

# Why Minimax Works: An Alternative Explanation

**Mitja Luštrek and Matjaž Gams**

Jožef Stefan Institute  
Department of Intelligent Systems  
Jamova 39, 1000 Ljubljana, Slovenia  
mitja.lustrek@ijs.si, matjaz.gams@ijs.si

**Ivan Bratko**

University of Ljubljana  
Faculty of Computer and Information Science  
Tržaška 25, 1000 Ljubljana, Slovenia  
bratko@fri.uni-lj.si

## Abstract

In game-playing programs relying on the minimax principle, deeper searches generally produce better evaluations. Theoretical analyses, however, suggest that in many cases minimaxing amplifies the noise introduced by the heuristic function used to evaluate the leaves of the game tree, leading to what is known as pathological behavior, where deeper searches produce worse evaluations. In most of the previous research, positions were evaluated as losses or wins. Dependence between the values of positions close to each other was identified as the property of realistic game trees that eliminates the pathology and explains why minimax is successful in practice. In this paper we present an alternative explanation that does not rely on value dependence. We show that if real numbers are used for position values, position values tend to be further apart at lower levels of the game tree, which leads to a larger proportion of more extreme positions, where error is less probable. Decreased probability of error in searches to greater depths is sufficient to eliminate the pathology and no additional properties of game trees are required.

## 1 Introduction

Most game-playing programs are based on the minimax principle. Such programs choose the best move by searching the game tree to a chosen depth, heuristically evaluating the leaves and then propagating their values to the root using the minimax principle. It is generally agreed that deeper searches produce better evaluations at the root. However, first attempts to explain this mathematically yielded the paradoxical result that minimaxing amplifies the error of the heuristic evaluations and that consequently deeper searches produce worse evaluations [Nau, 1979; Beal, 1980]. This phenomenon is called the minimax pathology [Nau, 1979].

It was evident that the setting of these mathematical analyses omitted some property of real games that eliminates the pathology. Several explanations were proposed, but eventually most researchers came to the conclusion that the property they were looking for is the similarity of positions close to each other [Bratko and

Gams, 1982; Beal, 1982; Nau, 1982; Scheucher and Kaindl, 1998; Luštrek, 2004].

In most research on the minimax pathology, true position values were losses or wins. This seems reasonable, since in games like chess, positions can indeed only be lost or won (or drawn). In practice, however, a game playing program needs an evaluation function that makes it possible to maintain a *direction* of play, gradually moving toward a win, not just maintaining a won position without achieving the final goal. This requires a multivalued or even real-valued evaluation function.

We introduce a minimax model using real-valued evaluation function and a way to interpret real values as losses and wins. This leads to a new, simpler explanation of the pathology. Namely, node values at lower levels of the game tree are more dispersed. If the error of the evaluation function, represented by normally distributed noise, is independent of the depth of search, it turns out that the error in terms of loss/win evaluations decreases with the depth of search, because larger dispersion means a larger proportion of more extreme positions where error is less likely. This is different from what the previous analyses assumed and is sufficient to eliminate the pathology.

The paper is organized as follows. Section 2 presents the minimax pathology and gives an overview of the attempts to explain it. Section 3 introduces a minimax model based on real-number position values. Section 4 shows why the model from section 3 is not pathological, while seemingly similar models used in previous research were. Section 5 explains whether minimax in general can be expected to behave non-pathologically. Section 6 concludes the paper and points out where further research is needed.

## 2 The Pathology and Related Work

The minimax pathology was discovered independently by Nau [1979] and Beal [1980]. Beal's basic minimax model made several assumptions:

1. game tree has a uniform branching factor;
2. nodes of the tree can have two values: loss and win;
3. node values are distributed so that at each level of the tree the proportion of losses for the side to move is the same;
4. node values within each level of the tree are independent of each other;

5. the error of heuristic evaluation at a node at the lowest level of search, being the probability of mistaking a loss for a win or vice versa, is independent of the depth of search and the true value of the node.

