NON-MINIMAX SEARCH STRATEGIES FOR USE AGAINST FALLIBLE OPPONENTS

Andrew L. Reibman
Bruce W. Ballard

Computer Science Dept.
Duke University
Durham, NC 27706

ABSTRACT

Most previous research on the use of search for minimax game playing has focused on improving search efficiency rather than on better utilizing available information. In a previous paper we developed models of imperfect opponent play based on a notion we call *playing strength*. In this paper, we use the insights acquired in our study of imperfect play and ideas expressed in papers by Slagle and Dixon, Ballard, Nau, and Pearl to develop alternatives to the conventional minimax strategy. We demonstrate that, in particular situations, against both perfect and imperfect opponents, our strategy yields an improvement comparable to or exceeding that provided by an additional ply of search.

I. INTRODUCTION

Any two-player, zero-sum, perfect information game can be represented as a minimax game tree, where the root of the tree denotes the initial game situation and the children of a node represent the results of the moves which could be made from that node. Most previous research on search for minimax game playing has focused on improving search efficiency. Results of this type improve the quality of player decision making by providing more relevant information. In contrast, our research focuses on better utilizing information rather than searching for more. In this paper, we summarize previous work on this issue, describe a new approach based on a model of opponent fallibility, and provide and discuss our results. In particular, we have devised a modification of the **-minimax search procedure* for tree containing *chance nodes* (Ballard [82,83]) to improve the overall performance of the minimax backup search algorithm. We shall demonstrate that, in particular situations, against perfect and imperfect opponents, our strategy yields an improvement comparable to or exceeding that provided by an additional ply of search. In the examples appearing below, we follow convention and call the two players "Max" and "Min" and use "+" to denote nodes where Max moves and "-" to represent similar nodes for Min. Positive endgame (leaf) values denote positive payoffs for Max. Readers unfamiliar with the conventional minimax backup search and decision procedure should refer to Nilsson [80].

This work has been supported by the Air Force Office of Scientific Research under grant AFOSR-81-0221.

II. PREVIOUS WORK ON PROBLEMS WITH MINIMAX

Given perfect play by our opponent, we know from game theory that a conventional minimax strategy which searches the entire game tree yields the highest possible payoff. However, most actual players, whether human or machine, lack the conditions needed to insure optimal play. In particular, because the trees of many games are very deep, and tree size grows exponentially with depth, a complete search of most real game trees is computationally intractable. In these instances, static evaluation functions and other heuristic techniques are employed to reduce the search used in making decisions. Before presenting our current work, we discuss previous efforts to deal with incomplete search and imperfect opponents.

A. Compensating for incomplete search

During the middle to late 1960's, James Slagle and his associates sought to improve the performance of minimax backup by attempting to predict the expected value of (D+1)-level minimax search with only a D-level search (Slagle and Dixon [70]). Their strategy was called the "M and N procedure" and determined the value of a Max node from its M best children and the value of a Min node from its N best children. The M and N procedure is based on the notion that *the expected backed-up value of a node is likely to differ from the expected backed-up value of its best child*. From empirical data, they defined a "bonus function" to be added to the static value of the best looking child of a node, hoping that this would lead to a better estimate of the true value of the parent. Using the game of "Kalah", they found that M and N yields an improvement in the expected outcome of the game about 13% as great as does an additional ply of search.

B. Modeling Imperfect Opponent Play

In Reibman and Ballard [83] we introduced general rules for constructing a model for an imperfect player based on a notion we call playing strength. Intuitively, playing strength is an indication of how well a player can be expected to do in actual competition, rather than against a theoretical perfect opponent. In our previous work, we also presented a model of an imperfect opponent based on a fixed probability of player error. The simulated imperfect Min player chose the best available move a fixed percentage of the time; otherwise Min chose another of the available moves. Thus the expected value of an imperfect opponent's "-" node was considered to be the value of its best child plus a fixed fraction of the value of any other children. Though it failed to consider the relative differences between the values of moves, this simple model was found to be better for use in our study than conventional minimax.

