A Theory of the Heuristic Game Tree Search*

by

Chun-Hung Tzeng

and

Paul W. Purdom, Jr.

Computer Science Department

Indiana University

Bloomington, Indiana 47405

TECHNICAL REPORT NO. 135

A THEORY OF THE HEURISTIC GAME TREE SEARCH

Chun-Hung Tzeng and Paul Walton Purdom, Jr.

December 1982

*Research reported herein was supported in part by the National Science Foundation under grant number MCS 79-06110.

ABSTRACT

A theory of heuristic game tree search and evaluation functions is developed. We give an exact way for selecting the best initial move in a game using the information from a heuristic search under the assumption that both sides will play perfectly after that move. The result is quite different from the traditional minimax theory of game playing, and it leads to the product-propagation rules for backing up values when subpositions in the game are independent. The theory avoids Nau's paradox; deeper searching leads to better moves if one has reasonable searched informations.

Although this theory leads to the best possible move, it does not necessarily lead to the move where the program, with its limitation of imperfect play, will do best. A powerful game playing program should not assume that both sides will always make perfect moves, but should instead have a more realistic model of how it and its opponent will play.

1. Introduction

Standard game playing programs choose their moves by searching a large tree of potential continuations. Since the total game tree is huge, the tree is searched only to a limited depth. A static evaluation function is used to estimate value of the positions where the search stops. The static values are then combined using the minimax algorithm to estimate the value of each move.

Nau [3], however, proved that the minimax algorithm can degrade the information provided by the static evaluation function, so that for some types of games, it is better to search only one level of the game tree and use the values from the static evaluators directly to decide which move to make rather than to search several levels and then use the minimax algorithm before deciding which move to make. Pearl developed a simple game, suitable for theoretical study, for which this pathology of the minimax algorithm arises [4, 9]. Pearl's study [9] of the pathology suggests why the reason that the minimax algorithm is often useful. Although the minimax algorithm degrades the information from static evaluation, often a deeper search results in enough of an increase in the accuracy of the static evaluations to more than compensates for the degradation.

There are two heuristic arguments which are usually advanced in support of look-ahead. The first is the notion of *visibility* claiming that since the result of the game is more apparent near its end, nodes at deeper levels of the game tree will be more accurately evaluated and decisions based on such information should be more reliable. The second is that the back-up value integrates the features of all the nodes lying on the search frontier, and so should be more informed.

Since the minimax algorithm is not a perfect back-up process, it is interesting to find more accurate back-up procedures where the back-up values will really integrate the features of all the nodes lying on the search frontier. Pearl [9] suggested one should consider product-propagation rules instead of minimax. If p_i is the probability that the i-th son of a node is a win position, MIN node (to borrow terminology from the minimax algorithm) is $\prod_i p_i$ and the value of a MAX node is $1 - \prod_i (1 - p_i)$, where the product is over all the sons of the node. Nau [5] investigated this method experimentally and found that for Pearl's game it made winning moves more often than minimaxing did. Although the number of winning moves made by the product propagation method was greater than the number of winning moves made by minimax, this made almost no difference in the final outcomes of the games; each method won almost the same percentage of games. Nau [5] also studied a class of nonpathological games, where each method made approximately the same number of correct moves (and had the same winning percentage).

This paper is a study of the theory about heuristic game tree search and evaluation functions. For purposes of theoretical study, an abstract game model is introduced, which is a generalization of Pearl's game and Nau's game discussed by Nau [5]. We assume that result from the game is win or loss; draw is not permitted. The values (1 for a win of MAX and 0 for a win of MIN) at the leaves of the corresponding game tree are assigned according to a probability measure. The assumption on the search is that the visibility is improved in the sense that the information given by the search of a level can be retrieved from the information obtained by the search of a deeper level. These assumptions all apply to Pearl's game and Nau's game, and they do not strictly apply to most game playing programs.

A theory of a decision making using information from heuristic search is developed. There is, however, one major assumption. The minimax value of the next move is the only criterion used for determining the correctness of the decision; that is, the theory assumes that both sides will make perfect moves after the first move. Clearly no one plays this way. It does however illustrate the importance of considering the estimation of evaluation functions, and it leads to new back up process. It is the authors hope that this study will lead to both experimental and theoretical studies of how to best play games.

Three particular results are derived in this paper. First, the best way to estimate the value of a move by searching is to use the conditional probability of winning, given the information from the search; that is, this conditional probability integrates all the information from the search. Second, if the conditional probability is used to evaluate moves then the result from a deeper search is on the average strictly more accurate (except for one degenerate case with zero improvement). Third, if the positions on the search frontier are independent (as in Pearl's game), then the conditional probability estimate for the value of a move is obtained using the product-propagation rules suggested by Pearl [8, 9].