We proceed to present Beal's basic model, although our analysis is mostly based on later work. Negamax representation is used in this section, i.e. node values are viewed as lost or won from the perspective of the side to move. Let  $b$  be the branching factor of the tree,  $d$  the depth of search and  $k_i$  the probability of a node at  $i$ -th level being lost. Levels are numbered downwards: from 0 for root to  $d$  for the lowest level of search.

A node can only be lost if all of its descendants are won (for the opponent), so the relation between the values of  $k$  at consecutive levels is governed by equation (1).

$$k_i = (1 - k_{i+1})^b \quad (1)$$

Assumption 3 requires  $k_i = k_{i+1}$ , which results in  $k_i = c_b$  for all  $i$ ; for example,  $c_2 = 0.3820$ .

Two types of evaluation error are possible: a loss can be mistaken for a win (false win) or a win for a loss (false loss). Let  $p_i$  and  $q_i$  be the probabilities of the respective types of error at  $i$ -th level. They are calculated according to equations (2) and (3).

$$p_i = \frac{1}{k_i} (1 - k_{i+1})^b (1 - (1 - q_{i+1})^b) = 1 - (1 - q_{i+1})^b \quad (2)$$

$$q_i = \frac{1}{1 - k_i} \sum_{j=1}^b \binom{b}{j} k_{i+1}^j (1 - k_{i+1})^{b-j} p_{i+1}^j (1 - q_{i+1})^{b-j} \quad (3)$$

It turns out that if the same  $p_d = q_d$  are used in searches to all depths, the error in the root, defined as  $p_0 k_0 + q_0 (1 - k_0)$ , increases with  $d$ . "This result is disappointing," concluded Beal, since it was exactly the opposite of what he set out to show. His model must have been flawed in some way.

In the years following the discovery of the pathology, several researchers attempted to find the flaw in Beal's basic model by attacking its assumptions (1. through 5. at the beginning of this section).

1. Michon [1983] observed that the pathology depends on the probability distribution of branching factor: game trees with uniform branching factor tend to be pathological, while game trees with, for example, geometrically distributed branching factor do not. It is not known whether real games have any of the non-pathological distributions, though.
2. Bratko and Gams [1982] were the first to experiment with multivalued evaluations. They assigned no particular meaning to different values, which resulted in behavior similar to the one with two values, i.e. pathological. Pearl's results were similar [1983]. Scheucher and Kaindl [1998] and Luštrek [2004] also used multiple and even real values, but only to establish realistic relations among node values and the magnitude of error, so even though their models were not pathological, they did not attribute the absence of pathology to real values. Sadikov *et al.* [2003] used multiple values in their analysis of king and rook versus king chess endgame. They explained the pathology, but their explanation involves multiple values

only indirectly and it is not known whether it applies to cases other than the endgame they studied.

3. Having node values distributed so that  $k_i = c_b$  for all  $i$  not only simplifies calculations, it is generally agreed to be necessary [Bratko and Gams, 1982; Beal, 1982; Nau, 1982]. If  $k_0 \neq c_b$  is chosen,  $k_i$  starts to oscillate between values close to 0 and 1 for relatively small  $i$ , meaning that we are dealing with games that are almost certainly won for one side and as such not interesting.
4. Most researchers [Bratko and Gams, 1982; Beal, 1982; Nau, 1982; Scheucher and Kaindl, 1998; Luštrek, 2004] agreed that similarity of positions close to each other eliminates the pathology, although they arrived at this conclusion in different ways. Pearl [1983] claimed that early terminations are the culprit, but these can also be interpreted as a form of node-value dependence.
5. Pearl [1983] showed that in order to overcome the pathology, the error of the evaluation function must decrease exponentially with the depth of search. It is generally believed that the quality of the evaluation cannot vary enough to account for the absence of pathology. Scheucher and Kaindl [1998] did use depth-dependent error. And although such error made increased depth of search more beneficial, node-value dependence was required to altogether eliminate the pathology.

The conclusion one can make based on the existing literature on the minimax pathology is that the pathology is usually not observed in real games because their position values are not independent of each other. This conclusion is reinforced by the fact that multiple authors have arrived at it in different ways.

