Independent-valued minimax: Pathological or beneficial?

Mitja Luštrek [a, *], Ivan Bratko [b], Matjaž Gams [a]

[a] *Jožef Stefan Institute, Department of Intelligent Systems, Jamova cesta 39, 1000 Ljubljana, Slovenia*
[b] *University of Ljubljana, Faculty of Computer and Information Science, Tržaška cesta 25, 1000 Ljubljana, Slovenia*

Abstract

Minimax search, which is used by most game-playing programs, is considered pathological when deeper searches produce worse evaluations than shallower ones. This phenomenon was first observed in theoretical analyses under seemingly reasonable conditions. It was most commonly explained by the lack of dependence between nearby positions in the analyses: if nearby positions have similar values, as is typically the case in real games, the pathology no longer occurs. In this paper we show that the pathology can be eliminated even without position-value dependence, by assigning enough different values to the positions and modeling the heuristic error as normally distributed noise that is independent of the depth in the game tree. This leads to the conclusion that minimax is less prone to the pathology than was previously thought and indicates the importance of the number of different position values.

Keywords: Minimax principle; Minimax pathology; Position independence; Real values

1. Introduction

Most game-playing programs are based on the minimax principle [1]. These programs represent the game using a game tree, in which the nodes correspond to positions and the edges correspond to moves. The tree is rooted in the current position. To choose a move, the descendants of the root must be evaluated in order that the program can move to the best descendant. This would ideally be accomplished by searching the tree all the way down to the terminal positions. With each terminal position is associated a true value, which corresponds to the outcome of the game in that position – for example, it may equal the final score of the game. These true values would then be propagated up the tree using the minimax principle. The minimax principle assumes that the player to move at the root (the max player) wishes to maximize the game score, whereas the opponent (the min player) wishes to minimize it. Therefore, the nodes at the levels of the tree where the max player is to move (the max levels) are assigned the maximum of the values of their descendants, and the nodes at the levels where the min player is to move (the min levels) are assigned the minimum of the descendants' values. Since searching the game tree to the terminal positions is often intractable, game-playing programs typically search it to a chosen depth and then use a heuristic evaluation function to approximate the true values at that depth. These heuristic values are then propagated up the tree and used to choose the move.

It is generally agreed that in game-playing programs using the minimax principle, searching the game tree to a greater depth produces better evaluations at the root, which leads to better play. A testimony to this is the increase in the performance of such programs as improvements in computers and algorithms made it possible to search deeper in the same amount of time [2]. However, the first attempts to explain the benefit of deeper minimax search mathematically yielded a surprising result: under seemingly reasonable conditions, minimaxing amplified the error of the heuristic evaluation function used to evaluate the leaves of the tree [3, 4]. As a consequence, deeper searches produced worse evaluations at the root. This phenomenon was termed the "minimax pathology" [4].

It was evident that the setting of these early mathematical analyses omitted some property of real games that eliminates the pathology. Several explanations were proposed, but eventually most researchers concluded that the property they were looking for is the dependence of the position values in the game tree [5-9]. Such a dependence is common in real games, because each move typically incurs only a small change

[*] Corresponding author.
E-mail addresses: mitja.lustrek@ijs.si (M. Luštrek), bratko@fri.uni-lj.si (I. Bratko), matjaz.gams@ijs.si (M. Gams).

in the position value, resulting in nearby positions having similar values. Dependence was, however, absent from the early, pathological minimax models.

After it was established that a sufficiently strong position-value dependence eliminates the pathology, the question of whether such a dependence is a *necessary* requirement for deeper minimax search to be beneficial has not been explored. This is the question that we focus on in our paper: we analyze the behavior of minimax on game trees with independent node values (independent to the extent permitted by the minimax relation) to determine whether deeper search can be beneficial in such a setting.

Several early mathematical models used only two heuristic values – losses and wins – to evaluate the positions. Scheucher & Kaindl [10] showed that the pathology can be inhibited by using multiple heuristic values. They built a model in which the heuristic values were arranged in a way inspired by chess. They observed that the values at lower levels of the game tree are more dispersed. This larger dispersion meant a larger proportion of more extreme values. In terms of losses and wins, it is less likely to misevaluate a position with an extreme value than a “close” one, so such a two-valued error in extreme positions was smaller. As a result, the error deeper in the tree was smaller, which made deeper search more beneficial and thus inhibited the pathology.

The research presented in this paper builds upon the work of Scheucher & Kaindl, but it has several important differences. Firstly, we eliminate the dependence of position values. Secondly, unlike Scheucher and Kaindl, we use not only multiple heuristic values, but also multiple true values[1]. This removes the need to explicitly model the smaller two-valued error in positions with extreme values. Instead, we obtain heuristic values by adding normally distributed noise to the true values, which achieves a similar result: the error in terms of losses and wins in positions with extreme values is smaller. Finally, we analytically prove that under reasonable conditions, deeper search in a real-valued[2] minimax model with a constant error of the evaluation function is beneficial. No additional properties of the game tree, such as the dependence of the position values, are required to achieve this.

To complete the analysis of independent-valued minimax search, we investigate the effect of the number of possible position values. Game-playing programs typically use a limited number of discrete values to evaluate positions instead of either real or two values. If the number of possible values is gradually reduced from an infinite number to two, minimax just as gradually transforms from beneficial to pathological. Real values and two values are just two extremes of a multivalued minimax model.

The paper is organized as follows. Section 2 presents the minimax pathology and gives an overview of the previous 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, whereas seemingly similar models used in previous research were. Section 5 explains whether deeper minimax search can also be expected to be beneficial in general. Section 6 deals with a minimax model using a limited number of discrete values. Section 7 concludes the paper with a discussion.

2. Minimax pathology and related work

The minimax pathology was discovered independently by Nau [4] and Beal [3]. Beal’s goal was to explain mathematically why heuristic values backed-up through minimax are more reliable than static heuristic values. To do this, he constructed a simple minimax model with five assumptions:

1. The game tree has a uniform branching factor.
2. The nodes of the tree can have two values: a loss or a win.
3. The node values are distributed so that at each level of the tree the proportion of losses for the side to move at that level is the same.
4. The node values within each level of the tree are independent of each other and distributed identically.
5. The error of the heuristic evaluation of 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 level in the game tree and the true value of the node.

[1] The need for multiple heuristic values is evident from the practice of game-playing programs. They are required to establish a *direction* of play, gradually moving toward a win. Without them, a program could maintain a won position without achieving the final goal. We define multiple true values as those that guide the program to play optimally. For further discussion, see Section 3.

[2] Game-playing programs typically use discrete values. We use real numbers to simplify the mathematical explanations and approximate them with floating-point numbers in simulations. For further discussion, see Section 3.

We next present Beal's basic model, although our analysis is mostly based on later work. For consistency with the following sections, we use the minimax representation instead of the original negamax representation, i.e., a node's value is viewed as lost or won from the perspective of the side to move at the root, not from the side to move at that node.

General tree-search definitions. Let b be the branching factor of the game tree, d the depth of a particular search, and $d_{\max}$ the maximum depth of search that we consider. The levels are numbered downwards: from 0 for the root to d for the lowest level of search. "Higher level" means a level higher in the game tree, which has a smaller index. Let k_i be the probability of a node at i-th level being lost. Two types of evaluation error can occur at a node: a loss can be mistaken for a win (false win) and a win can be mistaken for a loss (false loss). Let p_i and q_i be the probabilities of the respective types of error at the i-th level.

A node at a max level is only lost if all of its descendants are lost. If the level $i – 2$ is a max level, the relation between the values of $k_{i–2}$ and $k_{i–1}$ is described by Equation (1).

$$k_{i-2} = k_{i-1}{}^{b} \tag{1}$$

A node at a min level is lost if any of its descendants are lost. If level $i – 1$ is a min level, the relation between the values of $k_{i–1}$ and k_i is described by Equation (2).

$$k_{i-1} = 1 - (1 - k_i)^b \tag{2}$$

Beal's assumption 3 requires that $k_i = 1 – k_{i–1}$ for all i. The value of k_i that satisfies this condition at the max levels is c_b, a constant that depends on b (e.g., $c_2 = 0.3820$). The appropriate value of k_i at the min levels is $1 – c_b$.

At the max levels, false wins occur at nodes where all the descendants are lost, but at least one of them is a false win. Therefore, if level $i – 2$ is a max level, the probability of a false win $p_{i–2}$ is calculated according to Equation (3).

$$p_{i-2} = \frac{k_{i-1}{}^{b}}{k_{i-2}}(1-(1-p_{i-1})^b) = 1-(1-p_{i-1})^b \tag{3}$$

False losses at max levels occur at nodes where some descendants are won and all the won descendents are false losses, while all the other descendents are true losses. Therefore, if level $i – 2$ is a max level, the probability of a false loss $q_{i–2}$ is calculated according to Equation (4).

$$q_{i-2} = \frac{1}{1-k_{i-2}} \sum_{j=1}^{b} \binom{b}{j} (1-k_{i-1})^j \, k_{i-1}{}^{b-j} q_{i-1}^{j} (1-p_{i-1})^{b-j} \tag{4}$$

At min levels, the conditions for both types of error and consequently their equations are reversed. This results in Equations (5) and (6), which assume that level $i – 1$ is a min level.

$$p_{i-1} = \frac{1}{k_{i-1}} \sum_{j=1}^{b} \binom{b}{j} k_i{}^{j} (1-k_i)^{b-j} \, p_i^{j} (1-q_i)^{b-j} \tag{5}$$

$$q_{i-1} = \frac{(1-k_i)^b}{1-k_{i-1}}(1-(1-q_i)^b) \tag{6}$$

It turns out that if the same $p_d = q_d$ are used in searches to all depths, the error at the root, defined as $p_0\, k_0 + q_0\,(1 – k_0)$, increases with d. The reason for such behavior is that the error changes the proportion of the heuristic losses k' from k_d to k'_d. If k'_d does not equal c_b or $1 – c_b$, k' converges towards 0 or 1 at higher levels of the game tree. If k'_0 is near 0 or 1 and k_0 still equals c_b or $1 – c_b$, the error at the root must be large. Beal's basic model therefore exhibited a behavior opposite to the behavior observed in the practice of game playing, and this discrepancy became known as the minimax pathology.

In the years following the discovery of the pathology, several researchers attempted to find a flaw in Beal's basic model by attacking its assumptions (1 through 5 at the beginning of this section). An additional factor that affects the pathology, described below under number 6, was also found.

