f(n_i) \le f(n) + a$ $\qquad$ MIN: $f(n) - a \le f(n_i) \le f(n)$