### 3 A Minimax Model Based on Real Values

Even though all the explanations for the absence of pathology in minimax provided in the previous section are valid, at least under the assumptions their authors made, are these assumptions really necessary and realistic? We argue that real numbers should be used for position values, in which case another, more basic explanation is sufficient.

Both game-playing programs and humans use multivalued position evaluations. There is little doubt this is necessary in games where the final outcome is multivalued (Othello, tarok etc.). In games where the outcome can only be a loss, a win and perhaps a draw (chess, checkers etc.), multiple values might seem to be useful only as a way to express the uncertainty of a program or human. However, even given unlimited resources to determine the value of a position, in a losing position, the best one can do against a fallible and not fully known opponent is evaluate the position in terms of the probability of loss. In a winning position, even a perfect two-valued evaluation function could maintain a won position indefinitely without actually winning (or until termination due to 50-move rule in chess). In essence, multivalued evaluation function is necessary to differentiate between multiple winning (or losing, if only such are available) moves. Scheucher and Kaindl [1998] demonstrated on chess that a two-valued evaluation function performs poorly compared to a multivalued one.

We propose a minimax model similar to Beal’s basic model, except that it uses real numbers for position values:

1. game tree has a uniform branching factor;
2. nodes of the tree have real values;
3. if the real node values are converted to losses and wins, they are distributed so that at each level of the tree the proportion of losses for the side to move is the same;
4. node values within each level of the tree are independent of each other;
5. the error of heuristic evaluation at a node at the lowest level of search, being normally distributed noise, is independent of the depth of search and the true value of the node.

Game trees built according to our model are assigned independent uniformly distributed values from  $[0, 1]$  interval to the leaves at level  $d_{\max}$ . These are true values; true values of internal nodes are obtained by backing up the true leaf values using the minimax rule. When searching to depth  $d$ , heuristic values at level  $d$  are generated by corrupting the true values with normally distributed noise representing the error of the heuristic evaluation function; heuristic values of nodes at levels  $< d$  are obtained by backing up the corrupted values at level  $d$  using the minimax rule.

Two types of error can be observed at the root of a game tree: *position error*, which is the absolute difference between the true and the heuristic value of the root, and *move error*, which is the probability of choosing a wrong move because of position error at the root’s descendants. However, neither type of error corresponds directly to the error in two-value models used in most of the previous research: *two-value error* is defined as the probability of mistaking a loss at the root for a win or vice versa.

In order to measure two-value error, real values must be converted to losses and wins. This can be accomplished by establishing a threshold  $t$ : the values below it are considered losses and the values above it wins. According to Beal’s assumption 3, if negamax representation is used,  $k_i = c_b$  for all  $i$ . We do not use negamax representation in our model, so  $k_i$  alternates between  $c_b$  and  $1 - c_b$ . Since true values of the leaves are distributed uniformly in  $[0, 1]$  interval,  $k_d = c_b$  is achieved by setting  $t = c_b$ . Even though real-value minimaxing is used,  $k_i$  behaves as desired for  $i > 0$ . This happens for two reasons. First, leaf values in our real-value model, converted to two values, correspond exactly to leaf values in Beal’s basic model. The probability of a loss at a leaf with value  $X$  is  $P(X < t)$ , which for uniform distribution in  $[0, 1]$  interval and  $t = c_b$  equals  $c_b$ . In Beal’s basic model, the probability of a loss at each leaf is  $k_d$ , which also equals  $c_b$ . Second, real- and two-value minimaxing are equivalent in the sense that performing minimax on losses and wins from level  $i$  to  $j < i$  gives the same results at level  $j$  as performing minimax on the underlying real values from level  $i$  to  $j$  and converting them to losses and wins at level  $j$ . This is illustrated in Figure 1; losses are marked with “-” and wins with “+”.