1. **Branching factor:** Michon [11] observed that the pathology depends on the probability distribution of the branching factor: game trees with a uniform branching factor tend to be pathological, whereas game trees with, for example, a geometrically distributed branching factor do not. It is not known whether real games have any of the non-pathological distributions, though. Nau proved that under certain conditions, a sufficiently large branching factor guarantees pathology [12]. Schrüfer's research [13] also indicates that the pathology is more likely when the branching factor is large. However, the pathology was often observed with small branching factors as well, so the role of large branching factors seems to be to amplify the pathology in situations that are already prone to it. Sadikov et al.'s [14, 15] research on chess endgames showed that the pathology appears when one side has a significantly larger number of moves to choose from than the other, i.e., when the branching factor is different for the min and max levels. Since the models of Beal and others were pathological even with a uniform branching factor, an alternating branching factor is apparently more of an additional reason for the pathology than a uniform branching factor is a reason for its absence. Luštrek et al. [16] discovered that as the branching factor increases, the number of distinct position values needed to avoid the pathology also increases. A study of several factors affecting the pathology by Nau et al. [17] confirmed this, as well as the tendency of the branching factor to amplify the pathology.
2. **Number of values:** Bratko & Gams [5] experimented with multiple position values – both true and heuristic. There was no special relation between the true and heuristic values in their model, such as heuristic values close to the corresponding true values being more likely. The resulting behavior was similar to the one with two values, i.e., pathological. Pearl [18] only considered multiple (and real) heuristic values – with similar results. Nau [7, 9] used two true and multiple heuristic values because this was the natural choice for the game he studied, not as an answer to the pathology. In another work [12], though, Nau indicated that more distinct heuristic values make the pathology less likely. Scheucher & Kaindl [10] constructed a model with two true and multiple heuristic values, which was not pathological. Multiple values were required to realistically model practical game playing, in particular the notion of a smaller heuristic error in more clearly lost (or won) positions, which are typically characterized by small (or large) heuristic values. Luštrek et al. [8, 16] used real values – both true and heuristic – to achieve a relation between them similar to what they observed in a chess program. Their model was also not pathological, but the main reason was probably the dependence of the position values. They did notice that real-valued heuristic estimates can eliminate the pathology even without dependence, but they did not pursue that line of research any further at the time [16]. Sadikov et al. [14, 15, 19] used multiple values in their analysis of the king-and-rook versus king chess endgame. They explained the pathology, but their explanation involves multiple values only indirectly and it might not apply to cases other than the endgames they studied. A study of several factors affecting the pathology by Nau et al. [17] showed that enough true and heuristic values eliminate the pathology.
3. **Proportion of losses:** Most researchers ensured in some way that the games they modeled were interesting, i.e., not almost certainly won by one side. In models with node values that are independent within each level, this can be achieved by having the values distributed so that k_i alternates between c_b and $1 - c_b$ [3, 5-7, 9, 17, 20]. If k_i were different, one could choose k_d and minimaxing would determine k_i for $i < d$, as described by Equations (1) and (2). If k_d were substantially different from c_b (or $1 - c_b$, depending on the side to move at level d), k_i would quickly converge to 0 or 1, coming very near 0 or 1 only a few levels above the leaves. This would mean that we are dealing with games that are almost certainly won by one side, even for relatively small d. Some researchers who did not use c_b as the proportion of losses, used other means to ensure that the games they studied were interesting [10, 12].
4. **Dependence of positions:** Most researchers agree that the similarity of positions close to each other in the game tree eliminates the pathology, although they arrived at this conclusion in different ways. Beal [6] did it by introducing into game trees a fraction of nodes with all the descendants having the same value. Bratko & Gams [5] discovered that homogeneous subtrees have reliably evaluated nodes at their roots, which have a stabilizing effect. Pearl [18] claimed that early terminations (e.g., a quick checkmate) are the important omission from the Beal's basic model. Since these also produce reliably evaluated nodes, they use the same mechanism to eliminate the pathology as the homogeneous subtrees proposed by Bratko & Gams. Nau et al. [7, 9] showed that the pathology disappears if the node values only change incrementally when moving through a game tree. Abramson [20] eliminated the pathology

in the game studied by Nau using early terminations (as suggested by Pearl) instead of incrementally changing node values. Luštrek et al. [8, 16] also observed that incrementally changing node values eliminate the pathology. Scheucher & Kaindl [10] proposed a model in which only heuristic values change incrementally. As a result, there was a larger proportion of more extreme values lower in the game tree, which – combined with a smaller static error in such nodes – led to the elimination of the pathology. In addition, the dependence of the node values itself probably also helped inhibit the pathology. Schrüfer [13] discovered that in two-valued game trees, the pathology does not occur if the probability of a node having only one winning descendant is sufficiently low. This is a somewhat weaker restriction than requiring most or even all the descendants of a given position to be either winning or losing, as Beal or Bratko & Gams did, but still similar. Althöfer [21] continued Schrüfer's work, but restricted his analysis to game trees with a branching factor of 2. In such trees the pathology does not occur if the probability that both descendants of a node have different values is below 0.5. Althöfer showed that in such trees no root-evaluation algorithm can eliminate the error introduced in the leaves of the game tree.

5. **Variation of error:** Nau [7, 9] used an evaluation function that was more accurate closer to the leaves of the game tree, but this did not prevent the pathology. Pearl [18] showed that in order to overcome the pathology, the error of the evaluation function must decrease by more than 50% every two levels. It is generally believed that, in practice, the quality of the evaluation cannot vary enough to account for the absence of the pathology. Scheucher & Kaindl [10], however, considered the decrease in the static error lower in the game tree to be the main reason for the absence of the pathology in real games. This decrease is the result of a smaller static error in more extreme positions, where a mistake is less likely and which are more common at greater depths. Luštrek et al. [16] already hinted at the ideas presented in this paper, but did not fully develop them.
6. **Conspiracy numbers:** The conspiracy number of a game tree is the number of leaves that must change their values simultaneously in order for the root value to change. Althöfer [21] noted that if the conspiracy number of a tree is low, an error in the leaves is more likely to affect the root. He showed that non-pathological trees, as defined earlier (the probability that both descendants of a node have different values is below 0.5) – if they are sufficiently deep – have a conspiracy number of at least 2. Lorenz & Monien [22] developed this idea further and extended it to arbitrary two-valued game trees. They introduced the notion of leaf-disjoint strategies. A strategy is a subtree of a game tree that is sufficient to prove that the game can be won or lost. Leaf-disjoint strategies are strategies that have no leaves in common. Lorenz & Monien showed that a tree is non-pathological if it has at least two leaf-disjoint strategies, and pathological otherwise. This is equivalent to the requirement that it has the conspiracy number of at least 2. Lorenz [23] successfully used conspiracy numbers as a measure of how difficult a position is to evaluate in a chess program to lead the opponent into difficult positions. Finally, Doerr & Lorenz [24] extended the work of Lorenz & Monien [22] to game trees with multiple values and random nodes in addition to min and max nodes.

Table 1 summarizes previous attempts to explain the absence of the pathology in real games. All the attempts were successful and the reasons they gave are in the shaded cells. The only exception may be Nau [12], who focused more on why the pathology does occur, although this also provides some insight into why it does not. The table shows that most explanations for the absence of the pathology in real games involve some form of position-value dependence. The branching factor also has a strong influence on the pathology, but it is generally not considered to be the reason for its absence in real games. Finally, conspiracy numbers, while studied by only a few authors, may well provide the most general explanation, at least for two-valued game trees. Each explanation is correct under the assumptions it makes, but we believe they do not paint the complete picture. Specifically, several authors claim that position-value dependence is necessary to eliminate the pathology, and that multiple position values and the variation of error cannot realistically achieve this on their own. In this paper, however, we describe a way to eliminate the pathology even when the values of positions are independent. We believe that our explanation is more fundamental than the ones requiring dependence, and that it requires weaker assumptions. We pay special attention to relating our work to that of Beal [3, 6] and Scheucher and Kaindl [10], since it is closest to ours. Compared to Scheucher and Kaindl, we consider the variation of error (column 5 in Table 1) to be *the* solution and the dependence of position values (column 4) not to be necessary, while we did not study the branching factor (column 1).

This paper is a substantial extension of our earlier conference paper [25]. The overview of related work in Section 2 is now much more comprehensive. Instead of presenting already-known equations for the negamax representation, we derive new equations for the minimax representation, which better support the rest of the paper. Section 3 interprets true values in our real-valued minimax model and supports the findings regarding the pathology with a test of statistical significance. It also presents experimental results showing the effect of a large error on the pathology, which confirm the theoretical results later in the paper. Section 5 is generalized to deal with both types of error, not only false losses; the final proof reflects that and also corrects a mistake in [25]. Section 6 is altogether new: not only with respect to the previous paper, but also in that it addresses an aspect of the minimax pathology that has hardly been studied before.

Authors	1. Branching factor	2. Multivalued (true/heuristic)	3. Proportion of losses	4. Dependence of positions	5. Variation of error	6. Conspiracy numbers
Abramson [20]	Uniform	No/yes	c_b	Early terminations are a solution	Yes, not a solution	Not studied
Althöfer [21]	2	No/no	Some restrictions due to 4	Most positions having descendants with the same value is a solution	No	Conspiracy number 1 implies the pathology
Beal [6]	Uniform	No/no	c_b	Clusters of identical values are a solution	No	Not studied
Bratko & Gams [5]	Uniform	No/no Yes/yes, not a solution	Various, not too close to 0 or 1, not a solution	Subtrees with mostly identical values are a solution	No	Not studied
Lorenz at al. [22-24]	Arbitrary	No/no Yes/yes	Arbitrary	Arbitrary	No	Conspiracy number of at least 2 is a solution
Luštrek et al. [8, 16]	Uniform	Yes/yes, a tool for realistic dependence	Not applicable	Incrementally changing values are a solution	Generally no	Not studied
Michon [11]	Geometrically distributed branching is a solution	No/no	Various	No	No	Not studied
Nau [7, 9]	Uniform	No/yes, only a tool	0.5 and c_b	Incrementally changing values are a solution	Yes, not a solution	Not studied
Nau [12]	Uniform, small branching is a partial solution	No/yes, many values are a partial solution	Only interesting game trees	Yes, to simplify calculations	No	Not studied
Pearl [18]	2	No/no No/yes, not a solution	Arbitrary	Early terminations are a solution	Yes, not a solution	Not studied
Sadikov et al. [14, 15, 19]	Different branching for min and max levels causes the pathology	Yes/yes, only a tool	Real game	Real game, measured to be present, so may be a part of the solution	No	Not studied
Scheucher & Kaindl [10]	Uniform, small branching is a partial solution	No/yes, a tool to for realistic dependence and variation of error	Only interesting game trees	Incrementally changing heuristic values are a part of the solution	Smaller static error in positions with more extreme values is a part of the solution	Not studied
Schrüfer [13]	Uniform, small branching is a solution	No/no	Some restrictions due to 4	Few positions with only one winning move are a solution	No	Not studied
Nau et al. [17]	Uniform, small branching contributes to the solution	Yes/yes, many values are a solution	c_b	Incrementally changing values contribute to the solution	No	Not studied

Table 1. Summary of the explanations for the absence of the pathology in real games.

3. A minimax model based on real values

Multivalued position evaluations are, in practice, used by both game-playing programs and humans. There is little doubt that the number of possible values should not be smaller than the number of possible final outcomes of the game. Some games indeed have multivalued outcomes (Othello, tarok etc.), but in others the outcome can only be a loss, a win and perhaps a draw (chess, checkers etc.). However, if we assume that all the information about which positions are preferable is contained in the evaluation function, there is a need for multiple values even in games with only two possible outcomes. 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 a loss. This probability can then be used to move to positions where the opponent is more likely to make a mistake. Even more importantly, in a winning position, even a perfect two-valued evaluation function can only guarantee to maintain a won position indefinitely without actually winning. This problem could be circumvented by storing past positions in order to avoid repeating them. However, even though such measures may eventually lead a two-valued evaluation function to victory, the resulting play would be awkward and quite unacceptable, at least in a situation where invincible computer play would be used as exemplary play to teach humans. Also, long-winded play may lead to exceeding the time constraints in some games.

