THE REASON FOR THE BENEFITS OF MINIMAX SEARCH

Anton Scheucher
BergstraBe 24
8600 Bruck/Mur
Austria

Hermann Kaindl
Marxergasse 18/2/1
1030 Wien
Austria

ABSTRACT

Since there existed no convincing theoretical explanation for the usually observed benefits of *minimax search* in practice, we investigated two instances of a class of tree models which are based on the concept of *quiescence.* (This way the strict separation or *static* and *dynamic* aspects in practical programs is modeled.) We performed Monte Carlo simulations, enhanced by analytic results. The behaviour of these models in our studies generally corresponds quite well to observations in practice (especially that of the model based on the more restrictive definition of quiescence). Hence, we found empirical evidence for an earlier conjecture, and these results can serve as an important step towards understanding the reason for the benefits of minimax search.

1 INTRODUCTION

For a long time, there was universal agreement to use *minimax search* in programs for *two-person, perfect-information games.* In fact, this approach is very successful in games like chess, checkers, or kalah. Usually it shows a dramatic improvement in playing strength with increasing search depth. However, the discovery of *minimax pathology* [Nau 80] demonstrated that in certain game trees deeper minimax search can also have detrimental effects. In the meantime several models have been analysed which show pathological behaviour, and several models which do not (see [Kaindl 88] for a critical overview). Recently, [Althofer 89] proved for special game trees that under certain conditions not only minimax searches fail, but also *every* other algorithm. However, none of these models explains convincingly, why minimax search is this beneficial in practice (by showing dramatic improvements of the decision quality with increasing search depth, and having a convincing relationship of the assumptions and the observed phenomena with "reality").

From extensive experience with computer chess practice, [Kaindl 88] developed an illustrative model based on an abstract concept of *quiescence.* In the following we sketch this and a related model, describe the design of our experiments, and present the results of our simulation studies investigating the behaviour with increasing search depth.

2 BACKGROUND

In general, the move *decision* in "interesting" games is based on *bounded look-ahead* to *artificial terminal nodes.* Usually, the "worth" of such nodes can only be estimated *heuristically.* This is normally done by a *static evaluation function f(n),* assigning a single *point value*—usually ranging over an interval of integer values—to a node *n.* (In the following we assume that *fin)* represents this value from the viewpoint of the side on move in the evaluated position, which is represented by the node *n.)* These values are propagated towards the root of a search tree according to a *back-up rule,* usually via *minimaxing.*

Definition 2.1. A *minimax value* $MM_f(n)$ of a node n can be computed recursively as follows (in the *negamax* framework):

(1) If n is considered *terminal*: $MM_f(n) := f(n)$;
(2) else: $MM_f(n) := \max_i(-MM_f(n_i))$ for all successors n_i of n.

Usually, the primary interest is not the minimax value of the tree itself, but rather the move to be selected at the root (the given position). In accordance with this back-up rule, the choice is one of what may be several moves leading to the maximum (backed-up) value of the successors. Since the *depth* of such searches will be important for the purpose of this paper, we also define a special case of using this rule.

Definition 2.2. $MM_f^d(n)$ is the minimax value of node n resulting from exactly d applications of the recursion (2) in definition 2.1 in every branch of the search tree.

In fact, $MM_f^d(n)$ defines the minimax value from a *full-width search* of the tree below n to uniform depth d (according to the "type A strategy" described by [Shannon 50]). While this kind of look-ahead is still the usual paradigm in computer game-playing practice, f does not directly denote here the *static* evaluator itself, but a simple *dynamic quiescence evaluator* based on the static one. In this paper it is sufficient to consider f as some heuristic function which evaluates nodes with some error.

3 THE CONCEPT OF QUIESCENCE

These quiescence evaluators are *dynamic* in the sense that they themselves perform a *search*—a rather selective *quiescence search:* for instance in chess, most programs try capture moves, searching for (relatively) *quiescent* positions (where the side on move cannot profitably capture). The term "quiescent" was already coined by [Shannon 50], and explained using an example of capture moves from the domain of chess. Independently, also Turing derived the same concept from his considerations about chess programming—he called quiescent positions "dead" (see [Turing *et al.* 53]). Due to the difficulties in controlling more elaborate quiescence searches, even now most chess programs restrict themselves to consider capture (and sometimes also checking) moves there. More sophisticated domain-specific aspects as well as a general framework for controlling such a quiescence search can be found in [Kaindl 82a, b].

However abstracting from the issues of a specific game, what is the domain-independent concept of quiescence all about? Generally, we would not consider a position as quiescent if its value can be changed drastically by moves or move sequences. These values are derived from the static evaluation function and/or the back-up rule. (Capture moves in chess significantly change the static values assigned by the materially dominated evaluation functions.) Based on this, [Kaindl 88] defined an abstract model of quiescence (rewritten here more formally).

