Cowles Foundation Paper
Reprinted from Bulletin of the Santa Fe Institute, 6(2), 1991

When Seeing Further is Not Seeing Better

By using games like chess as models for the process of making decisions in a complex world, we discover circumstances under which looking far ahead in a decision tree leads to poor choices. In particular we assert that the Shannon algorithm, which is the basis of most computer chess programs, is flawed because it does not take into account its own bounded rationality. We derive several meta-principles for decision making, and show that they explain some recent successful deviations from the Shannon algorithm. We conjecture that they will lead to other important modifications, or even a completely new algorithm.

A new machine has called into question our assumptions about calculation. Deep Thought, a chess-playing machine that can calculate more moves ahead than all other computer program, has taken first place in the World Computer Chess Championships, defeated a number of human grandmasters, and even challenged, albeit unsuccessfully, World Champion Gary Kasparov. Deep Thought's designers are working on an even more powerful microchip with which Deep Thought will be able to calculate an additional two full moves ahead, giving it enough strength (so they claim) to beat Kasparov with ease. Can we agree with Gary Kasparov, who still believes that brute calculation does not necessarily provide the best answers?

Because chess is a complex setting in which we cannot accurately quantify our ignorance, and in which the calculation of all possible consequences of an action is beyond the capabilities of any computer, even beyond those of any computer imaginable, we can use the game as a metaphor for studying decision making, learning, and expertise. At the Santa Fe Institute conference on Learning in Economics, Psychology, and Computer Science, held in January 1991, the game of chess was a frequent example in many of the discussions. As a consequence, the two of us began chatting about chess and computers, and John posed the following puzzle.

If computers could perfectly assess the value of a position in chess, they would not need to look ahead more than one move. On the other hand, if they are incapable of evaluating a position at all, it would do them no good to look ahead more than one move. Why is it then, that when their ability to evaluate positions lies somewhere between the two extremes, it seems to help them to look ahead as many moves as possible? One easy explanation is that positions become appreciably easier to evaluate as one goes deeper into the game. But can this be the whole story or even be right? After all, the typical foresight horizon of the strongest chess computer is only four or five moves, and who would argue that positions are on the average much simpler at move 15 in a game than they are at move 11? Of course, some positions are clarified by looking ahead, but at the same time, many simple positions have complicated successors further down the tree of moves. The more we talked, the more the puzzle grew, and we began to hunt for some general principles. The trail led us to some surprising discoveries, and a new project at the Santa Fe Institute emerged.

The question that we have raised applies to many situations besides chess, and indeed our project is not really about chess, but about decision making in complex environments. Imagine, for example, a traveler who needs to drive from downtown Manhattan to the Upper West Side of New York City. She is only vaguely familiar with the layout of the city, but she does have a detailed map. Using the map is difficult in traffic, and it is time-consuming to park somewhere for a closer look. Furthermore, the map is not completely clear about one-way streets, and says nothing at all about traffic jams, road construction, or stop lights. Should she try to trace out as much of the route on the map as can be reliably remembered, including all sorts of alternate routes to allow for unforeseen complications (i.e., look ahead as far as possible, keeping in mind that no one could retain a complete plan, allowing for all contingencies, in such a complex situation), or is it better to adopt a one-step look-ahead plan, such as always trying to head uptown and somewhat toward the west. Consider also a father facing a difficult decision about how to discipline a child. Should he try to calculate all the possible ways in which his child, and his other children, might react, and then how they subsequently might react to his reaction to their reaction? Or should he simply do what seems fair on general principles? One can easily find similar examples in a variety of other settings, from the worlds of economics and politics, to the affairs of the heart.

Are there rules, "meta-principles," prescribing when making choices based on general principles is superior to looking as far ahead as possible, and vice versa? If some calculating is to be done, are there meta-principles prescribing what to calculate? The Shannon algorithm for game play by computers, which we will describe below, is based on the belief that seeing further is seeing better, and so comes firmly down on the side of calculation at the deepest possible level in a decision tree. But when sight is cloudy at best (as is the case when a computer looks at any given chess position), is this belief justified? Since the Shannon algorithm does not make any allowance for its own imperfect vision, we think not, at least not all of the time. In fact, we have discovered several ways in which looking ahead can be misleading, as well as ways in which it can be helpful.