In essence, multiple values are necessary to differentiate between multiple winning (or losing, if only such are available) moves. Scheucher & Kaindl [10] showed this in practice for heuristic values: a two-valued evaluation function in chess performed poorly compared to a multivalued one. Lorenz [23] compared several chess programs on endgames for which the true position values were known. Programs that knew the true values and were also equipped with a measure of difficulty of position performed better than a program that chose among true winning/losing positions randomly.

Multiple heuristic values are needed to play well, but what about true values? Heuristic values are often considered to approximate true values, so if there are multiple heuristic values, there should also be multiple true values. This is the case in our model. True values are true in the sense that they guide a program to play optimally; they need not have any direct meaning related to the game itself. In our minimax model we use real values instead of multiple discrete values. This simplifies the mathematical explanation in Section 5 and avoids the complex issue of how many distinct values to use. This issue is addressed in Section 6, while a more thorough treatment can be found elsewhere [17]. In computer simulations we approximate the real values with floating-point numbers.

The minimax model that we propose introduces real-number position values into Beal's basic model. This real-valued model will turn out to be non-pathological. Like Beal's model described at the beginning of Section 2, our model makes five assumptions:

1. The game tree has a uniform branching factor (identical to Beal's).
2. The nodes of the tree have real-number values (different from Beal's).
3. If the nodes' real-number values are converted to losses and wins by means of an appropriate threshold, they are distributed so that at each level of the tree the proportion of losses for the side to move at that level is the same (identical to Beal's after the conversion to losses and wins).
4. The node values within each level of the tree are independent of each other and distributed identically (identical to Beal's).
5. The error of the heuristic evaluation of a node at the lowest level of search, being normally distributed noise, is independent of the level in the game tree and the true value of the node (differs from Beal's only in the definition of error).

To study the pathology in Beal's basic model, only Equations (1) through (6) and the value of c_b are required. Because the relation between the true and heuristic node values in our model is more complex, we used simulations: we built a number of game trees with randomly generated node values, measured their properties and averaged them. A game tree built according to our model is assigned true values to the leaves at level $d_{\max}$. They are distributed uniformly in the [0, 1] interval. The true values of the internal nodes are obtained by backing up the true leaf values using the minimax rule. When searching to a depth d, the heuristic values at the level d (that is, at the lowest level of search) are generated by corrupting the true values at the same level with normally distributed noise representing the error of the heuristic evaluation function. The standard deviation of the noise is denoted σ. Finally, the heuristic values of the nodes at the levels above d are obtained by backing up the corrupted values at level d using the minimax rule. The procedure is illustrated in Figure 1.

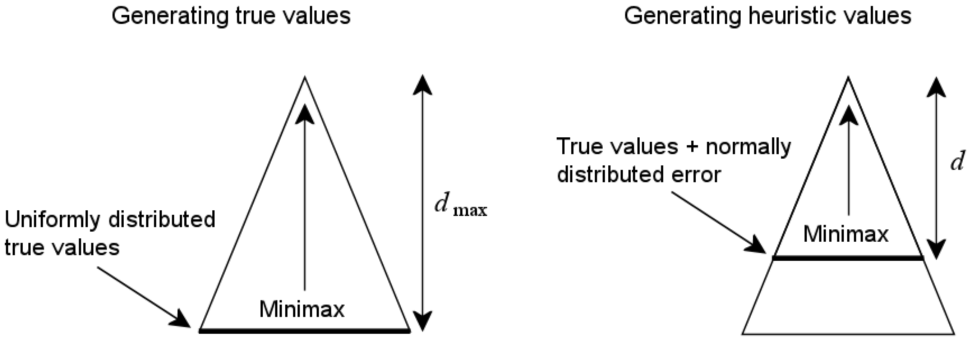
Generating true values
Generating heuristic values
Uniformly distributed true values
Minimax
$d_{\max}$
True values + normally distributed error
Minimax
d

Figure 1. Construction of game trees.

Error definitions. Two types of error can be observed at a node of a real-valued game tree: *RV* (meaning real-value) *position error* is the absolute difference between the true and the heuristic value of the node; *RV move error* is the probability of choosing a wrong move because of the RV position error at the node's descendants. However, neither type of error corresponds directly to the error most commonly used in two-valued models such as Beal's: the probability of mistaking a loss for a win and vice versa. This type of error will be termed *2V* (meaning two-value) *position error*. For the sake of completeness, we also define *2V move error* at a node as the probability of choosing a wrong move because of the 2V position error at the node's descendants. All types of error are most often measured at the root of the game tree or at the lowest level of search. The former is called the root error and the latter, the static error.

With the exception of the first experiment, the results for the 2V move error are not shown in the paper, because the qualitative behavior of the 2V move error is similar to that of its position counterpart and they are not needed for the comparison with Beal's basic model.

3.1. Conversion from real to two values

In order to compare our real-valued model to Beal's basic two-valued model, the real values are converted to losses and wins. This is 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, k_i alternates between c_b and $1 – c_b$. We assumed that the trees have an even height with max to move at the root. Since the true values at the leaves are distributed uniformly in the [0, 1] interval, $k_d = c_b$ is achieved by setting $t = c_b$. Even though real-valued minimaxing is used, k_i behaves as desired for $i > 0$. This happens for two reasons. First, the leaf values in our real-valued model, when converted to two values, correspond exactly to the leaf values in Beal's basic model. The probability of a loss at a leaf with the value X is $P\ (X < t)$, which for a uniform distribution in the [0, 1] interval and $t = c_b$ equals c_b. In Beal's model, the probability of a loss at each leaf is k_d, which also equals c_b. Second, real- and two-valued minimaxing are equivalent in the sense that performing minimax on the 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 2. The nodes are placed above the positions on the horizontal axis corresponding to their real values and are labeled as losses ("–") or wins ("+"). The minimax rule holds both for the real values and for the losses and wins. We can also see that the real values match the corresponding loss/win labels: the nodes to the left of the threshold are labeled with "–" and the nodes to the right with "+".

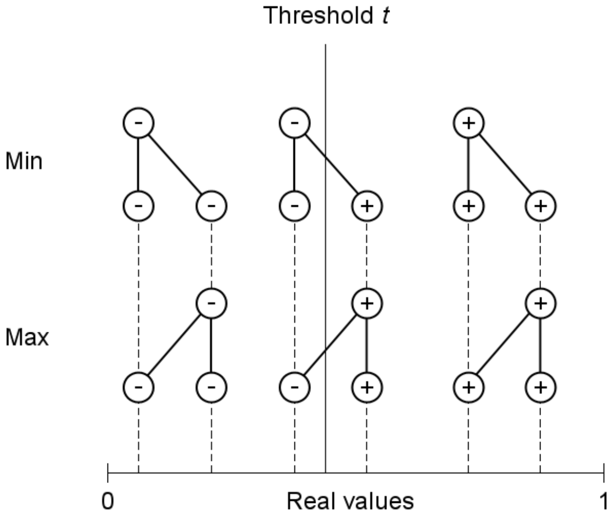

Figure 2. Equivalence of real- and two-valued minimaxing.

3.2. *Experimental results*

This subsection presents the experimental findings regarding the pathology for game trees with $b = 2$ and $d_{\max} = 10$; the results for larger branching factors and depths are similar. The results are averaged over 10,000 trees with randomly generated true values. For each tree, there were 10 repetitions with randomly generated heuristic values for each d. The same sample size was used in all the experiments in this paper. Figure 3 shows all four types of root error with respect to the depth of search; $\sigma = 0.1$.

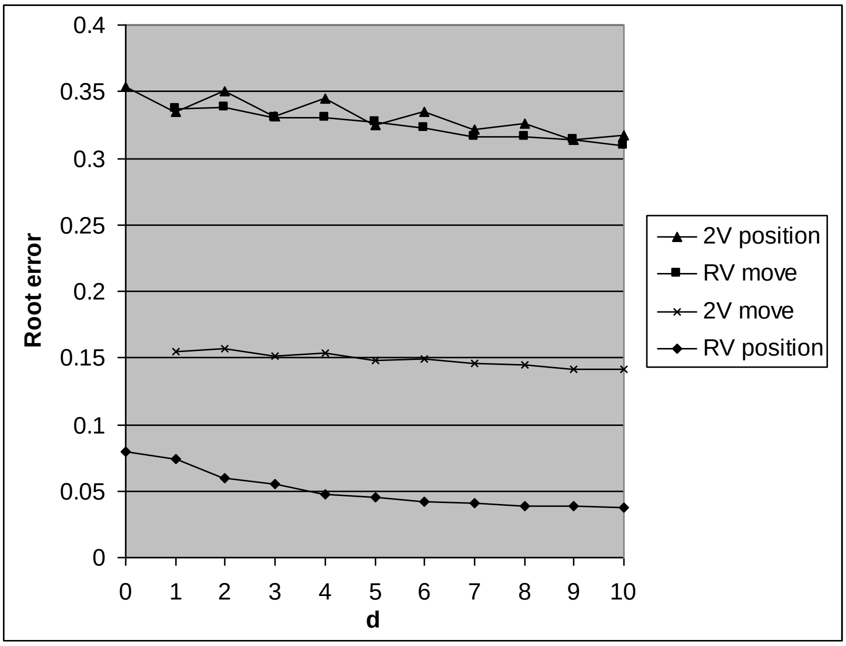

Figure 3. Root error with respect to the depth of search, $\sigma = 0.1$.

As can be seen in Figure 3, the root error of all four types decreases with the depth of search (with the exception of the even/odd level fluctuations of the 2V position and to a lesser extent the 2V move error), which means that deeper search is worthwhile. This is different from Beal's original results [3]. The reason for the difference will be analyzed in Section 4.

Since some of the curves in the figure are rather flat, we tested the statistical significance of the observation that deeper search is worthwhile versus the null hypothesis that it does not matter how deep we search. To do so, all pairs of root errors after searching to the depths i and $i + 2$ were compared for all of the randomly generated 10,000 game trees. The search depths were separated by two levels to eliminate the effect of the even/odd level fluctuations of the 2V errors. Since root errors do not follow any common distribution, the non-parametric paired Wilcoxon-Mann-Whitney's test was used. For both types of RV error at all pairs of depths, the test confirmed that the error decreases with increasing depth ($p < 0.01$, in most cases much lower). For the two types of 2V error, the significance level was in three cases higher ($p < 0.05$) and in the case of the 2V position error between depths 0 and 2, the decrease was not statistically significant.

Additionally, we investigated whether the absence of the pathology occurs only under conditions matching Beal's or more generally. We tried a uniform distribution of noise and a normal distribution of leaf values as well as different forms of dependence [16] between the node values. None of the experiments yielded the pathology. The only parameter capable of causing pathological behavior turned out to be the standard deviation of the noise. Figure 4 shows the root RV move error and Figure 5 the root 2V position error with respect to the depth of search for different σ. Since even/odd level fluctuations obscure the trends in Figure 5, linear approximations are also shown. The RV position error was not pathological, regardless of σ.