Definition 3.1. A node *k* is *n-ply-quiescent* iff/k) = $MM_f^n(k)$.

This definition is already tailored towards a simplified environment with two-valued functions. As indicated above, these functions usually return a wider range of values. Therefore in practice, strict equality would be achieved very seldom, and a relaxation, for example, in the sense of "small difference" would be more realistic. Moreover, it only relates the values of the nodes at a certain constant distance of a given node with its own value. While this has been considered sufficient for illustrative purposes, a more restrictive definition may provide a more realistic model, e.g. one requiring stability *up to* a certain depth.

Definition 3.2. A node *k* is *uo-to-n-ply-quiescent* iff *k* is *i-ply-quiescent* for all *i E({l..n\.*

4 TREE MODELS BASED ON QUIESCENCE

The question to be answered is, why deeper and deeper minimax searches usually achieve better and better results in practice. While this may seem intuitively "clear" at first glance when considering a concrete domain like chess, up to now a convincing explanation based on a general model has not been given. The key question for this can be stated using Ken Thompson's words:1) "What is it in the tree?"

We are rather convinced now that the central concept for such an explanation is *quiescence*. Based on it, the following criteria specify tree models with simplified properties, which should nevertheless capture what is important for strongly beneficial behaviour of minimax search. Hence the question is, whether searches in such trees using the minimax back-up rule show such behaviour.

Definition 4.1. A class *of game tree models* is specified by the following assumptions:

(1) The tree structure has a uniform branching degree 6.
(2) True values of nodes (TV) are either WIN or LOSS.
(3) True values have the game-theoretic relationship.
(4) Heuristic values (assigned by *f)* are either +1 (estimating WIN) or -1 (estimating LOSS).
(5) Probabilities of error *e+* and *e-* are defined as follows (*k* being a node):

$$e^+(k) = P(f(k) = +1 \mid \mathrm{TV}(k) = \mathrm{LOSS})$$
$$e^-(k) = P(f(k) = -1 \mid \mathrm{TV}(k) = \mathrm{WIN})$$

(6) For all nodes / that are *quiescent* and for all nodes *m* that are not *quiescent* the following conditions hold:

$$e^+(l) < e^+(m)$$
$$e^-(l) < e^-(m)$$
$$e^+(l) < e^-(m)$$
$$e^-(l) < e^+(m)$$

(7) The number of nodes that are *quiescent* is small compared to the number of nodes that are not *quiescent.*

Substituting *"n-ply-quiescent"* for *"quiescent"* here results in the model presented in [Kaindl 88], using '' *up-to-n-ply-quiescent'* yields a related model. Both of these model instances have been investigated to study also the influence of the respective definition of quiescence.

In order to make studying the behaviour of these models feasible at all, they are kept as simple as possible. Hence, the relationship of these assumptions to reality and their possible influence on the behaviour should be discussed. Criterion (1) assumes a uniform branching of the tree, which is normally not the case in practice. However, according to [Michon 83] this property as a special case of "non-inertness" even seems to further pathological behaviour (the contrary of what we are going to show). Criteria (2) and (4) assume only two possible values for the true as well as the heuristic values. While especially the second of these simplifications seems strong, for instance the studies of [Bratko & Gams 82] and [Pearl 83] show that they did not influence the behaviour of their models. Assumption (5) distinguishes two different probabilities, since the analyses of [Schrufer 86] and [Pearl 83] indicate that this may be important for the behaviour. We will describe below whether this is also the case with our models.

In summary, assumptions (1) to (5) are fairly standard. However, assumptions (6) and (7) are new and should model "what it is in the tree" (based on the concept of *quiescence).* Assumption (6) states that the errors of statically evaluating quiescent nodes are usually smaller than those of evaluating nonquiescent ones. (Why else are resources spent for quiescence search?) Assumption (7) models the current situation in domains like chess, checkers, or kalah, where at least up to now no static evaluation functions have been written that are sufficient to capture their dynamic aspects. If they were, the searches would not have to *discover* this much. (We have also investigated this case within our simulation studies.)

As the parameters of the models are important for the investigations, we will enumerate them explicitly. The shape of the game tree is completely specified by the uniform branching degree 6, but also the probability *p* = P(TV(k) = WIN) for all nodes *k* may be important (see e.g. [Pearl 83]). The remaining parameters characterize the properties of the static evaluation function *f:* The errors *e+* and *er* have to be considered twice, once for quiescent nodes (eq+, e_{q-}) and once for nonquiescent nodes (e_n+, e_n~). The number *n* of *n-ply -quiescent* or *up-to-n-ply-quiescent* is of special interest, since a relationship with the benefits of searching *n* plies deep has been conjectured by [Kaindl 88]. At last, also the probability *fraction — P(k* is *quiescent)* for all nodes *k* can be parametrized, determining the average fraction of the number of quiescent nodes to the total number of nodes.