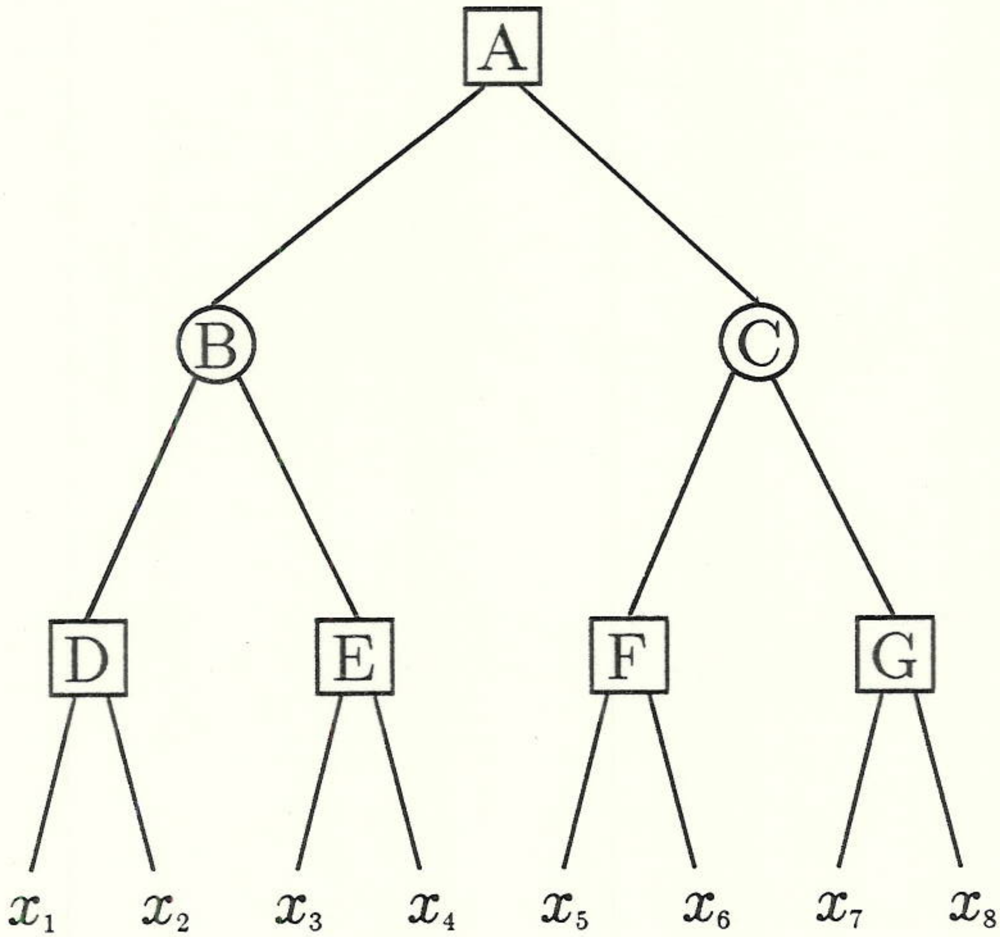

FIGURE 1. $\square$ is a MAX node and $\bigcirc$ is a MIN node.

2. An Example

An example on a simple game tree is discussed first, which shows how the ideas of the new models come from the usual probabilistic models. Consider an uniform binary game tree of 4 levels (Figure 1), where the terminal nodes may have independently the value 1 (win for MAX) or 0 (loss for MAX) with a probability of p $(0 < p < 1)$ and $1 - p$, respectively. Therefore, all leaf-patterns of this game tree form a probability space $\Omega = \{0, 1\}^8$, and the components of the point are independently identically distributed (i.i.d.). The minimax value M on the root A is then a random variable with value 1 or 0 on Ω. We also consider a heuristic search which finds the number of winning leaves under the searched node.

If the root A is searched, then we get a value k $(0 \le k \le 8)$, which is the sum of all leaf-values. We may associate this value with an event E_k:

$$E_k = \{(x_1, \ldots, x_8) \in \Omega \mid \sum_{i=1}^{8} x_i = k\}, \quad k = 0, 1, \ldots, 8.$$

The search determines thus the event E_k to which the searched leaf-pattern belongs; and E_k is the information given by the search at the first level. The whole space Ω is then divided into 9 different events, which form a partition of Ω.

If we search one level deeper, we have the number of 1's under the nodes B and C, two children of the root. Now the events associated with this level are the sets of the following form:

$$E_{ij} = \{(x_1, \ldots, x_8) \in \Omega \mid \sum_{k=1}^{4} x_k = i, \sum_{k=5}^{8} x_k = j\} .$$

Similarly, all such non-empty sets form a partition of Ω, which is finer than the previous one defined by the search at the root. Moreover, each E_k is a finite union of such sets :

$$E_k = \bigcup_{i+j=k} E_{ij} . \tag{2.1}$$

Therefore, each event E_{ij} at this level is contained in the event E_k $(k = i + j)$ corresponding to the first level; that is, the information given at this level is more accurate than that given at the first level.

The event given by the search at level 3 (nodes D, E, F and G are searched) is of the form

$$E_{i_1i_2i_3i_4} = \{(x_1, \ldots, x_8) \in \Omega \mid x_1 + x_2 = i_1, x_3 + x_4 = i_2, x_5 + x_6 = i_3, x_7 + x_8 = i_4\},$$

where $0 \leq i_1, i_2, i_3, i_4 \leq 2$. The minimax value at each node of this level can be determined by the values i_1, i_2, i_3, and i_4, hence, the minimax value at the root can be determined correctly by the search at this level. Moreover, the events $E_{i_1i_2i_3i_4}$ $(0 \leq i_1, i_2, i_3, i_4 \leq 2)$ form the finest partition of Ω:

$$E_{ij} = \bigcup_{i_1 + i_2 = i,\ i_3 + i_4 = j} E_{i_1i_2i_3i_4}. \tag{2.2}$$

Therefore, the events given by the heuristic search at different levels have the following relations:

$$E_{i_1i_2i_3i_4} \subset E_{ij} \subset E_k, \tag{2.3}$$

where $i_1 + i_2 = i$, $i_3 + i_4 = j$ and $i + j = k$. These relations show that our heuristic search has improved visibility.