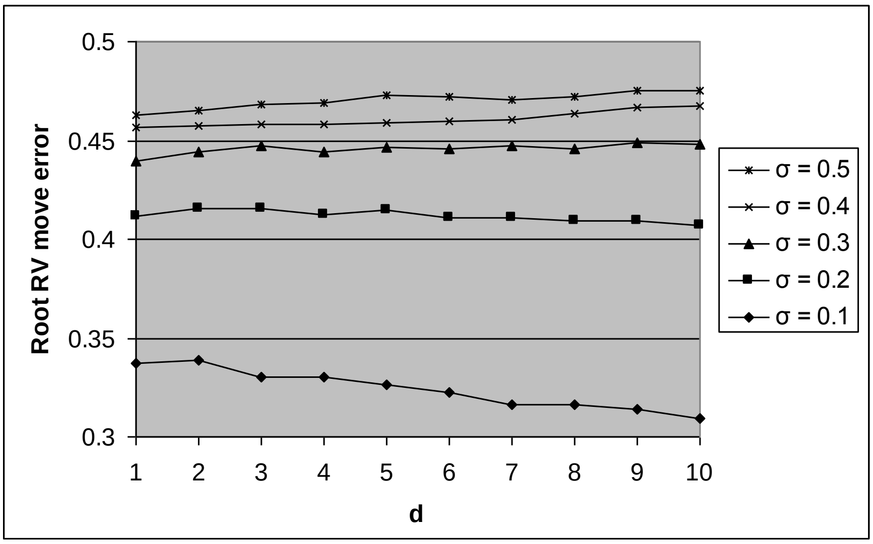

Figure 4. Root RV move error with respect to the depth of search for different σ.

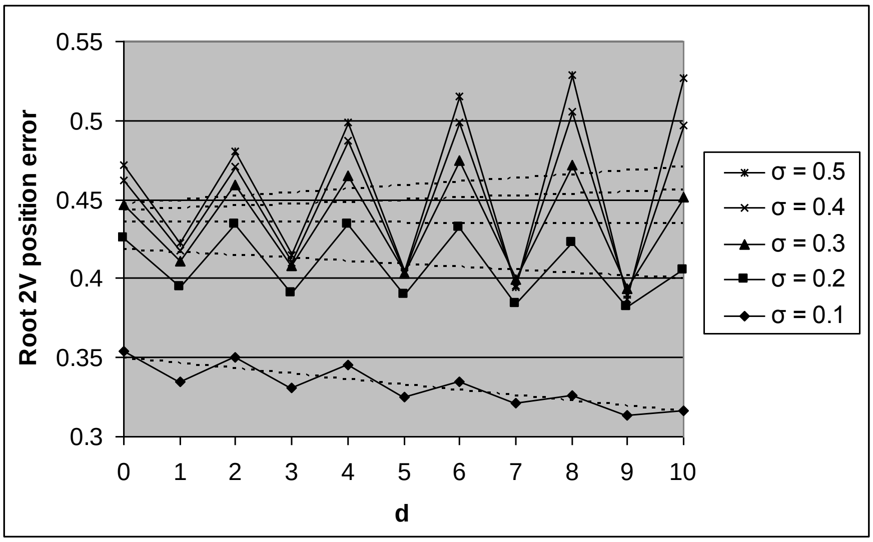

Figure 5. Root 2V position error with respect to the depth of search for different *σ*.

As can be seen in Figure 4 and Figure 5, the pathology only appears when *σ* is above 0.3, which means that the static error is near 0.45. Since the RV move error or the 2V position error 0.5 is the point where the evaluations become completely random, this seems to be of little practical importance.

In summary, we constructed a minimax model that on the surface appears similar to Beal's basic model, except that it uses real values instead of losses and wins. Unlike Beal's, however, our model is very resistant to the pathology and succumbs to it only with an implausibly bad evaluation function.

4. Why is our model not pathological

Considering that our model is very similar to Beal's, why is it not pathological? The reason why the RV evaluations are not pathological is explained in our earlier paper [16]. In this paper we attempt to give an answer for the 2V evaluations. To do that, we must examine the static 2V position error. Beal's assumption 5 states that it should be constant with respect to the depth in the game tree, but in our model, the static RV position error is constant instead (which is achieved by using normally distributed noise with the same standard deviation at all levels). The static 2V position error in our model is shown in Figure 6 for *σ* = 0.1.

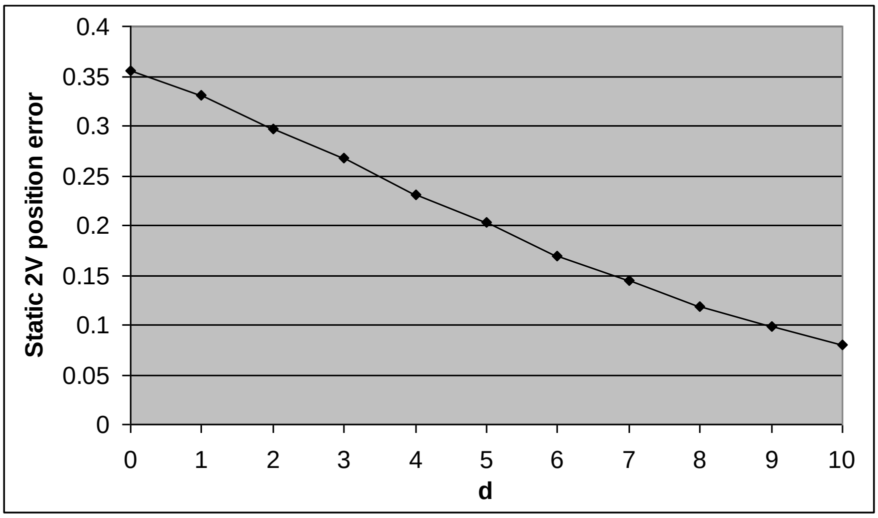

Figure 6. Static 2V position error with respect to the depth of search.

As can be seen in Figure 6, the static 2V position error decreases with the depth of search. This is different from Beal's assumption 5 and gives the first hint as to why our model is not pathological, even though Beal's is. Both models, Beal's and ours, assume that the quality of the static heuristic evaluations is equal at all levels of the game tree. Beal's model assumes a constant static 2V error and our model assumes a constant static RV error. However, when converting the RV model into the corresponding 2V model, it turns out that the constant static RV error does not convert into a constant static 2V error.

Let us now adjust our model so that it will conform to Beal's assumption 5. To do that, we need to adjust the noise introduced at the lowest level of search so that the static 2V position error will always be the same. Figure 7 shows the static RV position error when the static 2V position error is always 0.1.

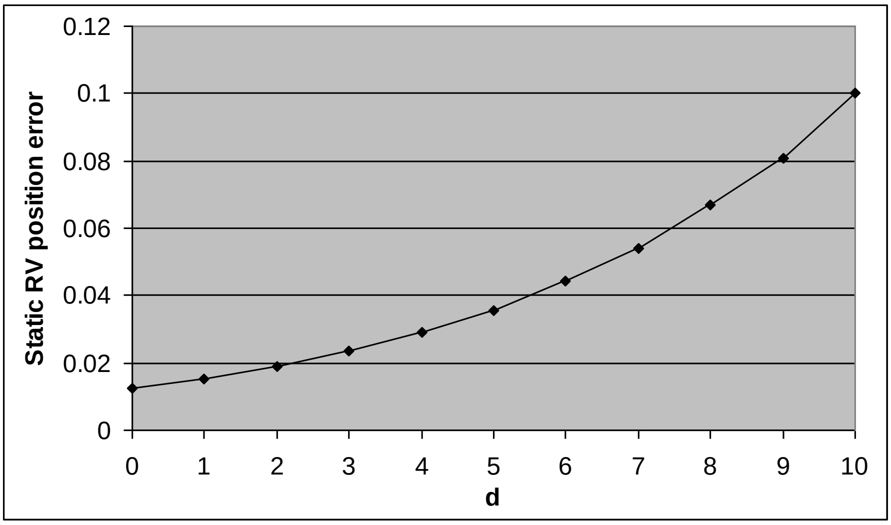

Figure 7. Static RV position error with respect to the depth of search when the static 2V position error is 0.1 at all depths.

The situation in Figure 7 corresponds to Beal's basic model (a constant static 2V error). We can see that the static RV position error is considerably larger at greater depths of search. In such circumstances, the root error that is also larger at greater depths of search should come as no surprise. In Beal's model, the root error is known to be pathological. Figure 8 shows the root 2V position error in our model under Beal's assumption 5, i.e., with the static 2V position error always equal to 0.1; the results for Beal's model are shown for comparison.

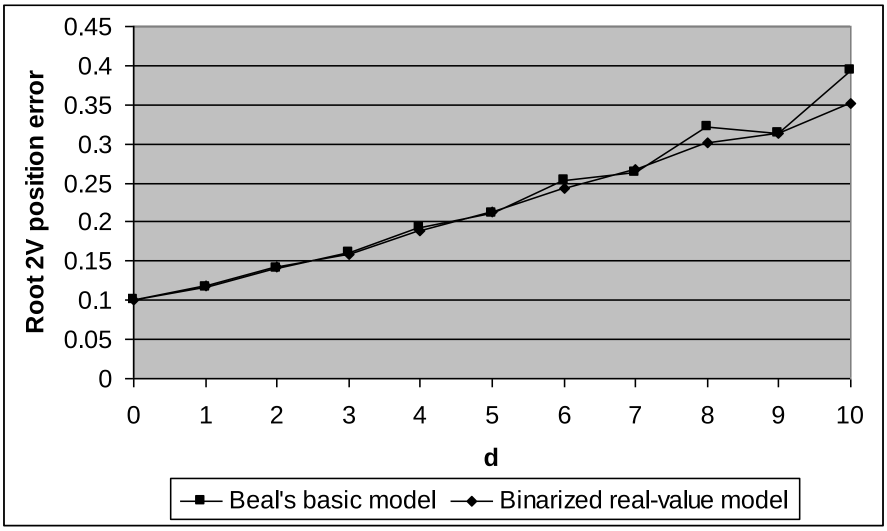

Figure 8. Root 2V position error with respect to the depth of search when the static 2V position error is 0.1 at all depths.

As can be seen in Figure 8, the results for the binarized real-valued model and the results for Beal's basic model match quite well: they are both pathological. The matching is not perfect because in the real-valued model the probability of a false win at the lowest level of search is higher than the probability of a false loss. False wins occur because the values from the [0, *t*) interval cross over the threshold and false losses occur because the same happens to the values from the (*t*, 0] interval. Since $t = c_2 = 0.3820$, the former interval is smaller. Therefore, node values below the threshold are, on average, closer to it, and hence false wins are more likely. The ratio of the probability of a false win : probability of a false loss is $(1 – t) : t$. If the overall 2V position error is to remain 0.1, Equations (7) must hold in Beal's model.

$$\frac{p_d}{q_d} = \frac{1-t}{t}$$
$$p_d\, k_d + q_d\, (1 – k_d) = 0.1 = p_d\, t + q_d\, (1 – t) \qquad (7)$$

Since it follows from the first equation that $p_d\, t = q_d\, (1 – t)$, it can easily be seen from the second equation that the appropriate settings for Beal's model are $p_d = 0.05 / t$ and $q_d = 0.05 / (1 – t)$. If the 2V position errors at higher levels are computed as described in Section 2, they match the measurements from our model perfectly. The adjusted Beal's model is of course still pathological, as seen in Figure 8.