The reader may have noticed in the preceding section a resemblance between the notion of a bonus function and our attempt to accurately predict the expected value of moves made by a fallible opponent. In Ballard and Reibman [83b], we prove that in a simplified form of the model we present below, with a fixed probability of opponent error, our strategy can be obtained by an appropriate form of M and N (and vice versa), although the exact backed up values being determined will differ. This is because Slagle and Dixon's bonus function was approximately linear, while ours, based on the arc-sum tree model we use below, is a 4-th degree polynomial.

III. THE UNRESOLVED PROBLEM OF OPPONENT FALLIBILITY

In addition to having an inability to completely search actual game trees, actual implementations of minimax assume perfect play by their opponent. However, this assumption often is overly conservative and can be detrimental to good play. We now present two general classes of situations where minimax's perfect opponent assumption leads to sub-optimal play.

A. Forced Losses and Breaking Ties

The first problem with minimax that we consider is its inability to "break ties" between nodes which, though they have the same backed-up value, actually have different *expected* results. A readily observable example of this problem is found in forced loss situations. In the two-valued game in Figure 1, Max is faced with a forced loss. Regardless of the move Max makes at the "+" node, if Min plays correctly Max will always lose. Following the conventional minimax strategy, Max would play randomly, picking either subtree with equal frequency. Suppose, however, that there is a nonzero probability that Min will play incorrectly. For illustration, assume Min makes an incorrect move 10% of the time. Then if Max moves randomly, the expected outcome of the game is .5(0) + .5(.9*0 + .1*1) = .05. If Max knows that, on occasion, Min will move incorrectly, this knowledge can be used to improve the expected payoff from the game. Specifically, Max can regard each "-" node as a "chance node" similar to those that represent chance events such as dice rolls in non-minimax games. (Ballard [82,83] gives algorithms suited to this broader class of "*-minimax" games.) Thus Max evaluates "-" by computing a weighted average of its children, based on their conjectured probabilities of being chosen by Min, rather than by finding just the minimum. Following this strategy, Max converts the pure minimax tree of Figure 1 into the *-minimax tree also shown, and determines the values of the children of the root as 0 and 0.1. The rightmost branch of the game tree is selected because it now has the higher backed-up value. In terms of expected payoff, (which is computed as 0*(0) + 1.0*(.9*0 + .1*1) = 0.1), this is clearly an improvement over standard minimax play. Furthermore, this strategy is an improvement over minimax in forced loss situations regardless of the particular probability that Min will err.

```
              +                                +
backup  /       \ backup          backup   /       \    backup
value=0 /        \ value=0         value=0 /          \ value=.1
      -          -                        *            *
     / \        / \               p=.5 / \ p=.5  p=.1 / \ p=.9
     0 0        1 0                     0 0            1 0
```

Figure 1 : In an attempt to salvage a forced loss situation the minimax tree on the left is converted to the *-minimax tree on the right.

Our observed improvement in forced loss situations is a specific example of "tie-breaking", where the equal grandchild values happen to be zero. Because minimax uses only information provided by the extreme-valued children of a node, positions with different expected results often appear equivalent to minimax. Variant strategies can thus improve performance by breaking ties with information minimax obtains but does not use.

B. Exploiting Our Opponent's Potential For Error

By always assuming its opponent is a minimax player, minimax misses another class of opportunities to improve its expected performance, although less obvious than the forced loss situation presented above. An example is found in Figure 2. Assume as above that Min makes the correct move with probability .9. If Max uses the conventional backup strategy and chooses the left node, the expected outcome of the game is 2.1. If, however, we recognize our opponent's fallibility and convert the Min nodes to "*'s", (as in Figure 2), we must choose the right branch and the game's expected result increases to 2.9. Thus by altering the way we back up values to our opponent's nodes in the game tree, we can improve our expected performance against an imperfect opponent.

```
              +                                +
backup  /       \ backup          backup    /       \     backup
value=2 /        \ value=1         value=2.1 /          \ value=2.9
      -          -                        *              *
     / \        / \               p=.9 / \ p=.1   p=.9 / \ p=.1
     2 3        1 20                    2 3             1 20
```

Figure 2 : By converting the minimax tree on the left to the *-minimax tree on the right we may capitalize on our opponents potential for error.