Besides specifiying the model assumptions we also have to define how the behaviour of the models shall be measured. The question is whether the errors made by the evaluation function are increased or decreased by backing up the heuristic estimates through several levels of the tree via minimaxing. Hence we define the probability of the error made by the minimax value of a certain search depth in estimating the true value as follows, using the fact that in our models P(TV(fe) = WIN) + P(TV(*) = LOSS) = 1. (The quality of the move choice at the root is directly related to this error.)

1) private communication of the second author with Ken Thompson in San Francisco, October, 1984

Definition 4.2. For node k and search depth d
$$e^d(k) = P(((MM_f{}^d(k) = +1) \wedge (\mathrm{TV}(k) = \mathrm{LOSS})) \vee ((MM_f{}^d(k) = -1) \wedge (\mathrm{TV}(k) = \mathrm{WIN}))).$$

5 EXPERIMENT DESIGN

Since we have investigated the models performing sim ulations, we had to generate game trees according to the criteria of the models. Within these trees, minimax searches were performed.

5.1 Tree Generation

While these models have resisted formal analysis up to now, even programming the generation of game trees according to their criteria was nontrivial. Hence we will sketch the most important aspects here. (The reader interested in algorithmic details is referred to [Scheucher 89].) Generally, every stochastic event has been simulated by a call to a pseudo-random number generator, parametrized independently of the relative frequencies achieved earlier in the tree generation process.[2)]

Due to the complex relationships between the true and the heuristic values within whole subtrees it was rather clear that the trees should be generated top-down from the root. These relationships can be viewed as constraints which have to be satisfied within the trees. And it is more convenient to propagate such con straints down the tree and to assign values satisfying them, rather than just assigning values at the bottom which have to be retracted often (since the constraints cannot be satisfied with them). Similarly, it was rather clear that the trees had to be stored explicitly, as gener ating them "on the fly" while searching them seemed unrealistic due to the complex relationships.

Also it proved useful to assign the true and the heu ristic values "simultaneously". Predetermining either of them (e.g. the true values) all over the generated tree first, leads to an undesirable effect: Since there are severe constraints to be satisfied for the remaining values, the probabilities of error (e+ and e- for quiescent and for nonquiescent nodes) as parametrized may differ too much from the actually achieved relative frequencies. Hence, we decided to generate the tree recursively as follows.

First, for some node k and its children nodes k, true values are assigned according to the game-theoretic relationship and the "probability-to-win" parameter p. They are also labelled as "quiescent" or "nonquiescent" according to the parametrized fraction (of the former). Then these successors are assigned heuristic values ac cording to the probabilities of error. (In fact, some of them already may have had one attached, which is not altered here.) After that, for each of the subtrees rooted in the nodes k_t the generating program tries to assign heuristic values to certain descendant nodes so as to satisfy the respective criteria, depending on the label of node k_t ("quiescent" or "nonquiescent") and the used definition of quiescence (3.1 or 3.2). This process itself works recursively through these subtrees, ensuring the reauired minimax values: Either one branch has to be followed (selected at random), or all the successors have to result in a predetermined value. Of course, it is easier to satisfy the constraints imposed by the defini tion of n-ply-quiescence than those by up-to-n-ply-quiescence. For the latter, it may happen that the process of assigning heuristic values deeper in the tree has to backtrack, since the constraints could not be satisfied. In such a case other choices are tried first at those nodes where only one branch has to result in a predetermined value.

5.2 Simulation Runs

After the generation of such a game tree, minimax searches were performed within this tree. (Of course, backward pruning was used to avoid unnecessary effort; see for instance [Knuth & Moore 75] for a de scription of the a-B procedure.) These searches computed minimax values MMf^d for all the possible depths d in such a tree. (f is characterized in such a simulated tree search by the corresponding parameters.) The goal of performing these searches was to gather data about the behaviour of the error made by the minimax value in estimating the true value, depending on the search depth. These data provided relative frequencies for e^d. In gathering them, we kept records whether the root nodes of the searches were labelled "quiescent" or "non-quiescent" since we expected different behaviour of the respective errors.

In order to study the influence of the parameters of the models, we varied them in certain ways (which are described together with the results). We investigated mainly parameter constellations fulfilling the assumptions of the models. However, "near misses"—in the sense that just one assumption is not fulfilled—seemed also to be of interest in order to see whether the respective assumption is necessary for the models to show the expected behaviour.