Let's assume that the purpose of above heuristic search is to estimate the minimax value M at the root. Since M is a random variable on the space Ω, the most reasonable estimation depending on an event given by the search is the conditional probability of a win (minimax value = 1) given the event:

$$M_1 = P(M=1 \mid E_k) \qquad \text{at level 1;} \tag{2.4}$$

$$M_2 = P(M=1 \mid E_{ij}) \qquad \text{at level 2;} \tag{2.5}$$

$$M_3 = P(M=1 \mid E_{i_1i_2i_3i_4}) \quad \text{at level 3.} \tag{2.6}$$

It will be shown that there is no pathology in the decision making which depends on these values. As for the back-up process, it is just the product-propagation rules introduced by Pearl ([8, 9]).

3. Probabilistic Models

Consider games between two players MAX and MIN. Assume that the games always result in a win for one player or the other. There is a finite game tree associated with each game. Label the terminal nodes where MAX wins a one and the terminal nodes where MIN wins with a zero. We'll consider the games from MAX's point of view. A leaf of a game tree can be moved to a deeper level by adding nodes with only one child. Therefore, we assume that all game trees discussed in this paper have their leaves on a same level. Suppose that a game tree T has k leaves and h levels. Then all leaf-patterns $\omega = (x_1, \ldots, x_k)$, $x_i = 0$ or 1, form the space

$$\Omega = \{0, 1\}^k.$$

Assume that the values on the leaves are assigned according to a probability measure P on Ω w.r.t. the Borel field F generated by every single point in Ω. A Borel field (Chung [1]) is a nonempty collection of subsets of Ω, in which the operations of complement and countable union are closed. Then a probabilistic game tree model is defined as follows.

Definition 1. *A probabilistic game model for a game tree T with h levels and k leaves is a pair (Ω, P), where*

$$\Omega = \{0, 1\}^k,$$

and P is a probability measure on Ω w.r.t. the Borel field F generated by every point of Ω.

Both the Pearl's game and Nau's game discussed in Nau [5] are in probabilistic models. Furthermore, we don't assume the uniformity of the game trees. And on a probabilistic game model (Ω, P), the minimax value M at any node of the game tree T is a random variable on Ω:

$$M : \Omega \rightarrow \{0, 1\}.$$

We define a probabilistic search model as follows.

Definition 2. *Let (Ω, P) be a probabilistic game model for a game tree with h levels and k leaves. Then a search S on this model is probabilistic if S consists of an increasing sequence of Borel fields:*

$$F_0 \subset F_1 \subset F_2 \subset \cdots \subset F_h = F, \tag{3.1}$$

where $F_0 = \{\phi, \Omega\}$ and each F_i $(1 \leq i \leq h)$ is generated by a partition of Ω. For each leaf-pattern $\omega \in \Omega$, the search S at level i determines the event E of the partition generating F_i, which contains the leaf-pattern ω:

$$\omega \in E \in F_i , \quad i = 1, \ldots, h.$$

The search of level 0 returns the whole space Ω for every leaf-pattern, that is, no special information is given. The increasing sequence (3.1) illustrates the improved visibility of the search. In our example, the heuristic search giving the number of winning leaves under the searched node is a probabilistic search, where F_1 is generated by all events E_k's, F_2 by all E_{ij}'s, and all $E_{i_1 i_2 i_3 i_4}$'s generate F_3. For a general Pearl's game, where the corresponding game tree is an uniform tree (with height h and branching factor d) and 1 (or 0) is independently assigned to each leaf with a probability $p > 0$ (or $1 - p > 0$), the heuristic evaluation function giving the number of 1's on the leaves under the searched node is similarly probabilistic. The information from the search at a given level is contained in the information from a deeper level.

4. Probabilistic Evaluation Functions

Let (Ω, P) be a probabilistic game model for a game tree T with a probabilistic search, where

$$F_0 \subset F_1 \subset \cdots \subset F_h = F \tag{4.1}$$

is the corresponding sequence of Borel fields. The purpose of the search is to estimate the minimax values of some nodes of the game tree. Let M be a minimax value to be estimated. Then we define the evaluation functions for M as follows.

Definition 3. *For each i, $0 \leq i \leq h$, the conditional expectation of the minimax value M w.r.t. F_i*

$$M_i = E(M \mid F_i)$$

is called the i-th evaluation function of M. M_i is also called the i-th estimation of M.

We note that this definition could be applied to any random variable on Ω. If the search finds the event E at level i, then the value of M_i is just the conditional probability of a win given the event:

$$M_i(\omega) = P(M=1 \mid E) \quad \text{for all } \omega \in E.$$

Since Ω is a finite set, M_i assumes only finitely many values between 0 and 1. In particular, we have $M_0 = P(M=1 \mid \Omega) = E(M)$ and $M_h = M$, where $E(M)$ is the mean of the random variable M. M_0 does not depend on any information except the whole space. M_h depends on a complete information and is the minimax value M itself.