To summarize this section, Beal's basic model assumes the static 2V position error to be constant with respect to the depth in the game tree, whereas our model assumes the static RV position error to be constant. These assumptions may appear equivalent, but they are in fact incompatible, at least under most distributions of game-tree values conforming to the minimax relation. The former assumption results in pathological behavior and the latter does not. The reason for the incompatibility will become clear in the next section.

5. The minimax pathology in general

In Section 4 we analyzed the assumption of the static evaluation error being independent of the depth in the game tree. We found that the effect of this assumption regarding the pathology depends on whether we use RV or 2V position values. Therefore, the two assumptions, first that static RV position error is constant, and second that the static 2V position error is constant, are essentially different. What remains to be seen is which of them is more realistic.

To judge whether the static RV or 2V position error should be constant, Figure 6 (showing the static 2V position error when the static RV position error is constant) and Figure 7 (showing the static RV position error when the static 2V position error is constant) should be considered. Game-playing programs use multiple values in their evaluation functions. The static RV position error is a better approximation of the fallibility of these functions than the static 2V position error. There is no reason to believe that the static RV position error should increase with the depth of search, as shown in Figure 7. However, can we expect the static 2V position error to decrease, as shown in Figure 6? Game-playing programs are generally not concerned with the 2V error, but if they were, one can easily imagine that the error would be large in uncertain positions whose values are close to the threshold separating losses from wins, and small in clearly lost or won positions far from the threshold. If there are more positions with values far from the threshold lower in the game tree, the static 2V error should behave as seen in Figure 6.

In real games, positions lower in the game tree have values farther from the threshold because each move typically incurs only a small change in the position value [26]. If the root position is to be interesting, i.e., not clearly lost for one side, its value should be close to the threshold. Each level downwards is one move away from the root position, so the values at each lower level can be farther from the root value. This was the case for heuristic values in the model of Scheucher and Kaindl [10], who discovered the relation between the greater dispersion of position values lower in the game tree and the pathology.

In game trees with independent node values, there is also a reason for positions lower in the game tree to have values farther from the threshold. Pearl's minimax convergence theorem [27] and Nau's last player theorem [28] both state that when the search depth increases, the value at the root of the game tree approaches the equilibrium point c_b. This is true for any distribution of leaf values, as long as they are independent of each other. If both sides are to have comparable chances to win at the root, the threshold cannot be far from c_b, so the root value is closer to the threshold than the values lower in the game tree. In the game trees described in this paper, the convergence occurs for true values and partially also for static heuristic values, since they are linked to the true values by the depth-independent heuristic error.

In the rest of this section we will analyze the effect of a constant static RV position error on the static 2V position error mathematically under the assumption of node-value independence. We will express the results of our analysis with two theorems. For simplicity, we will only consider $b = 2$ and limit the node values to the [0, 1] interval. Unlike in the model from Section 3, where the true leaf values were distributed uniformly, we will no longer assume any particular distribution of node values. We will, however, make a simplifying assumption regarding the static RV position error. In the model from Section 3, the error was a random variable with the same distribution at all nodes, so the expected error was always the same. In this section, we will assume that the actual error is always the same.

5.1. Static 2V position error lower in the tree is smaller

Theorem 1. We will compare the 2V position error at levels i and $i - 2$. Let $F_i\ (x)$ be the distribution function of the true values of the nodes at the i-th level of the game tree. Let e be the RV static position error at levels i and $i - 2$. The corresponding static 2V position error computed by means of the threshold t at the level $i - 2$ is larger than at the level i if $0.1624 < F_i\ (t - e) < F_i\ (t + e) < 0.7304$.

Comments on Theorem 1. The theorem states that, generally, the static 2V position error at higher levels of the game tree is larger than at lower levels, which is what we gave as the reason for the absence of the pathology in the independent-valued minimax. It holds for any distribution of position values that satisfies the stated condition. But can we expect the position values to satisfy this condition? Remember that $F_i\ (t) = \mathrm{P}\ (X_i < t) = k_i$. The value k_i is the probability of a loss at level i, but it can also be interpreted as the expected strength of the position at that level. In an interesting position where the players are evenly matched, it must alternate between a low value (k_i) when the max player is to move and a high value ($1 - k_i$) when the min player is to move. This is because when the max player is to move, he needs to have just enough disadvantage so that his move, which improves the position for him, puts the opponent at an equal disadvantage. In this section we do not require these two values to be exactly c_b and $1 - c_b$ as we did in our model from Section 3. However, for most reasonable distribution functions $F_i\ (t)$ cannot be very far from c_b (this section assumes $b = 2$, so $c_2 = 0.3820$) and can be expected to satisfy the condition $0.1624 < F_i\ (t) <$

0.7304. The same is true for $F\ (t - e)$ and $F_i\ (t + e)$, unless the error is very large. This is confirmed by the experimental results shown in Figure 4 and Figure 5: when the error was very large, the pathology occurred.

Proof of Theorem 1. We must first express the 2V position error in terms of the distribution function of the true node values $F_i\ (x)$. Let us start with false losses. Consider the probability of a false loss at a node with the true RV value X and the heuristic value $X - e$. A false loss means that $X > t$ and the heuristic value is on the other side of the threshold, i.e., $X - e < t$. The distribution function of a random variable X is defined as $F_i\ (x) = \mathrm{P}\ (X < x)$. So the probability of a false loss at a node whose true real value is distributed according to $F_i\ (x)$ is given in Equation (8).

$$\mathrm{P}(X > t \wedge X - e < t) = \mathrm{P}(t < X < t + e) = F_i(t + e) - F_i(t) \tag{8}$$

A false win means that the true value is a loss, i.e., $X < t$, and the heuristic value is a win, i.e., $X + e > t$. The probability of such a mistake is expressed by Equation (9).

$$\mathrm{P}(X < t \wedge X + e > t) = \mathrm{P}(t - e < X < t) = F_i(t) - F_i(t - e) \tag{9}$$

The 2V position error includes both false losses and false wins. As a consequence, the 2V position error at the i-th level of the game tree ($2VPE_i$) is simply the sum of the above two probabilities, as given in Equation (10). Figure 9 illustrates how the 2V position error is expressed in terms of the distribution function.

$$2VPE_i = F_i\ (t + e\ ) - F_i\ (t) + F_i\ (t) - F_i\ (t - e) = F_i\ (t + e) - F_i\ (t - e) \tag{10}$$

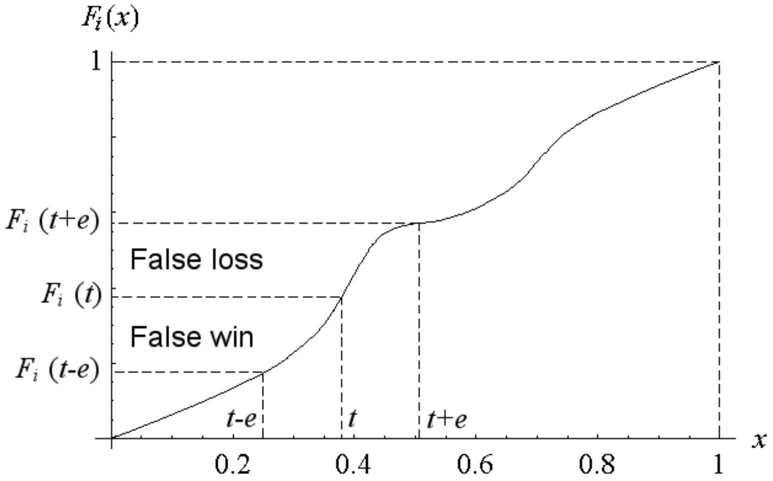

Figure 9. 2V position error expressed in terms of the distribution function of node values.

Let us now examine how the distribution of node values changes through the levels of a game tree. $F_i\ (x)$ is the distribution function of the node values at the i-th level of the game tree. If level $i - 2$ is a max level, the probability of a node value at level $i - 2$ being smaller than a given x equals the probability of both of its descendants being smaller than x. Since the descendants' values are independent, $F_{i-2}\ (x)$ is calculated from $F_{i-1}\ (x)$ according to Equation (11).

$$F_{i-2}(x) = \mathrm{P}(X_{i-2} < x) = \mathrm{P}(X_{i-1} < x)^2 = F_{i-1}(x)^2 \tag{11}$$

Analogously, if level $i - 1$ is a min level, the probability of a node at level $i - 1$ being larger than a given x equals the probability of both of its descendants being larger than x. Therefore, $F_{i-1}\ (x)$ is calculated from $F_i\ (x)$ according to Equation (12).

$$\begin{aligned} F_{i-1}(x) &= \mathrm{P}(X_{i-1} < x) = 1 - \mathrm{P}(X_{i-1} > x) = \\ &= 1 - \mathrm{P}(X_i > x)^2 = 1 - (1 - \mathrm{P}(X_i < x))^2 = 1 - (1 - F_i(x))^2 \end{aligned} \tag{12}$$

In order to calculate F_{i-2} (x) from F_i (x) in one step, Equations (11) and (12) are joined into Equation (13). The choice of having the max level above the min level is arbitrary and switching them would not affect the conclusions.

$$\begin{aligned} F_{i-2}(x) &= F_{i-1}(x)^2 = (1 - (1 - F_i(x))^2)^2 = \\ &= 4F_i(x)^2 - 4F_i(x)^3 + F_i(x)^4 \end{aligned} \tag{13}$$

We are attempting to prove that the static 2V position error at level i – 2 is larger than at level i. This is expressed by Inequality (14).

$$\begin{gathered} 2VPE_{i-2} > 2VPE_i \\ F_{i-2}\,(t + e) - F_{i-2}\,(t - e) > F_i\,(t + e) - F_i\,(t - e) \end{gathered} \tag{14}$$

Inequality (14) means that the difference between the values of the distribution function at points $t + e$ and $t - e$ is larger at higher levels. This is certainly true if the distribution function is steeper at higher levels. Let us take as an example our model from Section 3: $d_{\max}$ = 10 and the leaf values are distributed uniformly. F_i (x) is thus no longer arbitrary – instead, F_{10} $(x) = x$ and Equation (13) can be used to calculate F_8 $(x) = 4\,x^2 - 4\,x^3 + x^4$. Figure 10 shows F_8 (x) and F_{10} (x) in our model.

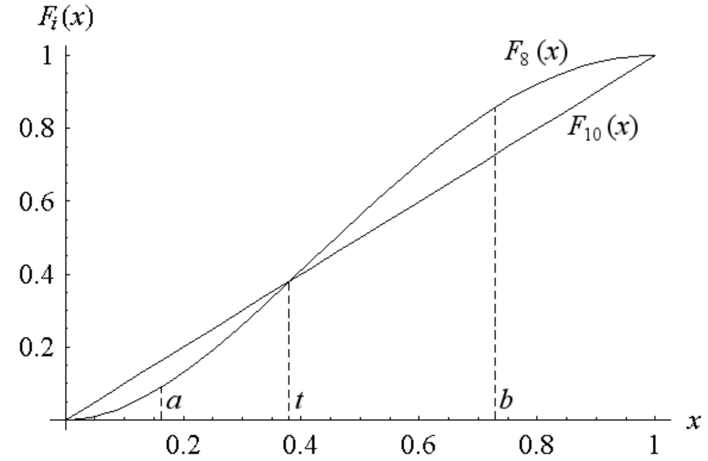

Figure 10. Distribution functions of the node values at levels 8 and 10 in our model from Section 3.