In addition to the parameters of the models them selves there are also parameters for the investigations via simulation. The size of a generated tree is deter mined here not only by the uniform branching degree 6, but also by its depth d_g. Since there was a maximum of available working memory determined by the given resources, pairs of b and d_g were chosen so as to result in trees just fitting into the available memory. While of course searches not only from the root of the generated tree were possible, none of them were directed into the region below (d_g — n) levels from this root, (n is the parameter in definition 3.1 or 3.2, respectively.) The reason is that within this region the conditions of the models cannot be guaranteed, since it contains nodes for which the property of n ply-quiescence (or up-to-n-ply-quiescence) cannot be ensured (because correlated nodes would be outside of the generated tree).

Another parameter for a simulation run is the seed for the pseudo-random number generator.[3)] For gathering data about several trees with an otherwise identical parameter constellation, this seed was varied. The simulations reported in this paper have been run over a period of about two months on a workstation, mainly overnight or on weekends.

6 RESULTS

Since the given parameter values (especially for proba bilities) could not always be achieved accurately within the generated trees (also due to the constraints to be satisfied), the sampling of the data was done according to the actual values in these trees (especially relative frequencies). For this reason the figures shown below plot the results of using certain parameter values out of an interval. We will discuss the results of parameter

2) The ones used are named nrand48 and erand48, and they are available under UNIX* System V.
UNIX is a registered trademark of AT&T.

3) In fact, the seeds for the "instances" of erand 48 were results of calls to nrand48, which was initialized by a single seed.

constellations "inside" and "outside" of the models separately, while always pointing to the peculiarities arising from the difference between *n-ply-quiescence* and *up-to-n-ply-quiescence.*

As these models were developed primarily from observations of minimaxing practice, we will emphasize the discussion of the relationship of phenomena shown by these models with "reality". Hence, we will restrict the presentation of the results to that of search depths $d \leqq n$, since for $d > n$ mainly unrealistic phenomena have shown up. (For a discussion of these, the interested reader is referred to [Scheucher 89].) Somehow it seems, as if there is too little to discover for the searches probing deeper than n within the trees according to our models. We might imagine that games like chess are characterized by a "large" n. Just before the presentation of the results gained from the simulations, let us briefly show straightforward analytic results about our models.

Lemma 6.1. Let $e^+(k)$ and $e^-(k)$ be error probabilities (according to definition 4.1), and $p = P(\mathrm{TV}(k) = \mathrm{WIN})$ for node k. Then the probability of the total (static) error can be computed as $e(k) = e^+(k) \cdot (1-p) + e^-(k) \cdot p$.

Proof: straightforward by using the multiplication theorem of probability theory and the fact that in our models $P(\mathrm{TV}(k) = \mathrm{WIN}) + P(\mathrm{TV}(k) = \mathrm{LOSS}) = 1$ ∎

Corollary. When restricting the scope to *quiescent* nodes l and using the corresponding error probabilities $e_q{}^+(l)$ and $e_q{}^-(l)$, the respective error probability $e_q(l)$ can be computed analogously.

Theorem 6.1. *Full-width searches* to depth $d = n$ using minimaxing and rooting in nodes l, that are *n-ply-quiescent*, result in the error probability

$$e_q{}^n(l) = e_q{}^+(l) \cdot (1-p) + e_q{}^-(l) \cdot p.$$

Proof:

$$\begin{aligned} e_q{}^n(l) &= P(((MM_f{}^n(l) = +1) \wedge (\mathrm{TV}(l) = \mathrm{LOSS})) \vee \\ &\qquad ((MM_f{}^n(l) = -1) \wedge (\mathrm{TV}(l) = \mathrm{WIN}))) \\ &= P(((f(l) = +1) \wedge (\mathrm{TV}(l) = \mathrm{LOSS})) \vee \\ &\qquad ((f(l) = -1) \wedge (\mathrm{TV}(l) = \mathrm{WIN}))) \\ &= e_q(l) = e_q{}^+(l) \cdot (1-p) + e_q{}^-(l) \cdot p \end{aligned}$$ ∎

Theorem 6.2. For all depths $d \in \{1..n\}$, *full-width searches* using minimaxing and rooting in nodes l, that are *up-to-n-ply-quiescent*, result in the same error probability $e_q{}^d(l) = e_q{}^+(l) \cdot (1-p) + e_q{}^-(l) \cdot p$.