In the example of a forced loss, the improvement in performance was due to the ability of a weighted average backup scheme to correctly choose between moves which appear equal to conventional minimax. In the second example, our variant backup yielded a "radical difference" from minimax, a choice of move which differed not because of "tie-breaking", but because differing backup strategies produced distinct choices of which available move is correct.

IV. A NEW MODEL FOR IMPERFECT PLAY

Having observed an opportunity to profit by exploiting errors which might be made by our opponent, we have formulated a more sophisticated model of an imperfect opponent than was previously considered. We will first provide the motivation for our enhancements and then describe the details of the imperfect player model used in the remainder of the paper.

A. Motivation for a Noise-based Model

In general, it should be fairly easy to differentiate between moves whose values differ greatly. However, if two moves have approximately the same value, it could be a more difficult task to choose between them. The strength of a player is, in part, his ability to choose the correct move from a range of alternatives. Playing strength can therefore correspond to a "range of discernment", the ability of a player to determine the relative quality of moves. An inability to distinguish between moves with radically different expected outcomes could have drastic consequences, while similar difficulties with moves of almost equal expected payoff should, on the average, have less effect on a player's overall performance.

We model players of various strengths by adding noise to the information they use for decision making. A player with noiseless move evaluation is a *perfect opponent*, while a player with an infinite amount of noise injected into its evaluation plays *randomly*. We introduce noise at the top of an imperfect player's search tree in an amount inversely proportional to the player's strength.

B. The Noise-based Model in Detail

We now describe the details of our imperfect player model. Each imperfect Min player is assigned a playing strength. In simulating actual games, the imperfect Min player conducts a conventional minimax backup search to approximate the actual value of each child of the current position. The backed-up values of each child are then normalized with respect to the range of possible backed-up values and a random number, chosen from the uniform distribution $0 <= x <= S$, (where S is inversely related to the player's strength), is added to the normalized value of each child. Thus the lower a player's strength the higher the average magnitude of the noise generated. The true node value with noise added is then treated as a conventional backed-up value. We add the noise to the top of a player's search tree because the actual effect of adding noise to the top of the tree can be studied analytically while the effect of introducing noise in the leaves is less well understood (Nau [80,82]). As described in Reibman and Ballard [83], we have verified that, in our noise-based model, ***decision quality degrades monotonically with respect to increases in the magnitude of the noise added*** to the conventional backed-up value.

V. A STRATEGY FOR USE AGAINST IMPERFECT OPPONENTS

We now present a strategy for use against imperfect opponents. We have based this strategy on the *-minimax search algorithms for trees containing chance nodes in order to compensate for the probabilistic behavior of fallible opponent. Three main assumptions are used as a foundation: (1) Against a Min player assumed to be perfect, we should use a conventional Max strategy. (2) Against an opponent who plays randomly, we should evaluate "-" nodes by taking an unweighted average of the values of their children. (3) In general, against imperfect players, we should evaluate "-" nodes by taking a weighted average of the values of their children, deriving the appropriate probabilities for computing this average from an estimate of our opponents playing strength.

In an attempt to predict the moves of our imperfect opponent, we assign our opponent a ***predicted strength***, denoted PS, between 0 and 1. To determine the value of "-" nodes directly below the root, our predictive strategy searches and backs up values to the "+" nodes directly below each "-" node using conventional minimax. A "-" node with branching factor Br is then evaluated by first sorting the values of its children in increasing order, then taking a weighted average using probabilities PS, (1-PS)*PS,...,(1-PS)**(Br-1) * PS. If PS=1, we consider only the minimum-valued child of a "-" node, in effect predicting that our opponent is perfect. At the other extreme, as PS approaches 0, a random opponent is predicted and, since the probabilities used to compute the weighted average become equal, the Min node is evaluated by an unweighted average of its children. How well our model predicts the moves of an imperfect opponent should be reflected in our strategy's actual performance against such a player.