Of course, we are not the first to have noticed that looking far ahead is not always the best policy. But there appears to be very little systematic work on finding an alternative. Some people have tried to figure out why search depth has been so successful in practice, particularly for chess computer programs. But relative to the tremendous amount of effort put into hardware and software intended to help computers look ever deeper, there has been relatively little work directed at understanding the *failures* of search depth, and when such work has been done, the usual recommendation is to look even further ahead.

Information and information-processing considerations play an important role in economic theory, including capital theory, finance, macroeconomics, game theory, and labor theory. In all of these situations it is assumed that the agents are aware of their own limitations, and that they can quantify their ignorance. Across all of these decision problems, knowing more means doing better. Chess is a fascinating paradigm for decision making in complex environments precisely because, as in most real life situations, it is a setting in which we cannot accurately quantify our ignorance. We should therefore not be surprised if some natural decision-making routines do not have the property that they perform better when they are given more information. As we shall see, the Shannon algorithm is particularly vulnerable to this problem since it makes no attempt to compensate for its own fallibility.

A straightforward example in which learning more may be harmful is when there is "roundoff error," so that after sufficiently many calculations, the small errors inevitably made at each step accumulate and the final answer is almost certainly wrong. One sometimes sees this in chess, when a commentator on some game declares that a move is good (or bad), giving as proof a 15-move variation with "best" play by both sides. Although the evaluation of the final position in his variation may be unambiguous, and there may be no obvious blunder in the given line of play, the possibility that at each step there might have been a slightly better move leaves the conclusion still very much in doubt. (The Shannon algorithm searches exhaustively every possible move up to a certain depth, so it is not necessarily subject to this kind of error.)

In order to compensate for one's ignorance, it would seem necessary to quantify it, but how does one do that in a real-life setting like chess? In our project we propose to solve this dilemma by following a two-step plan: First we build a precise model in which the ignorance is quantified, and we derive mathematically the correct decision-making algorithm. Second, keeping in mind that such a model cannot be a faithful portrait of reality, we look for principles of decision making that are independent of the way in which we quantify ignorance.

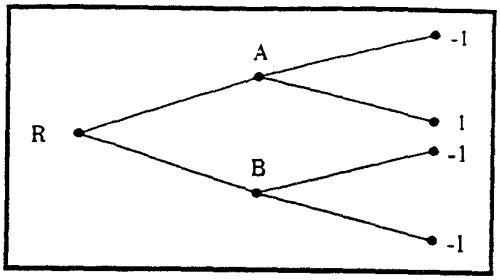

Figure 1: A two-stage decision tree.

A Simple Example

We illustrate some of the main ideas with an example. Imagine a person who must choose a course of action. The choice is to be made in two phases or moves, with two alternatives available at each move. Thus there are altogether four possible decision paths. Suppose that only one of the four paths is desirable. We depict the situation in Figure 1.

The black dots in the picture represent different points, or positions, that can be arrived at in the decision-making process. Positions on a tree of this sort are also called "nodes." The node labeled R is the starting point, or "root" of the decision tree. The two alternatives at the first move are labeled A and B. There are two further alternatives stemming from each of the nodes A and B. Of these, only one represents the desirable end result. This correct "terminal node" is labeled with a 1, and the remaining terminal nodes are labeled with –1's. We think of these numbers as the payoff to the decision maker for arriving at the corresponding positions. To achieve the payoff of 1, the person must first choose A, and then take the lower branch stemming from A. Any other path results in a payoff of –1. (A payoff of –1 means that the player loses 1.)

Note that once alternative B is chosen, it is not possible to reach any payoff higher than –1. Thus we may reasonably assign the value –1 to node B. On the other hand, it is possible to reach the payoff 1 from node A, so we assign the value 1 to node A. The values that we have assigned A and B are the highest possible values that can be reached from those nodes. Working back further, we may similarly assign the value 1 to the root node R.

We have illustrated a special case of a procedure, called "backward induction," by which we can systematically assign values to the nodes of any finite one-person decision tree, provided all of the terminal nodes have been assigned values. In general, to find the value of any node whose successors have already been assigned values, simply take the maximum of the values of its successors.

Continuing in this fashion, we can assign values, one level at a time, to all of the nodes, starting from the end of the tree and eventually reaching the root node.