From the increasing sequence (4.1), we know that the sequence $\{M_i, F_i\}$ forms a martingale (Chung[1]):

$$M_i = E(M_j \mid F_i) \quad \text{for } 0 \le i < j \le h \,. \tag{4.2}$$

This means that M_i is the average of M_j if $i < j$. (4.2) can be illustrated in the case of our example as follows:

$$P(M=1 \mid E_k) = \sum_{i+j=k} P(E_{ij} \mid E_k) P(M=1 \mid E_{ij}) = \sum_{i_1+i_2+i_3+i_4=k} P(E_{i_1 i_2 i_3 i_4} \mid E_k) P(M=1 \mid E_{i_1 i_2 i_3 i_4}),$$

where M is the minimax value at the root A. The following theorem shows that the conditional probability of a win given by the value of the evaluation function will be improved when increasing the search level.

Theorem 1. *Let i and j $(0 \le i \le j \le h)$ be any two integers. For any two reals x and y $(0 \le x,\, y \le 1)$, we have*

$$P(M=1 \mid M_i = x,\, M_j = y) = y$$

if $P(M_i = x,\, M_j = y) \neq 0$. In particular, we have $P(M=1 \mid M_i = x) = x$ if $P(M_i = x) \neq 0$.

Proof. Consider the set $E = \{\omega \in \Omega \mid M_i(\omega) = x,\, M_j(\omega) = y\}$. Note that $F_i \subset F_j$, so both M_i and M_j are measurable w.r.t. F_j; hence E is an element of F_j. Therefore,

$$\begin{aligned} P(M=1,\, M_i = x,\, M_j = y) &= \int_E M\,dP = \int_E E(M \mid F_j)\,dP \\ &= \int_E M_j\,dP = y \int_E dP = yP(M_i = x,\, M_j = y), \end{aligned}$$

i.e.,

$$P(M=1 \mid M_i=x,\ M_j=y) = y$$

if $P(M_i=x,\ M_j=y) \neq 0$. Q.E.D.

Therefore, if the events of two levels are given, then the conditional probability of a win given these two events is just the estimation of the deeper level. And this estimation integrates the whole information on the search frontier.

5. Decision Making

Because of symmetry, we only consider the decision of MAX. Let MAX move from a node A, which has n different children $B_1, \ldots, B_n$. Let T be the current game tree with A as the root. It is still assumed that all leaves are on a same level of the tree.

Suppose that the game is in a probabilistic game model (Ω, P) for the tree T and there is a search with the increasing sequence of Borel fields:

$$F_0 \subset F_1 \subset \cdots \subset F_h = F,$$

where F_1 is relative to the search at the nodes B_i's and F_h is relative to the search at the leaves. Let $M^{(i)}$ $(1 \leq i \leq n)$ be the minimax value at the node B_i, which is a random variable on Ω. Now relative to the search of level j $(j \geq 1)$, there is an estimation of $M^{(i)}$ for each i, $1 \leq i \leq n$:

$$M_j^{(i)} = E(M^{(i)} \mid F_j),\ \ 1 \leq j \leq h.$$

For each i $(1 \leq i \leq n)$, $\{M_j^{(i)}, F_j\}$ $(1 \leq j \leq h)$ forms a martingale and Theorem 1 holds for each $M^{(i)}$; that is, the estimation of a deeper level is more precise. A move is said to be correct if and only if a node with minimax value 1 is chosen. Now it is ready to prove our main result:

Theorem 2. *The decision making depending on the search of a fixed level j, which chooses the node with the largest estimation $M_j^{(i)}$ $(1 \leq i \leq n)$, will be improved, if j is increased.*

Proof. Let j_1 and j_2 be integers such that $1 \leq j_1 < j_2 \leq h$. Suppose that for any fixed game in this model the node B_{k_1} is chosen relative to the search of level j_1 and B_{k_2} is chosen relative to the search of level j_2. Furthermore, for this fixed game let

$$M_{j_1}^{(k_1)} = x_1, \ M_{j_2}^{(k_1)} = x_2, \ M_{j_1}^{(k_2)} = y_1, \ M_{j_2}^{(k_2)} = y_2.$$

Then $x_1 \geq y_1$ and $x_2 \leq y_2$. From Theorem 1 we have

$$P(M^{(k_1)}{=}1 \mid M_{j_1}^{(k_1)}{=}x_1, \ M_{j_2}^{(k_1)}{=}x_2) = x_2$$

and

$$P(M^{(k_2)}{=}1 \mid M_{j_1}^{(k_2)}{=}y_1, \ M_{j_2}^{(k_2)}{=}y_2) = y_2.$$

This shows that, given the informations of levels j_1 and j_2, the conditional probability that the chosen node relative to level j_2 is a win node is always greater than or equal to the conditional probability that the chosen node relative to level j_1 is a win node. Being a sum of all such conditional probabilities, the probability that the chosen node relative to the level j_2 is a win node is therefore greater than or equal to the probability that the chosen node relative to the level j_1 is a win node. If $x_2 < y_2$ for at least one game in this model, then the decision making of the deeper level is strictly improved. Q.E.D.

This theorem gives us a theoretical support that if the search has improved visibility and the evaluation function integrates all information on the search frontier, then the decision making is improved when increasing the search level. Furthermore, note that the estimation $M_j^{(i)}$ $(1 \leq j \leq h)$ depends on the search of the whole tree T instead of the subtree T_i under the node B_i. Since we are considering the general probabilistic games, the information on any other subtree T_k $(1 \leq k \leq n$ and $k \neq i)$ with the node B_k as the root may affect the estimation of $M^{(i)}$. The Nau's games discussed by Nau [5] are of this type, because for such games the minimax values at nodes of a same level are in general dependent.