VI. AN EMPIRICAL ANALYSIS OF THE PREDICTIVE STRATEGY

In Reibman and Ballard [83] we conducted an empirical analysis to investigate the correlation between playing strength as defined in our model and performance in actual competition. We now conduct an empirical study to compare the performance of our predictive algorithm with that of conventional minimax backup. We conduct our trials with complete n-ary game trees generated as functions of three parameters: D denotes the depth of the tree in ply, Br the branching factor, and V, the maximum allowable "arc value". In our study we assign values to the leaves of the game tree by growing the tree in a top-down fashion (Fuller, et al [73]). Every arc in the tree is independently assigned a random integer chosen from the uniform distribution between 0 and V. The value of each leaf is then the sum of arcs leading to it from the root.

The portion of our study presented here consists of several identical sets of 5000 randomly generated game trees with Br=4, D=5, and V=10. Against seven 2-ply Min opponents, ranging from pure minimax to almost random play, we pit conventional minimax players searching 1-, 2-, and 3-ply, and 10 predictive players, each with a 2-ply search and a PS chosen from between .1 and .9. The results of this experiment are found in Table 1. Before summarizing our observations, we note that the numbers in Table 1 represent points on a continuum; they indicate general trends but do not convey the entire spectrum of values which lie between the points we have considered.

In the first column of Table 1, we observe that, though it might be expected that pure Max backup would be the optimum strategy against conventional Min, several of our predictive players perform better

Table 1
Empirical Study Results
Trials=5000, Br=4, D=5, Game Values 0-50
Average payoff over all games

	Imperfect Player Noise Range						
Max's Strategy	0.00	0.25	0.50	0.75	1.00	2.00	6.00
1-ply minimax	27.23	30.60	32.34	33.13	33.62	34.14	34.58
2-ply minimax	28.15	31.29	32.90	33.46	33.90	34.47	34.76
3-ply minimax	28.98	32.05	33.36	33.96	34.31	34.65	35.01
2-ply PS = 0.9	28.21	31.40	33.03	33.62	34.03	34.58	34.91
2-ply PS = 0.8	28.21	31.40	33.03	33.62	34.03	34.58	34.92
2-ply PS = 0.7	28.21	31.40	33.02	33.62	34.04	34.58	34.92
2-ply PS = 0.6	28.21	31.40	33.02	33.62	34.05	34.60	34.95
2-ply PS = 0.5	28.20	31.41	33.05	33.66	34.10	34.66	35.00
2-ply PS = 0.4	28.20	31.42	33.10	33.70	34.15	34.72	35.07
2-ply PS = 0.3	28.17	31.41	33.11	33.75	34.19	34.99	35.12
2-ply PS = 0.2	28.13	31.40	33.13	33.77	34.22	34.83	35.16
2-ply PS = 0.1	28.08	31.39	33.14	33.79	34.24	34.85	35.19

than a conventional Max player searching the same number of ply. *The observed improvement is as much as 7% of the gain we would expect from adding an additional ply of search* to the conventional Max strategy. This result is analogous to that obtained with Slagle and Dixon's M and N strategy. Like M and N, our improvement is due, at least in part, to a strategy which, by considering information from more than one child of a node, partially compensates for a search which fails to reach the leaves.

In the central columns of Table 1, we see that against an opponent whose play is imperfect, *our strategy can provide almost half the expected improvement given by adding an additional ply of search* to the conventional Max strategy. We believe this gain is due primarily to the ability of our strategy to capitalize on our opponent's potential for errors.

If we examine the results in the last two columns of Table 1, we observe that, against a random player, *our strategy yields an improvement up to twice that yielded by an additional ply* of search. As the predicted strength of our opponent goes down, our predictions of our opponent's moves become more a simple average of the alternatives available to him than a minimax backup. We have previously conjectured that the most accurate prediction of the results of random play is such a weighted average and, as expected, our strategy's performance continues to improve dramatically as the predicted strength decreases.