Proof: analogous to the proof of theorem 6.1, but using definition 3.2 instead of 3.1 ∎

Especially the result of the last theorem may seem quite unrealistic, in telling that the error probability of searches from certain nodes is independent of the search depth ($\leqq n$). However on the contrary, also in practice the searches starting at nodes which are quiescent do not pay off: Whether their heuristic estimate is good or not, the value remains rather stable even when searches are performed. (Usually the fraction of quiescent nodes is very low.) The exact result of this theorem exaggerates, of course, but no more than the assumption of equality in the model definitions of quiescence it is based upon. In the figures shown below, the data about searches starting at nodes, that are *up-to-n-ply-quiescent*, are computed using this theorem.

6.1 "Inside"

In general (for $d \leqq n$), both models usually show a strong error reduction with increasing d, corresponding quite well to that usually observed in practice. Fig. 1 shows a typical case for the model based on *up-to-n-ply-quiescence.* The length of the 99.9% confidence interval for the data of nonquiescent nodes at $d = 1$ is 2 X 0.002288, that at $d = 5$ is 2 X 0.006265.4) However, especially the model based on *n-ply-quiescence* also shows certain phenomena of increasing error. First we will discuss these phenomena, and then we will analyse the influence of the various parameters of the models.

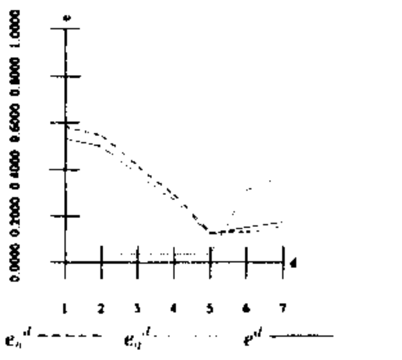
Let us compare the relative frequencies for e^1 and e^2 in the model based on *n-ply-quiescence.* With the exception of n = 2, the error rate generally increases here from the 1-ply to the 2-ply searches. Fig. 2 may illustrate this phenomenon by showing a typical case (comparable to that of Fig. 1). We cannot relate it to practice, and it is not really clear why it arises. But we can see from our data comparing both models, that its effect is related to the weaker stability provided by the definition of *n-ply-quiescence.* While also the model based on *up-to-n-ply-quiescence* shows different behaviour in this case, the error is usually reduced at d = 2 compared to that at d = 1.

More generally, the model based on *n-ply-quiescence* sometimes also leads to increases in error when going from *odd* to *even* search depths. Systematically, the behaviour here is worse than that when going from *even* to *odd* depths, while the overall tendency is always towards an error reduction. Fig. 3 illustrates such "oscillations" and the average behaviour (showing the straight line of least square distances). It is interesting to notice that minimaxing did also relatively better when the search depth was odd in the studies of [Nau *el al.* 86]. *Up-to-n-ply-quiescence* strongly dampens these oscillations (see e.g. Fig. 4). Since these phenomena are not known to have occured in practice, we attribute them to artificial assumptions in the models, such as keeping *n* constant for every quiescent node in a tree.

How does an increase of the parameter *n* influence the behaviour of the models? As to be expected, it reduces the overall rate of improvement per ply. But it also leads to stronger oscillations, especially for *n-ply-quiescence* (compare for instance Figs. 2 and 3). For varying values of 6, the same patterns can be observed, indicating no significant influence of this parameter on the general behaviour. However, different values of *p* influence the phenomena described above: When nodes *k* have a larger and larger probability P(TV(k) = WIN) (especially larger than 0.5), the strange phenomenon regarding 2-ply searches fades and even disappears. In much the same way the oscillations between odd and even depths are dampened by increasing values of p.

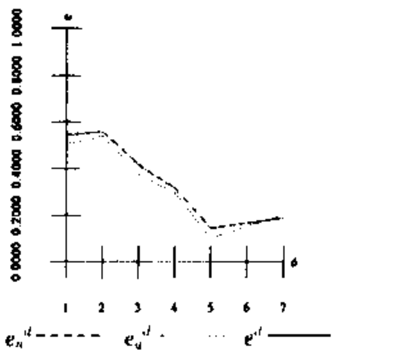
The influence of varying error probabilities for the static evaluation is of special interest here, since *increasing* errors at nonquiescent nodes (even any one of e_n* or e_n~) cause stronger improvement with increasing search depth, whereas low errors at such nodes do not even result in error reduction. There may be even a hint for the practical use of minimaxing, not to worry too much about estimating the values of nonquiescent nodes accurately. In fact, the most successful chess programs up to now do not use mechanisms (like swap-off analysis) which are attempting this for assigning static scores. The influence of these errors is stronger than that of the errors at quiescent nodes (especially it shows up, that any one of *en"* or e_n' should be large to achieve (strong) improvements).