As can be seen in Figure 10, F_8 (x) is steeper that F_{10} (x) between $x = a$ and $x = b$. This means that the static 2V position error at higher levels of the game tree is larger than at lower levels when $t + e$ and $t - e$ are in the (a, b) interval. To determine where F_{i-2} (x) is steeper than F_i (x) independently of F_i (x), Inequality (15) must be solved; F_{i-2} (x) is expressed as a function of F_i (x) given in Equation (13) and F_i (x) is written as F_i.

$$\begin{gathered} \frac{dF_{i-2}(F_i)}{dF_i} > \frac{dF_i}{dF_i} \\ \frac{d(4F_i^{\,2} - 4F_i^{\,3} + F_i^{\,4})}{dF_i} > \frac{dF_i}{dF_i} \\ 8\,F_i - 12\,F_i^2 + 4\,F_i^3 > 1 \end{gathered} \tag{15}$$

The expression dF_{i-2} / dF_i as a function of F_i is shown in Figure 11. We are looking for the values F_i (a) and F_i (b), between which the expression is greater than 1.

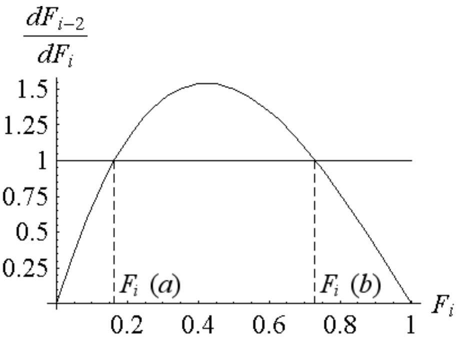

Figure 11. dF_{i-2} / dF_i as a function of F_i.

The solution of Inequality (15) is $0.1624 < F_i < 0.7304$, where $0.1624 = F_i$ (a) and $0.7304 = F_i$ (b). This concludes the proof of Theorem 1. We can only add that the values a and b are dependent on F_i, which is why they are not included in the theorem. For the example in Figure 10, $a = F_{10}^{-1}$ (0.1624) and $b = F_{10}^{-1}$ (0.7304); since F_{10} (x) = x, this means $a = 0.1624$ and $b = 0.7304$.

5.2. Static 2V position error lower in the tree is sufficiently *smaller*

Theorem 1 states that the 2V position error at lower levels is generally smaller than at higher levels. But is it sufficiently smaller? If p_i and q_i when searching to the depth $d_{\max}$ were computed for all i, these values could be used for the static error p_d and q_d when searching to depths $d < d_{\max}$. Such a static error could be called neutral: it produces the same root 2V position error regardless of the depth of search, making minimax neither pathological nor beneficial. Since Beal’s basic model, which has a depth-independent static 2V position error, is pathological, the neutral static error must increase from the leaves towards the root to counter this. This leads to the second theorem.

Theorem 2. Let P_{m} be the 2V position error at level $i - 2$, caused by the RV position error e at that level, and P_{n} the neutral error at level $i - 2$. $P_{\mathrm{m}} > P_{\mathrm{n}}$ as long as the error of the heuristic evaluation function is small enough and the distribution function of the node values is sufficiently “smooth” that F_i (t) can be approximated with (F_i ($t - e$) + F_i ($t + e$)) / 2 (what “smooth” means will be clarified in the next paragraph).

Comments on Theorem 2. The theorem means that the static 2V position error always increases from the leaves towards the root sufficiently to make minimax beneficial – in other words, that minimax with a constant static RV position is never pathological. This is subject to two conditions that must be satisfied for the approximation $F_i(t) \approx (F_i(t - e) + F_i(t + e)) / 2$ to be reasonably accurate. First, the error should not be too large, which matches the results of the experiments in Section 3 – the pathology was observed there only when the error was large. And second, the distribution function of the node values should be “smooth”. This is illustrated in our model from Section 3 – Figure 12 shows F_{10} (x) and F_0 (x). The approximation is obviously perfect for the completely “smooth” F_{10} (x). For F_0 (x), it is still fairly accurate when the error is small (e_1), but it becomes inaccurate when the error increases (e_2). Should the game tree be larger, the distribution function high in the tree would be even less “smooth”, which suggests that the theorem (or at least its proof) may not be true for large game trees.

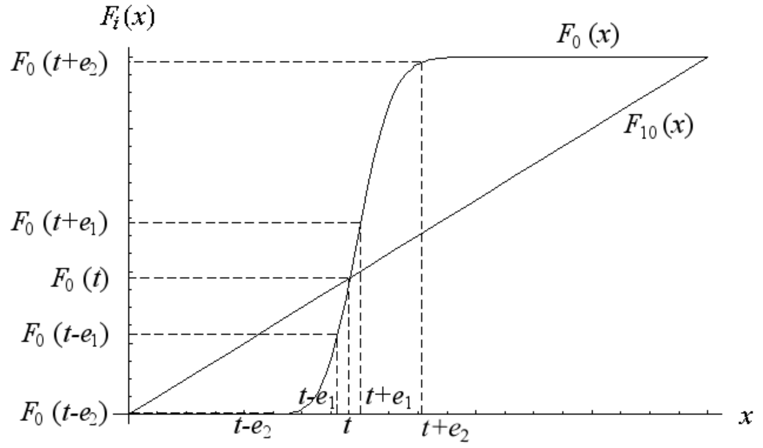

Figure 12. Approximation of $F_i\ (t)$ with $(F_i\ (t - e) + F_i\ (t + e))\ /\ 2$ in our model from Section 3.

Proof of Theorem 2. To compare P_m and P_n, both will be expressed as functions of the RV position error at level *i*. P_m is calculated using Equations (10) and (13), resulting in Equation (16).

$$\begin{aligned} P_\mathrm{m} &= 2VPE_{i-2} = F_{i-2}(t+e) - F_{i-2}(t-e) = \\ &= 4F_i(t+e)^2 - 4F_i(t+e)^3 + F_i(t+e)^4 - (4F_i(t-e)^2 - 4F_i(t-e)^3 + F_i(t-e)^4) \end{aligned} \tag{16}$$

P_n is most easily expressed in terms of Section 2 using Equations (3) and (4), resulting in Equation (17).

$$\begin{aligned} P_\mathrm{n} &= p_{i-2}k_{i-2} + q_{i-2}(1-k_{i-2}) = \\ &= (1-(1-p_{i-1})^2)k_{i-2} + \frac{2(1-k_{i-1})k_{i-1}q_{i-1}(1-p_{i-1}) + (1-k_{i-1})^2 q_{i-1}{}^2}{1-k_{i-2}}(1-k_{i-2}) \end{aligned} \tag{17}$$

The variables p_{i-1}, q_{i-1}, k_{i-2} and k_{i-1} appearing in Equation (17) can all be expressed as functions of p_i, q_i and k_i using Equations (5) and (6) for p_{i-1} and q_{i-1}, (1) for k_{i-2} and (2) for k_{i-1}, resulting in Equations (18). The full Equation (17) expressed with level-*i* variables is not reproduced here due to its length.

$$\begin{aligned} p_{i-1} &= \frac{2k_i(1-k_i)p_i(1-q_i) + k_i{}^2 p_i{}^2}{k_{i-1}} \\ q_{i-1} &= \frac{(1-k_i)^2(1-(1-q_i)^2)}{(1-k_{i-1})} \\ k_{i-2} &= k_{i-1}{}^2 \\ k_{i-1} &= 1 - (1 - k_i)^2 \end{aligned} \tag{18}$$

To express P_n in terms compatible with Equation (16), the variables p_i, q_i and k_i are written as $F_i\ (t) - F_i\ (t - e)$, $F_i\ (t + e) - F_i\ (t)$ and $F_i\ (t)$ respectively. Finally, we simplify the inequality by approximating $F_i\ (t) \approx (F_i\ (t - e) + F_i\ (t + e))\ /\ 2$. Inequality $P_\mathrm{m} > P_\mathrm{n}$ was solved analytically with Mathematica [29], a software package for symbolic (and numerical) computation. It turned out to be always true, which concludes the proof of Theorem 2.

6. Between real values and two values

In the previous sections, we were discussing only real-valued and two-valued minimax models. Since game-playing programs use a range of discrete values, neither is completely realistic, so this section will deal with a model that has multiple discrete values – a multivalued model. A real-valued model can be converted into a multivalued one by dividing the interval within which the real values lie in a number of subintervals and treating the values within each subinterval as one value. The number of subintervals will be termed the granularity of the model and denoted g.

To convert our real-valued model of Section 3 into a multivalued one, we first divided the interval within which the real values lie in subintervals of equal width. The border between two subintervals closest to the equilibrium point c_b was then shifted to c_b, as described in our previous work [17]. This ensured that the multivalued model with g = 2 is equivalent Beal's two-valued model of Section 2. When g is realistically large, i.e., several hundred or even thousand, the shift is hardly noticeable.

Since this section mostly deals with a multivalued minimax model, we will use the term multivalued (MV) error instead of RV error. When $g = \infty$, the MV position error and the MV move error are identical to their RV counterparts; when g = 2, the MV move error is identical to the 2V move error.

To see how the pathology is affected by the granularity, we conducted simulations with game trees generated according to the multivalued model. Figure 13 shows the root 2V position error with respect to the depth of search for different granularities. The static MV position error is constant for each granularity and set so that the static 2V position error is 0.1 (to achieve this, the standard deviation of the static RV position error σ ranges from 0.194 at g = 2 to 0.0157 at $g = \infty$); b = 2 and $d_{\max}$ = 10. We chose to display the 2V position error because this is the type of error which behaved pathologically in Beal's two-value model and as such it is best suited to explain the transition from two values to real values.