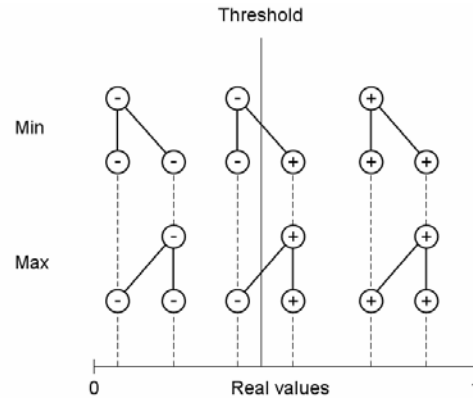


Figure 1: Equivalence of real- and two-value minimaxing.

We conducted Monte Carlo experiments with game trees generated according to our model. Only the results for  $b = 2$  and  $d_{\max} = 10$  are presented in this paper; the results for larger branching factors and depths are similar. The results are averaged over 10,000 game trees. For each tree, there were 10 repetitions with randomly generated noisy values for each  $d$ . Figure 1 shows position, move and two-value error at the root of the game tree with respect to the depth of search; standard deviation of noise is 0.1.

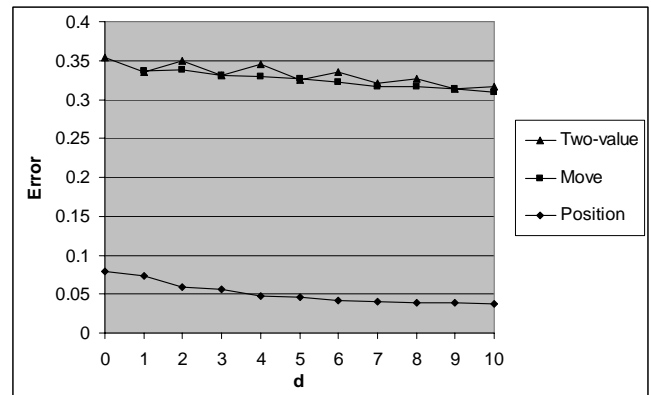


Figure 2: Error at the root with respect to the depth of search.

As can be seen in Figure 2, all three types of error decrease with the depth of search (with the exception of even/odd level fluctuations of two-value error). Note that these observed behaviors are different from Beal’s original results [1980], which were pathological.

Even though our primary concern is comparison with Beal’s basic model, we checked whether the absence of pathology occurs only under the described settings or is it more general. We tried uniform distribution of error and normal distribution of node values as well as different forms of dependence among node values. None of the experiments yielded pathology except for some rare cases where slightly pathological behavior was caused by very large static error. There was no pathology in terms of position error, but move and two-value error did in some cases behave pathologically when their static values were close to 0.5. However, since 0.5 is the point where evaluations become completely random, this seems to be of little practical importance.

## 4 Why is Our Model Not Pathological?

Considering that our model is very similar to Beal’s, why is it not pathological? To answer this question, we must examine two-value error at the lowest level of search. Beal’s assumption 5 states that it should be constant with search depth, but in our model, real-value position error is set to be constant instead (which is achieved by using normally distributed noise with the same standard deviation at all levels). Two-value error at the lowest level of search is shown in Figure 3; standard deviation of noise is 0.1.

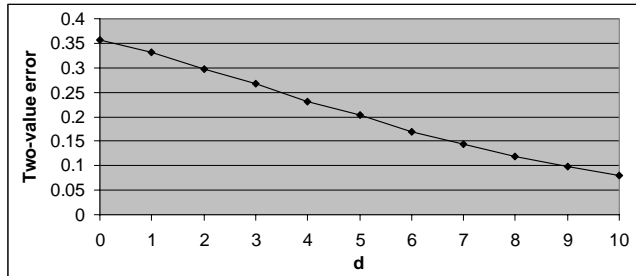


Figure 3: Two-value error at the lowest level of search with respect to the depth of search.

As can be seen in Figure 3, two-value error at the lowest level of search decreases with the depth of search. This is different from Beal’s assumption 5. However, to eliminate the pathology, Pearl [1983] observed that if the error is small, it should decrease by a factor of 1.528 every two levels, i.e. exponentially with the depth of search, while in Figure 3, it decreases roughly linearly. For Pearl’s observation to be true, the error should be quite small, though. For example, if the error at depth 10 is 0.1 (the value chosen by Bratko and Gams [1982] and close to the values we experimented with), Pearl’s approximation gives two-value error 0.8326 at depth 0, while the exact error is 0.3932. We chose not to work with smaller errors in this paper because move and two-value error are probabilities computed from frequencies and when they are very small, very large samples are required to obtain meaningful results.