4) Because of the way of tree generation and searching, we have more data for the shallower searches, and those of quiescent nodes are partially computed using the theorems Whenever feasible, the computed values of *e/* are plotted instead of relative frequencies The complete set of figures is available in [Scheucher 89|, showing a much stronger variation of parameters than is possible here due to space limitations.



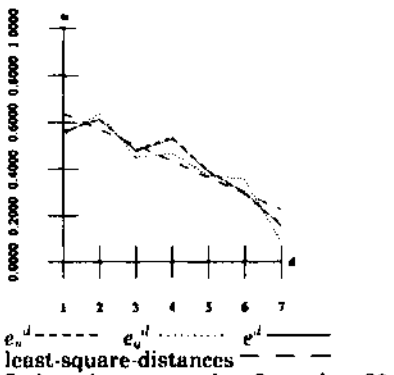
up to 5-ply-quiescent $b = 2$ $d_g = 20$
$e_q^+ \in [0.0, 0.2)$ $e_q^- \in [0.0, 0.2)$
$e_n^+ \in [0.8, 1.0)$ $e_n^- \in [0.8, 1.0)$
fraction $\in [0.0, 0.2)$ $p \in [0.4, 0.6)$

FIG. 1



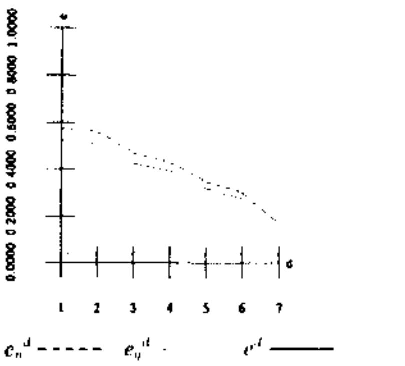
5-ply-quiescent $b = 2$ $d_g = 20$
$e_q^+ \in [0.0, 0.2)$ $e_q^- \in [0.0, 0.2)$
$e_n^+ \in [0.8, 1.0)$ $e_n^- \in [0.8, 1.0)$
fraction $\in [0.0, 0.2)$ $p \in [0.4, 0.6)$

FIG. 2



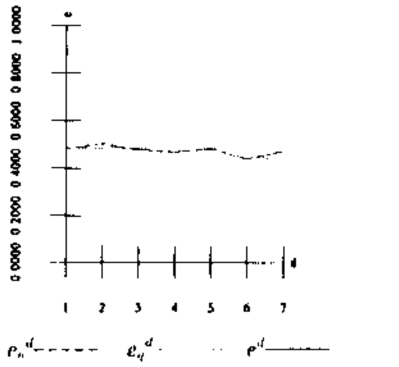
7-ply-quiescent $b = 2$ $d_g = 20$
$e_q^+ \in [0.0, 0.2)$ $e_q^- \in [0.0, 0.2)$
$e_n^+ \in [0.7, 0.9)$ $e_n^- \in [0.7, 0.9)$
fraction $\in [0.0, 0.2)$ $p \in [0.4, 0.6)$

FIG. 3



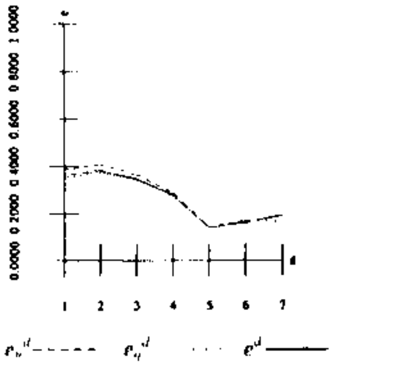
up-to-7-ply-quiescent $b = 2$ $d_g = 20$
$e_q^+ \in [0.0, 0.2)$ $e_q^- \in [0.0, 0.2)$
$e_n^+ \in [0.7, 0.9)$ $e_n^- \in [0.7, 0.9)$
fraction $\in [0.0, 0.2)$ $p \in [0.4, 0.6)$

FIG. 4



up to 5-ply quiescent $b = 2$ $d_g = 20$
$e_q^+ \in [0.4, 0.6)$ $e_q^- \in [0.4, 0.6)$
$e_n^+ \in [0.4, 0.6)$ $e_n^- \in [0.4, 0.6)$
fraction $\in [0.0, 0.2)$ $p \in [0.4, 0.6)$

FIG. 5



5 ply-quiescent $b = 2$ $d_g = 20$
$e_q^+ \in [0.0, 0.2)$ $e_q^- \in [0.0, 0.2)$
$e_n^+ \in [0.8, 1.0)$ $e_n^- \in [0.8, 1.0)$
fraction $\in [0.6, 1.0)$ $p \in [0.4, 0.6)$

FIG. 6

For different error probabilities of overestimating and underestimating (e^+ e~), we could not find differ ent results compared to the case of (nearly) equal probabilities. Especially in contrast to the result of [Schrufer 86], it was not necessary for e- to be negligible to avoid pathology.