In the case of Pearl's games, the independence of the values at all leaves makes the estimations simpler, which will be discussed in the next section.

6. Independent Models and Product-Propagation Rules

First part of this section gives conditions under which the estimation $M_j^{(i)}$ at the node B_i in Theorem 2 depends on the subtree T_i under B_i only. Second part derives the product propagation rules as the appropriate backing up method in the case that all searched nodes are independent.

Now consider the same situation in the previous section. Let T_i $(1 \le i \le n)$ be the sub-game tree with the node B_i as the root. Suppose that the probability space (Ω, P) is a product of probability spaces (Ω_i, P_i), $1 \le i \le n$:

$$\Omega = \Omega_1 \times \cdots \times \Omega_n$$

and

$$P = P_i \times \cdots \times P_n,$$

where (Ω_i, P_i) is the probability space for the leaf-patterns under the sub-game tree T_i. Then the sub-game at the node B_i is in the probabilistic game model (Ω_i, P_i). In this case, we say that the game model (Ω, P) is a *product* of game models (Ω_i, P_i). Furthermore, the minimax value $M^{(i)}$ at the node B_i, which is a random variable on Ω, can also be regarded as a random variable on Ω_i. We use the same notation $M^{(i)}$ to denote both the random variable on Ω and the one on Ω_i, which have the same value. Thus for any $\omega = (\omega_1, \ldots, \omega_n) \in \Omega$, with $\omega_k \in \Omega_k$ $(1 \le k \le n)$, we have $M^{(i)}(\omega) = M^{(i)}(\omega_i)$. Moreover, the random variables $\{M^{(i)} \mid 1 \le i \le n\}$ are independent (Chung [1]).

For an integer j, $1 \le j \le h$, assume further that the Borel field F_j is a product of Borel fields

$$F_j = F_j^{(1)} \times \cdots \times F_j^{(n)},$$

where $F_j^{(i)}$ $(1 \le i \le n)$ is a Borel field generated by a partition of Ω_i. This means that the event given by the search of level j is a product of events in Ω_i w.r.t. the subtree T_i $(1 \le i \le n)$. Now let $E_1 \times \cdots \times E_n$ be a searched event at level j with E_i in $F_j^{(i)}$ for $1 \le i \le n$. Then the value of the estimation $M_j^{(i)}$ in the last section is

$$P(M^{(i)}=1 \mid E_1\times \cdots \times E_n) = \frac{P(E_1\times \cdots \times E_{i-1}\times(E_i,\, M^{(i)}=1)\times E_{i+1}\times \cdots \times E_n)}{P(E_1\times \cdots \times E_n)}$$

$$= \frac{P(E_i,\, M^{(i)}=1)}{P(E_i)} = P(M^{(i)}=1 \mid E_i),$$

thus

$$M_j^{(i)} = E(M^{(i)} \mid F_j^{(i)}),\ \ 1 \le i \le n.$$

Thus the estimation of the minimax value at node B_i only depends on the search of the subtree T_i. Therefore we proved the following theorem:

Theorem 3. *If the game model (Ω, P) for the tree T is a product of the game models (Ω_i, P_i) for subtree T_i $(1 \le i \le n)$ and the searched event is a product of the events w.r.t. the T_i, then the estimation of $M^{(i)}$ depends on the event w.r.t. the subtree T_i* only.

Since Pearl's games with the search finding the number of winning leaves under each searched node satisfy above conditions, it suffices to use the information under a subtree to estimate the minimax value at the root of the subtree, and the information outside the subtree won't affect the estimation.

In the second part, we consider the estimation of the minimax value at the root of a tree with the information under the tree. Let T be the game tree with root A and (Ω, P) the corresponding probabilistic game model. Given a probabilistic search model on this game tree with the increasing sequence of Borel fields:

$$F_0 \subset F_1 \subset \cdots \subset F_h = F\ ,$$

let the corresponding evaluation functions of the minimax value M at A be

$$M_i = E(M \mid F_i), \quad 0 \le i \le h.$$

For the search of the first level, where the node A is searched, let E be the searched event (in F_1), then the evaluation function M_1 gives the conditional probability of a win given the event E:

$$P(M=1 \mid E).$$

In our previous example, if the root of the game tree is searched, and the number of winning leaves is k ($0 \leq k \leq 8$), i.e., the searched event is E_k, then the evaluation function gives the conditional probability of a win given k winning leaves:

$$P(M=1 \mid \sum_{i=1}^{8} x_i = k).$$

When searching only one level in Pearl's games, it is not necessary to calculate this conditional probability because the number of winning leaves is also a suitable criterion. A node with more winning leaves never has a smaller probability of a win ([4]).