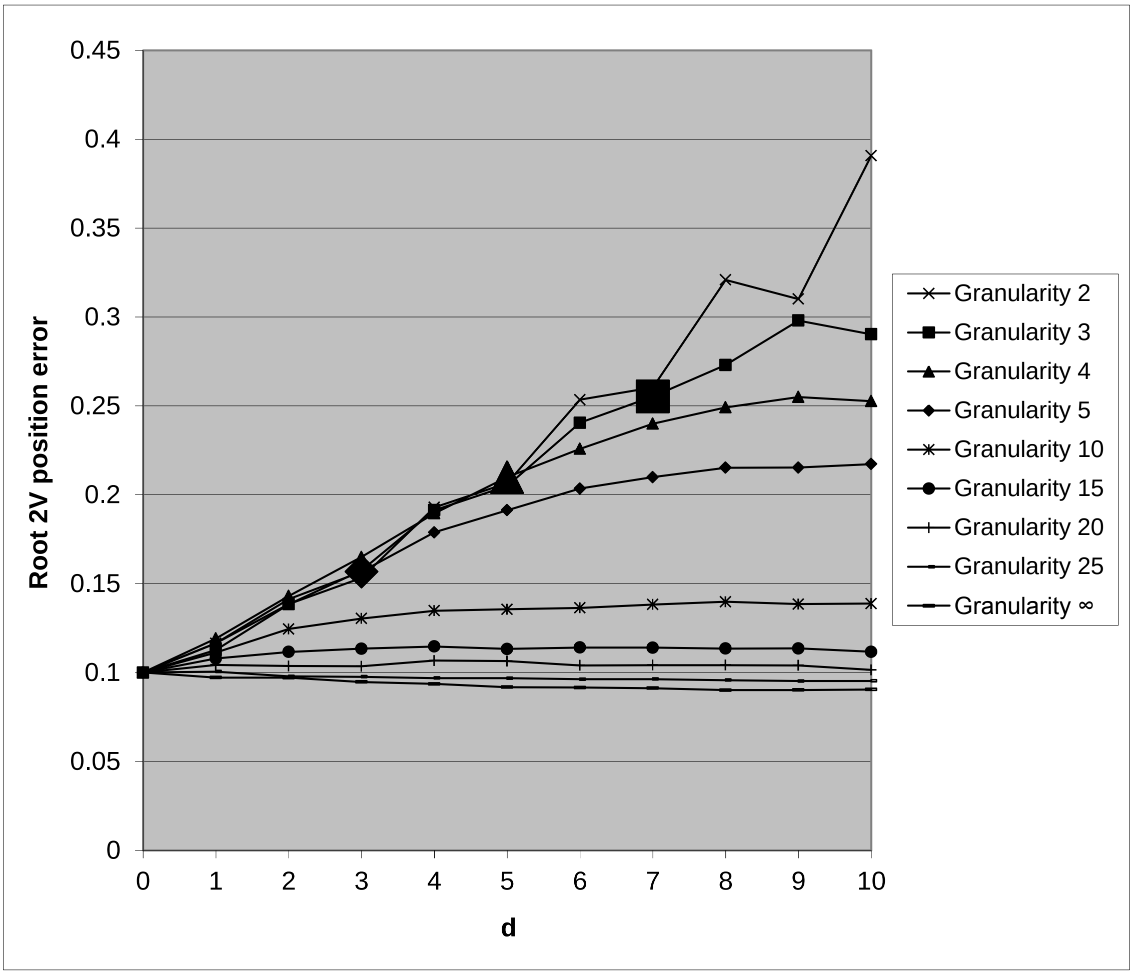

Figure 13. Root 2V position error with respect to the depth of search for different granularities.

As can be seen in Figure 13, the degree of pathology decreases with increasing granularity, until the pathology altogether disappears between g = 20 and g = 25 (where the curves start sloping downwards). The uppermost curve is identical to the curve depicting the root 2V position error in our binarized real-value model in Figure 8. The lowermost curve is similar to the curve depicting the error in our model without the binarization shown in Figure 3. The difference is that the benefit of minimax here is smaller, because the error is smaller ($\sigma = 0.0157$ as opposed to $\sigma = 0.1$ in Figure 3) and hence there is less room for improvement. The emphasized data points on the uppermost curve are those where the error for g = 3, 4 and 5 starts behaving markedly less pathologically than in the binarized real-value model. To understand the placement of these points and the transition from pathological two values to non-pathological real values, in general, one must examine the distribution of true node values throughout the game tree. Figure 14 shows density plots of the distributions of true values for g = 2, 3, 5 and 25; b = 2 and $d_{\max} = 10$. A darker color of an area indicates that a node is more likely to have the value represented by that area.

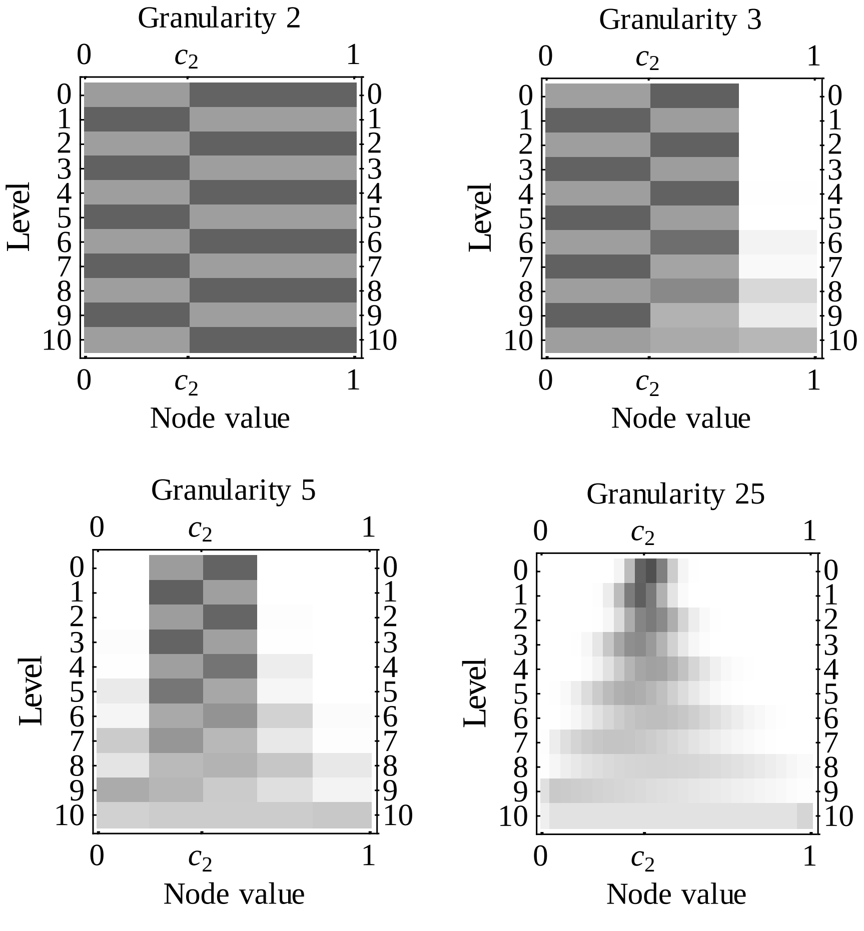

Figure 14. Density plots of the distributions of true node values for g = 2, 3, 5 and 25.

As can be seen in Figure 14, for the pathological g = 2, 3 and 5 only the node values adjacent to c_b are common at higher levels of the game tree. This is to be expected, since selecting the maximum from sets of values removes the lower values and selecting the minimum removes the higher values, leaving only values close to c_b. What this also means is that at higher levels, game trees with $g > 2$ become similar to game trees with g = 2, that is two-valued trees. Therefore, searching to smaller depths behaves similarly to searching two-valued game trees, i.e., pathologically. This explanation is confirmed by the fact that the emphasized data points in Figure 13 correspond to the lowest levels of the game tree where the node values not adjacent to c_b appear with a probability lower than 5%. Below those levels, other values are common enough that the pathology starts to diminish. For the non-pathological g = 25, however, multiple node values are present throughout the game tree.

Figure 13 shows that increasing the granularity changes the behavior of the root 2V position error with respect to the depth of search from pathological to non-pathological. Another aspect of this transition can be demonstrated on the static 2V position error. When g = 2, a constant static MV position error implies a constant static 2V position error. With real values, if the static MV position error is constant, the static 2V position error decreases with the depth of search, as shown in Figure 6. Figure 15 shows the static 2V position error with respect to the depth of search for different granularities. The static MV position error is constant for each granularity and set so that the static 2V position error is the same as in Figure 6 (σ ranges from 0.667 at g = 2 to 0.1 at $g = \infty$); b = 2 and $d_{\max}$ = 10.

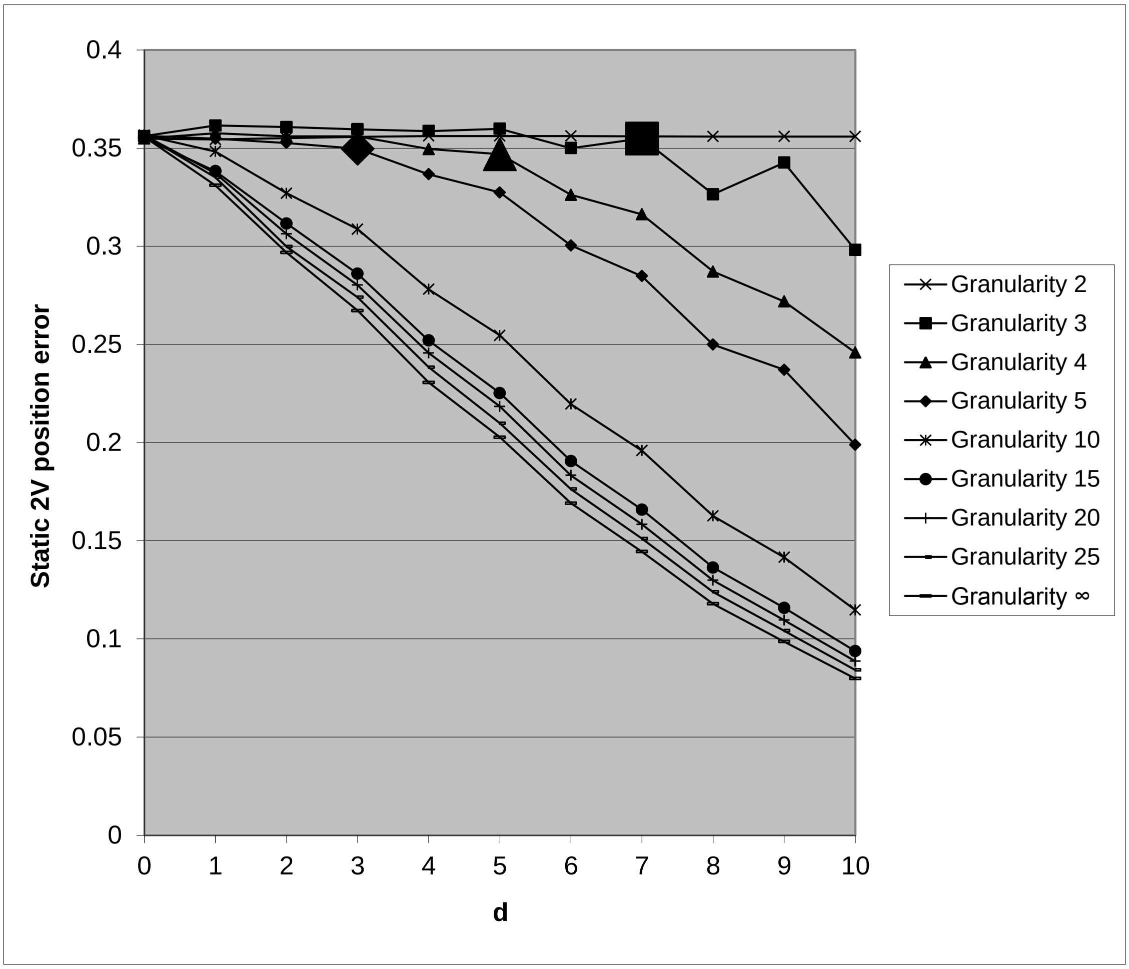

Figure 15. Static 2V position error with respect to the depth of search for different granularities.