6.2 "Outside"

Let us look at the effects of violating the assumptions (6) and (7) in definition 4.1, first one at a time. When some error probability at quiescent nodes becomes larger or equal to one of that at nonquiescent ones, the relative frequencies for e^d are nearly independent of the search depth d, i.e. no improvements can be found. (See Fig. 5, for instance.) According to theorem 6.2, for *up -to-n-ply-quiescence* the probabilities of the backed-up errors at quiescent nodes become large, and strangly enough even larger than those at nonquiescent nodes.

What happens, when the number of quiescent nodes is *not* "small" compared to that of nonquiescent ones? When the *fraction* of the number of quiescent nodes increases, the relative frequencies for e^d become already smaller at lower depths d, but they also cannot be reduced very much (see e.g. Fig. 6). This phenomenon can be related to rare cases in computer chess practice, in particular to certain chess endgames with very few possibilities for capturing pieces or promoting pawns.

Hence, from the results of our simulation studies we can see that both these new and critical assumptions are necessary. In addition, experiments of violating them simultaneously lead to completely unrealistic effects.

7 RELATED WORK

Now let us briefly compare our results with those of some of the earlier attempts to explain the benefits of minimax search observed in practice. Previous work suggests that independence of the true values or inde pendence of the errors made by the static evaluation function is a necessary condition for pathology in mini maxing. Note, that assuming independence instead of our critical assumptions (6) and (7), our models would be a generalization of a pathological one analysed by [Pearl 83]. Due to these assumptions (and our way of tree generation) we also avoid the "minimax conver gence" and the related necessity to have a specific value of the "probability-to-win" parameter p for "fair games" (see e.g. [Nau 821). Hence, most of the attempts to show nonpathological behaviour are based on various assumptions about "dependencies".

[Nau 82] investigated (via *N-games)* the influence of dependencies between the true values which arise "incrementally" so that the strength or weakness of a board position tends to be roughly the same for sibling nodes. But although this sibling correlation in Nau's simulation was considered extremely strong by [Pearl 83, p. 443], for instance Table 5 in [Nau 82] shows only modest improvements in decision quality with increasing search depth. These dependencies also cause the same static evaluation function to be clearly more ac curate for *N-games* than for the structurally equiv alent *P-games* (in which pathology occurs). This can be related to the more general result by [Chi & Nau 88] that low error rates favour minimaxing, especially versus *product propagation.* In order to achieve bene ficial behaviour in *our* models only a minority of nodes must be evaluated with low error—the quiescent ones. (This seems to be more realistic.)

Very similar to this "incremental" sibling relationship is the idea of "clustering" of true values investi gated by [Beal 82]. In order to make a comparison feasible, we gathered data about the "clustering factor" (as defined by [Beal 82]) in our models during the sim ulation runs. (In our models this factor cannot be con stant over all the levels of a search tree.) In the aver age, for the data at b = 2 a value of about 0.6 was found, at b = 4 about 0.4. Most interestingly, compared to the data plotted in the Appendix of [Beal 82] this would mean error reduction in the former and error increase in the latter case. However, we could neither find such an influence on the behaviour depending on b nor one depending on the data about the "clustering factor" in our models. Hence, while our assumptions obviously induce such clustering in our trees, its actual strength does not influence the behaviour critically.

In general, some of the earlier investigated dependencies are plausible to exist in real games (while their actual strength is not this clear). However, is the enor mous effort spent on searching deeper, *because* there are such dependencies? We think that the searches are performed deeper to *discover dynamic aspects* not incor porated in the usual *static* evaluation function (see also [Kaindl 88]). And this is modeled here based on the concept of *quiescence.* The concepts introduced by these models appear to be more fundamental, in having a very plausible relationship to "reality", and in indue ing certain dependencies, for instance those denoted as "clustering".

[Pearl 83] investigated the argument that deeper searching avoids traps, in a very simplified model con sidering only actual terminal nodes (that is, mates and stalemates) at all levels of the game tree as "traps". Based on practical data and experience, [Kaindl 88] argues that this type of trap alone does not seem to make a strong contribution to the benefits of minimax search in computer chess. In a higher sense, this "trap argument" actually served as a basis in developing the models investigated here. But they neglect Pearl's type of "trap" which easily can be evaluated *without* error. [Abramson 86] postulates instead of such "traps" (but with the same consequences) that the evaluation func tion must be able to recognize more and more forced wins (evaluated *without* error) as search deepens, to avoid pathology. Our models require neither of these strong assumptions to show beneficial behaviour of minimaxing—only a minority of nodes must be evalu ated with *low* error. (This distinction is insofar very important as e.g. the first model in [Pearl 83] can avoid pathological behaviour only when one of the errors (e^* or $e\sim$ in our notation) is equal to 0.)