If noise introduced at the lowest level of search is adjusted so that two-value error at the lowest level of search is always 0.1, position error at the lowest level of search increases with the depth of search as shown in Figure 4.

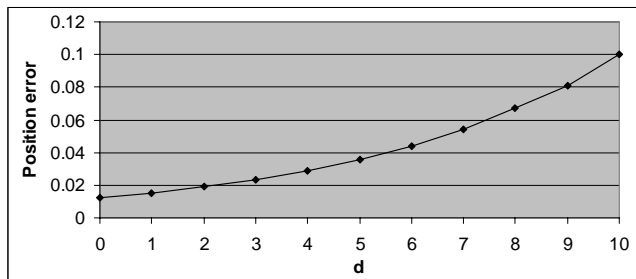


Figure 4: Position error at the lowest level of search with respect to the depth of search when two-value error at the lowest level of search is always 0.1.

Figure 5 shows two-value error at the root when two-value error at the lowest level is always 0.1; results for Beal’s basic model are shown for comparison.

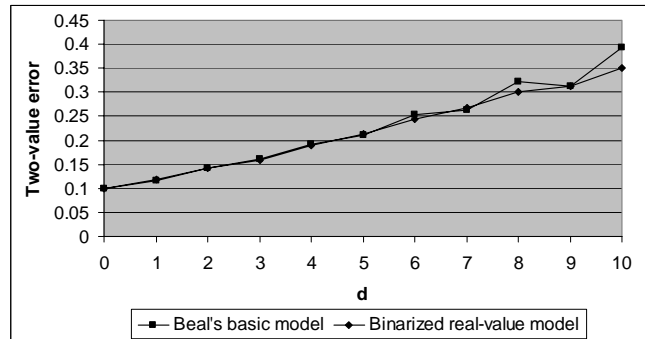


Figure 5: Two-value error at the root with respect to the depth of search when two-value error at the lowest level of search is always 0.1.

As can be seen in Figure 5, the results for binarized real-value model and the results for Beal’s basic model match quite well. The matching is not perfect because in the real-value model, the probability of a false win at the lowest level of search is higher than the probability of a false loss. This happens because false wins occur in  $[0, t)$  interval, while false losses occur in  $(t, 0]$  interval. Since  $t = c_2$ , the former is smaller, therefore node values are on average closer to the threshold, and hence two-value error is more likely. The ratio of probability of a false win : probability of a false loss is  $(1 - t) : t$ . If the overall probability of two-value error is to remain 0.1, the appropriate settings in Beal’s model are  $p_d = 0.05 / t$  and  $q_d = 0.05 / (1 - t)$ . Under these settings, the models match perfectly.

## 5 Minimax Pathology in General

What remains to be considered is whether constant position error or constant two-value error at the lowest level of search is more realistic. There is no pathology in the former case, while the latter case corresponds to Beal’s basic model and is pathological. Game-playing programs use real (or at least multiple) values in their evaluation functions. Position error is the most direct representation of the fallibility of these functions and there is no reason to believe that it should increase with the depth of search as shown in Figure 4. But can we expect two-value error to behave as shown in Figure 3? We will show here that the answer is “yes”. Game-playing programs are generally not concerned with two-value error; if they were, one can easily imagine that it would be large in uncertain positions whose values are close to the threshold and small in clearly lost or won positions far from the threshold. If both sides are to have comparable chances to win at the root, the value at the root should be close to the threshold. Each level downwards from the root is one move away from the root position. Position values usually change gradually, so with each move, position values can be more different from the root value. Therefore the average distance of a position value from the threshold at lower levels can indeed be expected to

be larger than at higher levels, as was also stated by Scheucher and Kaindl [1998]. This is illustrated in Figure 6; darker area represents higher probability of two-value error.