We also observe a possible drawback to the indiscriminate use of our strategy. When we begin to overestimate our opponents fallibility, our performance degrades. In Column 1 of Table 1, our performance peaks. If we inaccurately overestimate the weakness of our opponent, our performance declines and eventually falls below that of minimax. We have observed similar declines in other columns as we let the predicted strength move even closer to 0 than the minimum predicted strengths shown in Table 1.

Having derived the results given above, we decided to tabulate the maximum improvement our strategy achieves over minimax. This summary is found in Table 2. We also give the results (in Table 2) of statistical confidence tests we have applied to our empirical analysis. These tests help to assess whether our strategy actually performed better than minimax. The percentages indicate the level of confidence that the improvements observed were not due to chance (given the actual normal distribution of our sample of trees). We note that in all but the first two columns our confidence levels are well over 90%.

VII. CONCLUSION

In this paper we have discussed the problem of adapting game playing strategies to deal with imperfect opponents. We first observed that, against a fallible adversary, the conventional minimax backup strategy does not always choose the move which yields the best expected payoff. To investigate ways of improving minimax, we formulated a general model of an imperfect adversary using the concept of "playing strength". We then proposed an alternative game playing strategy which capitalizes on its opponents potential for error. An empirical study was conducted to compare the performance of our strategy with that of minimax. Even against perfect opponents, our strategy showed a marginal improvement over minimax and, in some other cases, great increases in performance were observed.

We have presented some results of our efforts to develop variant minimax strategies that improve performance of game players in *actual competition*. Our present and future research includes a continued effort to expand and generalize our models of play, our predictive strategy, and the assessment of opponents using a playing strength measure. Further study of our models has included not only additional empirical experiments but also closed-form analysis of some closely related game tree search problems. We hope to eventually acquire a unified understanding of several

Table 2

Statistical Analysis of Empirical Study

	Imperfect Player Noise Range					
	0.00	0.25	0.50	0.75	1.00	2.00
Predictive play % % improvement over 2-ply minimax (in % of 1-ply)	7.2%	17.1%	52.1%	66.0%	82.9%	211.0%
Statistical Confidence: Is our optimum expected payoff better than that of 2-ply minimax?	58.2%	82.9%	98.2%	99.8%	99.8%	99.8%

distinct problems with minimax in order to develop a more general game playing procedure which retains the strong points of minimax while correcting its perceived inadequacies.

REFERENCES

[1] Ballard, B. A Search Procedure for Perfect Information Games of Chance: Its Formulation and Analysis. *Proceedings of AAAI-82*, August 1982.

[2] Ballard, B. The *-Minimax Search Procedure for Trees Containing Chance Nodes. *Artificial Intelligence*, to appear.

[3] Ballard, B. and A. Reibman, "What's Wrong with Minimax?". *1983 Conf. on Artificial Intelligence*, Oakland University Rochester, Michigan, April, 1983.

[4] Ballard, B. and A. Reibman, "Experience with Non-Minimax Search Criteria for Minimax Trees". Technical Report in preparation, to be submitted for publication, 1983.

[5] Fuller, S., Gaschnig, J. and J. Gillogly. An analysis of th alpha-beta pruning algorithm. Dept. of Computer Science Report, Carnegie-Mellon University, July 1973.

[6] Nau, D. Quality of Decision Versus Depth of Search in Game Trees: A summary of Results. *First Annual National Conference on Artificial Intelligence*, August 1980.

[7] Nau, D. Pathology on Game Trees Revisited and an Alternative to Minimaxing. Technical Report, Dept. of Computer Science, University of Maryland, Jan. 1982.

[8] Nilsson, N. *Principles of Artificial Intelligence*. Tioga, Palo Alto, 1980.

[9] Pearl, J. On the Nature of Pathology in Game Search. Technical Report UCLA-ENG-CSL-82-17, School of Engineering and Applied Science, University of California, Los Angeles, Jan. 1982.

[10] Reibman, A. and B. Ballard. Competition Against Fallible Opponents, *21st Southeast Region ACM Conference*, Durham, North Carolina, April 1983.

[11] Slagle, R. and J. Dixon. Experiments with the M & N Tree-Searching Program. *Communications of the ACM*, March 1970, 13 (3), 147-153 .