Now let $B_1, \ldots, B_d$ ($d \geq 1$) be the children of A, and T_{B_i} ($1 \leq i \leq d$) the subtree of T with B_i as root. Suppose that the model (Ω, P) is a product of models (Ω_{B_i}, P_{B_i}), where (Ω_{B_i}, P_{B_i}) is the probabilistic game model for the sub-game tree T_{B_i} ($1 \leq i \leq d$). If M_{B_i} is the minimax value at the node B_i, which may be regarded as random variables on both Ω and Ω_{B_i}, then the random variables $\{M_{B_i} \mid 1 \leq i \leq d\}$ are independent. Suppose that the Borel field F_2 is a product of Borel fields:

$$F_2 = \bigtimes_{i=1}^{d} F^{(i)},$$

where $F^{(i)}$ ($1 \leq i \leq d$) is a Borel field generated by a partition of Ω_{B_i}. This means that the events given by the search of level 2 (B_i's are searched) is a product of events of Ω_{B_i} w.r.t. the subtree T_{B_i} ($1 \leq i \leq d$). Now let

$$E_1 \times \cdots \times E_d$$

be a searched event at level 2 with E_i in $F^{(i)}$ for $1 \leq i \leq d$. If A is a MIN node then (by the product probability measure)

$$P(M=1, E_1 \times \cdots \times E_d) = P(M_{B_1}=1, E_1) \times \cdots \times P(M_{B_d}=1, E_d)$$

and

$$P(E_1 \times \cdots \times E_d) = P(E_1) \times \cdots \times P(E_d).$$

So it follows that the value of M_2 is

$$P(M{=}1 \mid E_1 \times \cdots \times E_d) = P(M_{B_1}{=}1 \mid E_1) \times \cdots \times P(M_{B_d}{=}1 \mid E_d). \tag{6.1}$$

If A is a MAX node then we have

$$P(M{=}0,\, E_1 \times \cdots \times E_d) = P(M_{B_1}{=}0,\, E_1) \times \cdots \times P(M_{B_d}{=}0,\, E_d)$$

and

$$P(M{=}1,\, E_1 \times \cdots \times E_d) = P(E_1) \times \cdots \times P(E_d) - P(M_{B_1}{=}0,\, E_1) \times \cdots \times P(M_{B_d}{=}0,\, E_d).$$

Therefore, the value of M_2 becomes

$$P(M{=}1 \mid E_1 \times \cdots \times E_d) = 1 - (1 - P(M_{B_1}{=}1 \mid E_1)) \times \cdots \times (1 - P(M_{B_d}{=}1 \mid E_d)). \tag{6.2}$$

Generally, let all nodes $C_1, \ldots, C_n$ of level l (≥ 2) be searched, T_{C_i} $(1 \leq i \leq n)$ the subtree with C_i as the root, and Ω_{C_i} $(1 \leq i \leq n)$ the space of all leaf-patterns of T_{C_i}. Similarly, suppose that (Ω, P) is a product of models (Ω_{C_i}, P_{C_i}) on each component. On each Ω_{C_i}, let M_{C_i} be the random variable representing the minimax value of the node C_i, which may also be regarded as a random variable on Ω. Similarly, the random variables $\{M_{C_i}\}$ are independent.

If $l > 2$ then the set $\{C_i \mid 1 \leq i \leq n\}$ is partitioned into d groups:

$$G_i = \{C_{g_{i-1}+1}, \ldots, C_{g_i}\}, \quad i = 1, \ldots, d,$$

where $1 = g_0 < g_1 < \cdots < g_d = n$ and each G_i $(1 \leq i \leq d)$ consists of all nodes of level l under the node B_i. Then we have

$$\Omega_{B_i} = \mathop{\times}_{j=g_{i-1}+1}^{g_i} \Omega_{C_j}.$$

Let

$$P_{B_i} = \mathop{\times}_{j=g_{i-1}+1}^{g_i} P_{C_j},$$

then

$$P = \mathop{\times}_{i=1}^{d} P_{B_i}.$$

Suppose further that the Borel field F_l is the product of Borel fields F_{C_i} on Ω_{C_i}, where each F_{C_i} is generated by a partition of Ω_{C_i} $(1 \leq i \leq n)$:

$$F_l = \mathop{\times}_{i=1}^{n} F_{C_i}.$$

Now let

$$E_1 \times \cdots \times E_n$$

be a searched event in F_l (under the tree T), where each E_i is a searched event in F_{C_i} (under the tree T_{C_i}) for $1 \leq i \leq n$. Then the value of l-th evaluation function is represented as follows:

Case 1. A is a MIN node: In this case we have

$$P(M=1, E_1 \times \cdots \times E_n) = \prod_{i=1}^{d} P(M_{B_i}=1, E_{g_{i-1}+1} \times \cdots \times E_{g_i})$$

and

$$P(E_1 \times \cdots \times E_n) = \prod_{i=1}^{d} P(E_{g_{i-1}+1} \times \cdots \times E_{g_i}).$$

Therefore, the value of M_l is

$$P(M=1 \mid E_1 \times \cdots \times E_n) = \prod_{i=1}^{d} P(M_{B_i}=1 \mid E_{g_{i-1}+1} \times \cdots \times E_{g_i}). \tag{6.3}$$

Case 2. A is a MAX node: Since

$$P(M=0, E_1 \times \cdots \times E_n) = \prod_{i=1}^{d} P(M_{B_i}=0, E_{g_{i-1}+1} \times \cdots \times E_{g_i})$$

$$= \prod_{i=1}^{d} (P(E_{g_{i-1}+1} \times \cdots \times E_{g_i}) - P(M_{B_i}=1, E_{g_{i-1}+1} \times \cdots \times E_{g_i})),$$

the value of M_l is

$$P(M=1 \mid E_1 \times \cdots \times E_n) = 1 - \prod_{i=1}^{d} (1 - P(M_{B_i}=1 \mid E_{g_{i-1}+1} \times \cdots \times E_{g_i})). \tag{6.4}$$

The rules (6.3) (containing (6.1)) and (6.4) (containing (6.2)) are called the *product-propagation* rules. Therefore, by induction the value M_l given the event $E_1 \times \cdots \times E_n$ is derived by this product-propagation rules from the conditional probabilities of a win at all searched nodes of level l:

$$P(M_{C_i}=1 \mid E_i) = P(M_{C_i}=1 \mid E_1 \times \cdots \times E_n), \quad 1 \le i \le n.$$

We summarize these results as follows.