8 CONCLUSION

In summary, we found empirical evidence for the con jecture stated in [Kaindl 88], that the benefits of minimax search usually observed in practice can be explained via the concept of *quiescence.* However, while the definition of *n-ply-quiescence* has been considered sufficient for illustrative purposes, the more restrictive definition of *up -to-n-ply-quiescence* resulted in a more realistic model. The most unrealistic issue here appears to be that n is *constant* all over the tree. Hence, using a random variable may be an interesting idea. Moreover, we investigated the (important) case "during" a game, ignoring completely the real end with actual terminal nodes. However, this can be modeled by introducing nodes which can be evaluated *without* error (like Pearl's "traps"). While our searches have been performed to uniform depth d, *quiescence searches* also might be simulated within our game tree models—after all they are based on *quiescence.*

ACKNOWLEDGMENTS

We would like to thank SIEMENS AG Osterreich for supporting this research, especially by providing machine time for the simulation studies and facilities for typing this paper. Wolfgang Ginzel's effort in writing software for plotting the figures is great ly acknowledged.

REFERENCES

Abramson, B., An Explanation of and Cure for Minimax Pathology, in *Uncertainty in Artificial Intelligence* (L. N. Kanal and J F. Lemmer, Eds.), Amsterdam North-Holland. 1986,495-504.

Althofer, I., Generalized Minimax Algorithms are no Better Error Correctors than Minimax Itself, in *Advances in Computer Chess 5 (D.F.* Beal, Ed), Amsterdam: North-Holland. 1989, 265-282.

Beal, D.F., Benefits of minimax search, in *Advances in Computer Chess 3* (M.R.B Clarke, Ed), Oxford: Pergamon Press, 1982, 17 24.

Bratko, I., and Gams, M , Error analysis of the minimax principle,in *Advances in Computer Chess 3* (M.R.B. Clarke, Ed.), Oxford: Pergamon Press, 1982, 1-15

Chi, P., and Nau, D.S , Comparison of the MINIMAX and PRODUCT Back Up Rules in a Variety of Games, in *Search in Artificial Intelligence* (L Kanal and V. Kumar, Eds), New York: Springer Verlag, 1988,450-471.

Kaindl, H., Quiescence Search in Computer Chess, in *SIGART Newsletter* 80 (special issue on game-playing (M A. Bramer, Ed.)), 1982a, 124-131, reprinted in *Computer Game -Playing: Theory and Practice,* Chichester, England Ellis Horwood, 1983, 39-52."

Kaindl, H., Dynamic Control of the Quiescence Search in Com puter Chess'in *Proc EMCSR-82,* Vienna, April, 1982b, 973 978. Amsterdam: North-Holland

Kaindl, H , Searching to Variable Depth in Computer Chess, in *Proc IJCAI-83,* Karlsruhe. August. 1983, 760-762

Kaindl, H , Minimaxing Theory and Practice, *AI Magazine* 9 (3), 1988,69-76.

Knuth, I) E , and Moore, R.W . An Analysis of Alpha-Beta Pruning, *Artificial Intelligence* 6(4) 1975, 293 326

Michon, G , *Recursive Random Games A Probabilistic Model for Perfect Information Games* Ph D Dissertation, University of California, Los Angeles, CA, 1983

Nau, D.S , Pathology on game trees a summary of results, in Proc *First National Conference on Artificial Intelligence,* Stanford, CA, 1980, 102-104

Nau, D.S., An investigation of the causes of pathology in games, *Artificial Intelligence* 19(3), 1982,257-278.

Nau, D.S , Purdom, P., and Tzeng, H.C., Experiments on alternatives to minimax, *International Journal of Parallel Programming* 15(2), 1986, 163-183

Pearl, J , On the nature of pathology in game searching, *Artificial'Intelligence* 20(4), 1983,427-453.

Scheucher, A., *Minimaxing und Quiescence: Eine experimentelle Untersuchung* Diplomarbeit, Technical Univ of Vienna, 1989.

Schrufer, G , Presence and Absence of Pathology on Game Trees, in *Advances in Computer Chess* 4 (D.F. Beal, Ed.), Oxford. Pergamon Press, 1986, 101 112

Shannon, C E , Programming a computer for plaving chess, *Phil. Mag* 41,1950,256-275.

Turing, A.M. et a/., Digital computers applied to games, in *Faster than Thought* (B.V Bowden, Ed), London: Pitman, 1953, 286-295