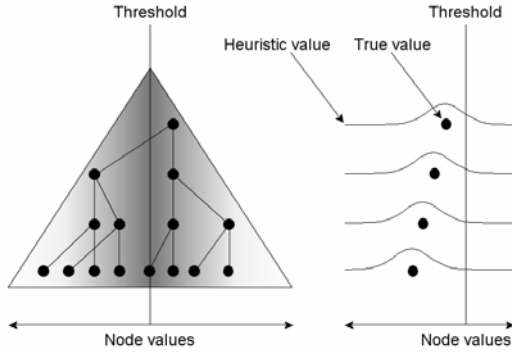


Figure 6: Distance of nodes values from the threshold and its relation to two-value error.

In game trees with independent node values, the effect of position error on two-value error can be analyzed mathematically. For simplicity, we will only consider  $b = 2$  and limit node values to  $[0, 1]$  interval.

Due to space limitation, we will only examine false losses. Consider the probability of a false loss at a node with true value  $X$  and heuristic value  $X - e$ , where  $e$  is the node's position error. False loss means that  $X > t$  and the heuristic value is on the other side of the threshold, i.e.  $X - e < t$ . The probability for such a mistake to happen at a node whose true real value is distributed according to distribution function  $F(x)$  is calculated according to equation (4).

$$P(X > t \wedge X - e < t) = P(t < X < t + e) = F(t + e) - F(t) \quad (4)$$

Consider now the probability of a false loss at different levels in a game tree. Let  $F_i(x)$  be the distribution function of node values at  $i$ -th level of the game tree. If  $i - 2$  is a max level,  $F_{i-2}(x)$  is calculated from  $F_{i-1}(x)$  according to equation (5).

$$F_{i-2}(x) = P(X_{i-2} < x) = P(X_{i-1} < x)^2 = F_{i-1}(x)^2 \quad (5)$$

If  $i - 1$  is a min level,  $F_{i-1}(x)$  is calculated from  $F_i(x)$  according to equation (6).

$$\begin{aligned} F_{i-1}(x) &= P(X_{i-1} < x) = 1 - P(X_{i-1} > x) = \\ &= 1 - P(X_i > x)^2 = 1 - (1 - P(X_i < x))^2 = 1 - (1 - F_i(x))^2 \end{aligned} \quad (6)$$

In order to calculate  $F_{i-2}(x)$  from  $F_i(x)$  in one step, (5) and (6) are joined into equation (7).

$$\begin{aligned} F_{i-2}(x) &= F_{i-1}(x)^2 = (1 - (1 - F_i(x))^2)^2 = \\ &= 4F_i(x)^2 - 4F_i(x)^3 + F_i(x)^4 \end{aligned} \quad (7)$$

We will show that the probability of a static false-loss error at higher levels is usually greater than at lower levels, which is expressed by inequality (8).

$$\begin{aligned} P(\text{false loss at level } i-2) &> P(\text{false loss at level } i) \\ F_{i-2}(t+e) - F_{i-2}(t) &> F_i(t+e) - F_i(t) \end{aligned} \quad (8)$$

Inequality (8) means that the difference between the values of distribution function at points  $t + e$  and  $t$  is greater at higher levels – in other words, that the distribution function is steeper at higher levels. Let us take as an

example our model from section 3:  $d_{\max} = 10$  and leaf values are distributed uniformly, therefore  $F_{10}(x) = x$ . Equation (7) can be used to calculate  $F_8(x) = 4x^2 - 4x^3 + x^4$ . Figure 7 shows  $F_8(x)$  and  $F_{10}(x)$  in our model.

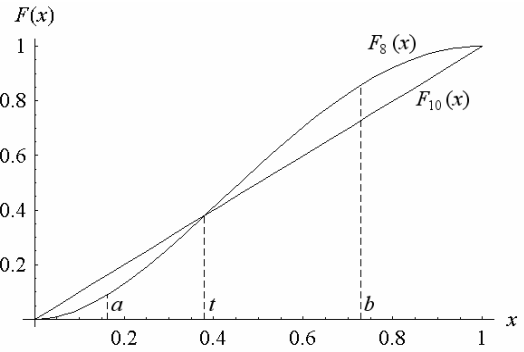


Figure 7: Distribution functions of node values at levels 8 and 10 in our model from section 3.