The lowermost curve in Figure 15 is identical to the curve depicting the static 2V position error in our real-valued model shown in Figure 6: it decreases with the increasing depth of search, which prevents the pathology. Above it, each curve is closer to constant, ending with g = 2, which corresponds to the pathological Beal's model. The emphasized data points for g = 3, 4 and 5 have the same meaning as in Figure 13: they mark the lowest levels of the game tree where the node values not adjacent to c_b appear with a probability lower than 5%, and they also mark the level where the static 2V position error ceases to be approximately constant.

7. Discussion

To analyze the behavior of independent-valued minimax, 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 the heuristic evaluation was independent of the depth in the game tree. However, under these settings, the static 2V position error is not independent of the depth in the game tree, which is contrary to what was commonly assumed in early research on the pathology. Due to the minimax relation between the true values, both types of error cannot be independent of the depth simultaneously. The reason is that at larger depths, position values are on average farther away from the threshold separating the losses from the wins and therefore the same RV position error causes a smaller 2V position error, because the position value is less likely to cross over the threshold. We argued that the assumption of a constant static RV position error is better justified than that of a constant static 2V position error.

We analytically confirmed that in game trees with independent node values and a constant static RV position error, the static 2V position error at higher levels is larger than at lower levels. Moreover, we showed that it is large enough to eliminate the pathology. Our conclusion is valid, regardless of the distribution of true node values, as long as we are dealing with interesting positions, i.e., positions that are not clearly won for one side, and the static error is not excessively large.

We also examined the behavior of minimax when a limited number of discrete values are used instead of real or two values. It turned out that if the number of possible values is gradually reduced from an infinite number to two, minimax just as gradually transforms from beneficial to pathological. This happens because minimaxing removes the low and high values from the game tree and if the initial number of different values is too low, a few levels of minimaxing leaves only two.

Table 2 shows how the findings of this paper fit into the existing knowledge about minimax pathology with respect to the dependence of position values and the variation of static error. The space of the parameters describing the game tree and the heuristic evaluation function can be separated into the part where a deeper minimax search is beneficial and the part where minimax is pathological. The contribution of this paper can be described as extending the non-pathological part into the area of independent position values, which was previously thought to be pathological. The shaded area of the table represents the non-pathological part and the checkered pattern marks our extension. It should be noted that a weak position-value dependence or insufficient decrease of the static 2V position error with depth may cause the pathology to appear in the shaded area as well. The shading of each quadrant is justified with some representative publications (see Section 2 for a more extensive overview).

	Dependent position values	**Independent position values**
Static 2V position error decreases with depth	Scheucher & Kaindl's [10] complete model, *non-pathological*	According to Scheucher & Kaindl [10]: Pearl's basic model, *pathological* **Our model, *non-pathological***
Static 2V position error constant	Beal's [6] and Bratko & Gams's [5] dependent models, *non-pathological*	Beal's basic model, *pathological*

Table 2. The effect of the dependence of positions and variation of error on the pathology; the shaded area is non-pathological, the checkered pattern represents our extension of the area.

Dependent position values, which belong to the left two quadrants of Table 2, are well known to prevent the pathology (see Section 2 for details). The pathology in the bottom right quadrant also seems undisputed. The upper right quadrant, however, has been considered pathological so far. Nau [7, 9] and Pearl [18] found a decrease in the static 2V error to be insufficient to prevent the pathology. Scheucher & Kaindl, who actually focused on the variation of static 2V error, claimed that the upper-right quadrant is covered by the pathological Pearl model [18]. We, however, showed for this quadrant that if the static RV position error is constant, the static 2V position error decreases with depth sufficiently so that the pathology does not occur. This was not possible with Pearl's model because it used only real heuristic values, whereas the concept of a RV position error also requires real true values. It should be noted, though, that independence according to

Scheucher & Kaindl has a somewhat different meaning than according to our definition. When considering (in)dependence, Scheucher & Kaindl were concerned with heuristic values. Static heuristic values need not be in the minimax relation, so independence to them meant both independence within each level of the game tree and between levels. According to our definition, the values must be independent only within levels. Static heuristic values between levels are dependent due to the underlying true values being in the minimax relation. However, regardless of the interpretation of independence, the upper-right quadrant has never been shown to be non-pathological before.

The factors known to affect the pathology that we did not study in this paper are the branching factor of the game tree, the proportion of losses in the game tree and the conspiracy numbers. We did some experiments with branching factors larger than b = 2, and their results were similar to those with b = 2. If independent-valued game trees do not have the proportion of losses equal to the equilibrium point c_b or 1 – c_b, they quickly become almost certainly won for one side and thus not interesting. We computed the conspiracy numbers (or the numbers of leaf-disjoint strategies) in two-valued game trees of depth 10 built according to our model. The heuristic values were set so that in one experiment the static 2V position error decreased with the depth of search (the static RV position error was constant with σ = 0.1), and in one experiment the static 2V position error was constant at 0.1. We measured the pathology separately for the trees with the conspiracy number 1 and > 1, since according to Lorenz et al. [22] the former are expected to be pathological and the latter non-pathological. The results are shown in Table 3. One can see that most of the trees we studied had the conspiracy number 1, which is one of the reasons why such trees are particularly prone to the pathology. When the static 2V position error decreased with the depth of search as per our model, even the trees with the conspiracy number 1 were non-pathological, although the benefit of deeper search was very small. Lorenz et al. did not consider variable static 2V error, but the small benefit of deeper search when the conspiracy number is 1 is in line with their findings. When the static 2V position error was constant as per Beal's basic model, not even the conspiracy number > 1 could prevent the pathology. This seems to disagree with the findings of Lorenz et al., but it should be noted that those findings are valid generally across all two-valued game trees, whereas Beal's game trees are only a subset of two-valued game trees – a particularly pathological one.

	Conspiracy number 1	**Conspiracy number > 1**
Percentage of trees	77.95 %	22.05 %
Static 2V position error decreases with depth	Our model, *non-pathological (barely)*	Our model, *non-pathological*
Static 2V position error constant	Arbitrary two-valued game trees by Lorenz et al. [22], *pathological* Beal's basic model, *pathological*	Arbitrary two-valued game trees by Lorenz et al. [22], *non-pathological* Beal's basic model, *pathological*

Table 3. The percentage of game trees with the conspiracy number 1 and > 1, and the effect of the conspiracy number and variation of error on the pathology.

In summary, we showed that minimax is not necessarily pathological even if the node values are independent within each level of the game tree, which is the setting that was until now considered most prone to the pathology. A depth-independent real-valued static error proved sufficient to ensure the benefit of deeper search.

Acknowledgement

This work was supported by the Slovenian Ministry of Higher Education, Science and Technology, and by the Slovenian Research Agency under the Research Programme P2-0209 Artificial Intelligence and Intelligent Systems. We would also like to thank Dana S. Nau and Aleksander Sadikov for helpful discussions.

References

[1] J. von Neumann, O. Morgenstern, Theory of games and economic behavior, Princeton University Press, Princeton, USA, 1944.

[2] M. Campbell, Deep Blue, Artificial Intelligence, 134 (2002) 57-83.
[3] D.F. Beal, An analysis of minimax, in: M.R.B. Clarke (Ed.) Advances in Computer Chess 2, Edinburgh University Press, Edinburgh, UK, 1980, pp. 103-109.
[4] D.S. Nau, Quality of decision versus depth of search on game trees, in, Duke University, 1979.
[5] I. Bratko, M. Gams, Error analysis of the minimax principle, in: M.R.B. Clarke (Ed.) Advances in Computer Chess 3, Pergamon Press, Oxford, UK, 1982, pp. 1-15.
[6] D.F. Beal, Benefits of minimax search, in: M.R.B. Clarke (Ed.) Advances in Computer Chess 3, Pergamon Press, Oxford, UK, 1982, pp. 17-24.
[7] D.S. Nau, An investigation of the causes of pathology in games, Artificial Intelligence, 19 (1982) 257-278.
[8] M. Luštrek, Minimax pathology and real-number minimax model, in: B. Zajc (Ed.) Thirteenth International Electrotechnical and Computer Science Conference, Slovenian Section IEEE, Portorož, Slovenia, 2004, pp. 137-140.
[9] D.S. Nau, Pathology on game trees revisited, and an alternative to minimaxing, Artificial Intelligence, 21 (1983) 221-244.
[10] A. Scheucher, H. Kaindl, Benefits of using multivalued functions for minimaxing, Artificial Intelligence, 99 (1998) 187-208.
[11] G.P. Michon, Recursive random games: A probabilistic model for perfect information games, in, University of California, Los Angeles, 1983.
[12] D.S. Nau, Decision quality as a function of search depth on game trees, Journal of the ACM, 30 (1983) 607-708.
[13] G. Schrüfer, Presence and absence of pathology on game trees, in: D.F. Beal (Ed.) Advances in Computer Chess 4, Pergamon Press, Oxford, UK, 1986, pp. 101-112.
[14] A. Sadikov, I. Bratko, I. Kononenko, Bias and pathology in minimax search, Theoretical Computer Science, 349 (2005) 268-281.
[15] A. Sadikov, Propagation of heuristic evaluation errors in game graphs, in: Faculty of Computer and Information Science, University of Ljubljana, 2005.
[16] M. Luštrek, M. Gams, I. Bratko, Is real-valued minimax pathological?, Artificial Intelligence, 170 (2006) 620-642.
[17] D.S. Nau, M. Luštrek, A. Parker, I. Bratko, M. Gams, When is it better not to look ahead?, Artificial Intelligence, 174 (2010) 1323-1338.
[18] J. Pearl, On the nature of pathology in game searching, Artificial Intelligence, 20 (1983) 427-453.
[19] A. Sadikov, I. Bratko, I. Kononenko, Search vs knowledge: Empirical study of minimax on KRK endgame, in: H.J. van den Herik, H. Iida, E. Heinz (Eds.) Advances in Computer Games: Many Games, Many Challenges, Kluwer Academic Publishers, Boston, USA, 2003, pp. 33-44.
[20] B. Abramson, An explanation of and cure for minimax pathology, in: L.N. Kanal, J.F. Lemmer (Eds.) Uncertainty in Artificial Intelligence, North-Holland, Amsterdam, The Netherlands, 1986, pp. 495-504.
[21] I. Althöfer, Root evaluation errors: How they arise and propagate, ICCA Journal, 11 (1988) 55-63.
[22] U. Lorenz, B. Monien, Error analysis in minimax trees, Theoretical Computer Science, 313 (2004) 485-498.
[23] U. Lorenz, A new Implementation of Error Analysis in Game Trees, ICGA Journal, 19 (2006) 55-64.
[24] B. Doerr, U. Lorenz, Error propagation in game trees, Mathematical Methods of Operations Research 64 (2006) 79-93.
[25] M. Luštrek, I. Bratko, M. Gams, Why minimax works: An alternative explanation, in: International Joint Conference on Artificial Intelligence (IJCAI), Edinburgh, UK, 2005, pp. 212-217.
[26] H. Kaindl, Minimaxing: Theory and practice, in: AI Magazine, 1988, pp. 69-76.
[27] J. Pearl, Asymptotic properties of minimax trees and game-tree searching procedures, Artificial Intelligence, 14 (1980) 113-138.
[28] D.S. Nau, The last player theorem, Artificial Intelligence, 18 (1982) 53-65.
[29] Wolfram Research, Mathematica, in.