Theorem 4. *Let the probabilistic game model* (Ω, P) *for a game tree* T *be the product of probabilistic models* (Ω_{C_i}, P_{C_i}) *for all subtrees* T_{C_i} $(1 \le i \le n)$ *which have nodes* C_i *of level* $l \ge 1$ *as roots. Suppose that the Borel field* F_l *of a probabilistic search model is the product of the Borel fields* F_{C_i} *(w.r.t. the subtree* T_{C_i} *), which are generated by partitions of* Ω_{C_i}*, respectively. Then the value of the evaluation function* M_l *of the minimax value* M *of the root is derived by the product-propagation rules from the values of*

$$E(M_{C_i} \mid F_{C_i}), \qquad 1 \le i \le n,$$

where M_{C_i} *is the minimax value at the node* C_i $(1 \le i \le n)$.

This theorem can be applied to Pearl's games, where all nodes of a same level are independent and the search finding the number of 1's (win for MAX) at leaves under each searched node has improved visibility in our sense. Let all nodes C_i $(1 \le i \le n)$ of a fixed level are searched, and the numbers of wins under C_i are b_i $(1 \le i \le n)$. Then starting from the conditional probabilities of a win at C_i given b_i:

$$p_i = P(M_{C_i}=1 \mid \text{number of wins under } C_i = b_i), \ (1 \le i \le n),$$

where M_{C_i} is the minimax value at the node C_i, we repeat the product-propagation process until we find the estimation at the root of the tree. Note that the value b_i's should be transformed to the conditional probabilities before the backing up process.

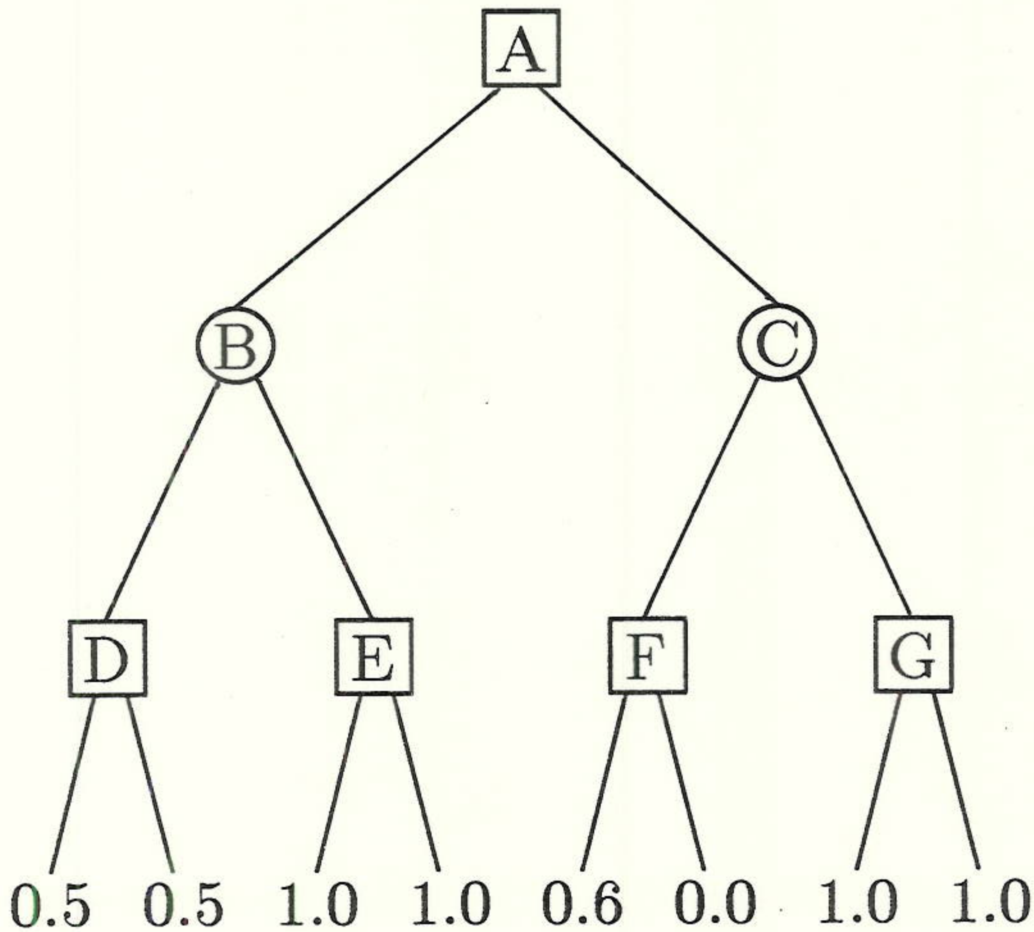

FIGURE 2.