As can be seen in Figure 7,  $F_8(x)$  is steeper than  $F_{10}(x)$  between  $x = a$  and  $x = b$ . To determine where  $F_{i-2}(x)$  is steeper than  $F_i(x)$  independently of  $F_i(x)$ , inequality (9) must be solved; note that  $F_i(x)$  is written as  $F_i$ .

$$\begin{aligned} \frac{dF_{i-2}}{dF_i} &> \frac{dF_i}{dF_i} \\ 8F_i - 12F_i^2 + 4F_i^3 &> 1 \end{aligned} \quad (9)$$

The expression  $dF_{i-2}/dF_i$  as a function of  $F_i$  is shown in Figure 8.

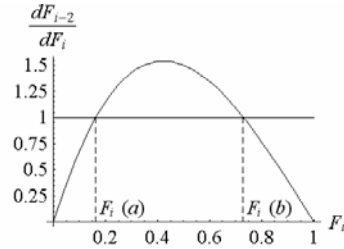


Figure 8:  $dF_{i-2}/dF_i$  as a function of  $F_i$ .

With the help of Figure 8 we can solve inequality (9):  $0.1624 < F_i < 0.7304$ . Values  $a$  and  $b$  in Figure 7 are therefore  $a = F_{10}^{-1}(0.1624)$  and  $b = F_{10}^{-1}(0.7304)$ ; since  $F_{10}(x) = x$ , this means  $a = 0.1624$  and  $b = 0.7304$ . So whenever the values of  $F_i(t+e)$  and  $F_i(t)$  are in the interval between 0.1624 and 0.7304, false-loss error at level  $i - 2$  is greater than false-loss error at level  $i$  for any distribution function  $F_i(x)$ . Since  $F_i(t) = k_i$ , if  $k_i$  is to be the same for all  $i$ , most reasonable distribution functions should satisfy the condition  $0.1624 < F_i(t) < 0.7304$ .

We now see that two-value error at higher levels is larger than at lower levels. But is it larger enough? If  $p_i$  and  $q_i$  when searching to depth  $d_{\max}$  were computed for all  $i$ , these values could be used for  $p_d$  and  $q_d$  when searching to depths  $d < d_{\max}$ . Under these conditions, two-value error at the root would be the same for searches to all depths – minimax would be neither pathological nor beneficial. If we can prove that given position error  $e$  causes greater two-value error at level  $i - 2$  if it is introduced at level  $i - 2$  (denoted

$P_1$ ) than if it is introduced at level  $i$  and then backed-up to level  $i - 2$  (denoted  $P_2$ ), we will also have proven that two-value error as a result of depth-independent position error decreases from the leaves towards the root sufficiently to make minimax non-pathological.  $P_1$  and  $P_2$  are calculated using equations (4) and (7) in different order, resulting in equations (10) and (11).

$$\begin{aligned} P_1 &= F_{i-2}(t+e) - F_{i-2}(t) = \\ &= 4F_i(t+e)^2 - 4F_i(t+e)^3 + F_i(t+e)^4 - \\ &\quad - (4F_i(t)^2 - 4F_i(t)^3 + F_i(t)^4) \end{aligned} \quad (10)$$

$$\begin{aligned} P_2 &= 4(F_i(t+e) - F_i(t))^2 - 4(F_i(t+e) - F_i(t))^3 + \\ &\quad + (F_i(t+e) - F_i(t))^4 \end{aligned} \quad (11)$$

We solved inequality  $P_1 > P_2$  on  $[0, 1]$  interval using Mathematica software, resulting in (12).

$$F_i(t) \leq F_i(t+e) \leq \frac{6 + F_i(t) - \sqrt{4 + 12F_i(t) - 7F_i(t)^2}}{4} \quad (12)$$

The upper limit for  $F_i(t+e)$  from inequality (12) is shown in Figure 9.

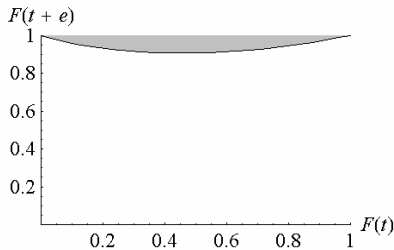


Figure 9:  $F(t+e)$  as a function of  $F(t)$ .

As can be seen in Figure 9, minimax will behave pathologically only when  $F(t+e)$  is close to 1 (shaded area). Since  $F(t)$  is generally not expected to be close to 1, this means a large position error. This is consistent with our observations from section 3 where there was no pathology except in some cases when static error was very large.

## 6 Conclusion

To analyze the pathology, we designed a minimax model with real-number position values. The model did not behave pathologically under a wide range of settings, as long as normally (or uniformly) distributed noise used to model the error of heuristic evaluations was independent of the depth of search. However, under these settings, two-value error was not independent of the depth of search, which is contrary to what was assumed in most of the previous research. Due to minimax relations among the true values, both types of error cannot be independent of the depth of search simultaneously, because at greater depths, position values are on average farther away from the threshold separating losses from wins; therefore at greater depths, the same position error causes smaller two-value error. In real games, this is caused by position values starting close to the threshold at the root and dispersing gradually as we advance downwards through the game tree. We showed

mathematically that in game trees with independent node values, two-value error at higher levels is also smaller than at lower levels. Furthermore, we showed that the reduction of the error is in most cases sufficient to eliminate the pathology. The pathology sometimes persists, particularly when the error is very large, but such cases are probably rare. Establishing which cases exactly are these is left for further work.

In summary, the explanation for the minimax pathology, or the absence of it, presented in this paper, is a necessary consequence of a real-value minimax model, and does not require any other assumption. Even if one were to claim that in a purely theoretical sense, real values are inappropriate, they are certainly necessary in practice. And the minimax pathology is considered pathological because it is at odds with what is observed in practice.

## References

- [Beal, 1980] Don F. Beal. An analysis of minimax. In *Advances in Computer Chess 2*, pages 103–109, 1980. Edinburgh University Press.
- [Beal, 1982] Don F. Beal. Benefits of minimax search. In *Advances in Computer Chess 3*, pages 17–24, 1982. Pergamon Press.
- [Bratko and Gams, 1982] Ivan Bratko and Matjaž Gams. Error analysis of the minimax principle. In *Advances in Computer Chess 3*, pages 1–15, 1982. Pergamon Press.
- [Luštrek, 2004] Mitja Luštrek. Minimax pathology and real-number minimax model. In *Proceedings of the Thirteenth International Electrotechnical and Computer Science Conference*, Volume B, pages 137–140, Portorož, Slovenia, September 2004. Slovenian Section IEEE.
- [Michon, 1983] Gerrard P. Michon. *Recursive Random Games: A probabilistic model for Perfect Information Games*. Ph.D. thesis, University of California, Los Angeles, 1983.
- [Nau, 1979] Dana S. Nau. *Quality of decision versus depth of search on game trees*. Ph.D. thesis, Duke University, 1979.
- [Nau, 1982] Dana S. Nau. An investigation of the causes of pathology in games. *Artificial Intelligence*, 19(3): 257–278, November 1982.
- [Pearl, 1983] Judea Pearl. On the nature of pathology in game searching. *Artificial Intelligence*, 20(4): 427–453, July 1983.
- [Sadikov et al., 2003] Aleksander Sadikov, Ivan Bratko, and Igor Kononenko. Search vs knowledge: Empirical study of minimax on KRK endgame. In *Advances in Computer Games: Many Games, Many Challenges*, pages 33–44, 2003. Kluwer Academic Publishers.
- [Scheucher and Kaindl, 1998] Anton Scheucher and Hermann Kaindl. Benefits of using multivalued functions for minimaxing. *Artificial Intelligence*, 99(2): 187–208, March 1998.