7. Discussion

A careful study of Figure 2 will show the essential difference between the traditional minimax approach to game playing and the approach in this paper. The nodes at the search frontier have been labeled with the probability of winning. Assume that the positions below each node are independent so that the product propagation version of the theory in this paper applies. With the information at hand should one move to node A or to node B? The minimax theory says to move to node B, because after our opponent moves we will be in a position where we can choose between nodes with values 0.6 and 0.0. We will choose the 0.6 node and have a good chance of winning. If we had gone to node A, then on our next move we would have to choose between two nodes of value 0.5. Whichever one we choose, we only have an even chance of winning. The product propagation theory says to move to node A. If we move to node B, on the next move we will have to choose between nodes with probabilities 0.0 and 0.6. We will choose the 0.6 node and have 0.6 chance of winning. On the the other hand if we move to node A, then we will be able to choose between two nodes which independently have 0.5 chance of being winning nodes. If on our second move we can tell which node is best, then we are in great shape. There is a 0.75 probability that at least one of the nodes is a winning node.

From these informal arguments we can see that the minimax theory is assuming that we do not have any new information when we go to select our second move, while the theory in this paper assumes that we have become infinitely wise, and that on the second move we can suddenly determine (without error) what the best move is. This helps us understand the results of Nau's measurements [5]. He found that the product propagation program was indeed able to make winning moves significantly more often than a minimax program when the two programs were playing the same game with the same evaluation function. The advantage did not, however, accumulate. When the product propagation program played against the minimax program, there was no statistically significant difference in how often each program won. Although the product propagation program was making better moves, it was often arriving a positions where it was unable to take advantage of the winning position.

A real game playing program is faced with a situation intermediate between that where we would expect minimax to do particularly well and that where we would expect product propagation to do well. On the second move the program does have additional information, but it does not have perfect information. We hope that this paper will encourage additional theoretical work on alternatives to the minimax algorithm. A realistic theory of game playing needs to allow for mistakes by both the program and its opponent. Perfect play should just be a limiting case for a complete theory of game playing. A crucial aspect of an improved theory should be a model of how the evaluation of nodes vary with the depth of search.

People who are experimenting with game playing programs may wish to consider backing up evaluations with the following formula

$$v_0 = \exp(-((-\ln v_1)^k + (-\ln v_2)^k + \cdots)^{\frac{1}{k}}). \tag{7.1}$$

For $k = 1$, it reduces to

$$v_0 = v_1 v_2 \cdots,$$

i.e. the product propagation rule for a min node, while for $k = \infty$, it reduces to

$$v_0 = \min(v_1, v_2, \cdots),$$

i.e. the minimax rule for a min node. Thus as one adjusts k, one obtains propagation rules that are intermediate between minimax and product propagation.

The alpha-beta algorithm greatly reduces the amount of searching required when using the minimax algorithm. As more sophisticated propagation methods are developed, it is important to also develop methods to limit searching. The alpha-beta algorithm depends on the fact that some nodes have no effect on the final value. It is rare to have such nodes in a product propagation tree. Instead some nodes have a small effect on the final value while other nodes have a large effect. Any theory for value propagation that will be applied to deep searches will need a companion theory to determine which nodes are the most important to expand. If such a theory is able to make fine distinctions between the importance of expanding various nodes, it may be able to do better than minimax at controlling the search.

One difference between a conventional static evaluation function and our probabilistic evaluation function is that the static evaluation function evaluates each searched node separately and then a back-up process is involved, but the latter one evaluates the whole searched nodes altogether and no back-up process is involved, Nevertheless, they turn out to be the same if all searched nodes are independent (but a different back-up process is involved).

For the general dependent searched nodes, our evaluation functions exist so far only theoretically. The practical methods finding the values of evaluation functions should depend on the dependence of the searched nodes and should be studied for each individual case. The product-propagation method is only for the independent case, and some unexpected results like pathology in minimax process are also possible to arise if it is applied to the general games.

Even in the independent case, we still can not apply the product-propagation rule directly to any static evaluation function. We should transform the evaluated values to the conditional probabilities of a win before using the back-up rules. This conversion between two values is also a project in itself.

Acknowledgment

We wish to thank Professors Clyde Kruskal and Doug West for helping us come up with equation (7.1).

References

1. Chung, k. L., *A Course in Probability Theory.* New York, Academic Press 1974.

2. Knuth, D,F. and Moore, R.N., An Analysis of Alpha-Beta Pruning, *Artificial Intelligence* **6** (1975) 293-326.

3. Nau, D. S., Decision Quality as a Function of Search Depth of Game Tree. Technical Report TR-866, Computer Science Department, University of Maryland, 1980.

4. Nau, D. S., Pearl's Game is Pathological. Technical Report TR-999, Computer Science Department, University of Maryland, January 1981.

5. Nau, D. S., Pathology on Game Trees Revisited, and an Alternative to Minimaxing, to appear in *Artificial Intelligence.*

6. Nilsson, N., *Principles of Artificial Intelligence.* Palo Alto, CA. Tioga Publishing Company, 1980.

7. Pearl, J., Asymptotic Properties of Minimal trees and Game-Searching Procedures, *Artificial Intelligence* **14** (1980) 113-138.

8. Pearl, J., Heuristic Search Theory: Survey of Recent Results. UCLA-ENG-CSL-8017, June 1980, *Proc. 7th Int. Joint Conf. on AI,* August 1981.

9. Pearl, J., On the Nature of Pathology in Game Searching. UCLA-ENG-CSL-8217, January 1982.