Once we specify who moves at each node of the tree, backward induction also works for two-person decision trees in which the two persons (traditionally called "White" and "Black") compete against one another. Such a situation is often called a "game" (or more precisely, a "zero-sum game"). To find the value (to White) of any node whose successors have already been assigned values, take the maximum of the values of the successors if the node represents a position in which it is White's move, and if Black is on the move, take the *minimum* of the values of the successors. This is the so-called "min-max" procedure. Both it and the simpler "max" procedure used for the one-person situation work equally well when the terminal nodes have been assigned values other than 1's and −1's.

Once backward induction has been carried out, it is often useful to talk about decision paths along which the "best" choices are made by the people involved. To find such a path, start at the root node, and then choose at each successive node what is best for the player on the move; that is, choose the node that has been assigned the best value for that player. (If there is a tie, choose arbitrarily among the best nodes.) This procedure results in a sequence of nodes that we call an "intended path" or "intended line of play." In the computer chess literature, an intended path is also known as a "principal continuation."

For a finite game, such as chess, in which the outcome or payoff depends solely on the final position, backward induction implies that the strength of any intermediate position is entirely determined by the strengths of the positions that can be reached from it. This insight led the logician Zermelo in 1912 to prove that for chess, every position, including the starting position, must have a value (win for White, win for Black, or draw), representing the outcome that would inevitably result from best play by both players.

This kind of argument was implicitly familiar to the leading chess players of the 19th century. The concept of backward valuation had also been introduced into economics by the great Austrian economist Karl Menger in the 1870s. (He imputed value to productive machinery from the value of the consumption goods produced.) Wilhelm Steinitz, the first universally acclaimed world chess champion and the acknowledged father of positional chess, who incidentally lived in Vienna at the same time as Menger, took the argument one step further. He reasoned that the valuation of a position could sometimes be obtained from general principles, without any further calculation. To the extent that positional considerations lead to correct valuations, they must agree with the results of backward induction. As Emmanual Lasker, the mathematician who defeated Steinitz for the world title, put it, Steinitz's insight led to the death of the romantic school of chess. If it can be seen on purely positional grounds that White has at least a draw, there is no point in Black making long calculations to find a brilliant winning move. Until Black can see that White has made a mistake, no such move will be available.

Let us now return to our simple one-person, two-move example. Think of the tree in Figure 1 as the initial part of a much larger decision tree, with the values being the ones that would be arrived at by backward induction from the final nodes of the larger tree, *if such calculation were possible.* But suppose that such calculation is not possible. Instead, the decision maker is only capable of guessing at the values of each of the nodes in the smaller tree, basing his guesses on "positional principles." This is essentially the situation faced by the chess-playing computer. In the 1950s, the information theory pioneer Claude Shannon suggested an algorithm, combining the ideas of Zermelo and Steinitz, for how a choice should be made between A and B. According to Shannon, the decision maker should assume that his guesses for the values of the four terminal nodes of the tree in Figure 1 are the *actual* values of these nodes, assign values to A and B accordingly by backward induction, and then use those values to choose between A and B.

This is the Shannon algorithm with two-move look-ahead. Of course Shannon realized that the guesses being made are likely to be wrong at least some of the time, and he suggested two ways in which the procedure could be improved. One way is to program more sophisticated positional principles into the computer, so that it can make better guesses. The second way is for the computer to look further ahead. In practice almost all of the improvement has apparently come from the second approach.

Our question is why this should help at all, at least when the later positions are as hard to evaluate as the earlier ones. In our example, let us assume that there is some probability p that the decision maker correctly guesses the value of a given position, so that with probability $q = 1 - p$, he guesses the wrong value. We also assume that the errors in these guesses are independent of one another. (By making these assumptions, we are implicitly eliminating the possibility that decisions are easier deeper in the tree.) Now comes a surprise: it turns out that, no matter what the terminal values, and no matter what the value of p for $1/2 \leq p \leq 1$, the decision maker would do better to look only one move ahead! (There is no point in considering values of p smaller than 1/2, since even a random guesser gets the value right half of the time).

As stated earlier, there are others who have noticed that the Shannon algorithm's ability to make a correct move diminishes with deeper search depth, at least when the error probability is the same for all positions. In a book called *Heuristics*, by Judea Pearl, there is a chapter devoted to this phenomenon, called the "search-depth pathology," in which it is proved that if the error probability is constant, then the Shannon algorithm degrades to complete randomness as the depth of search depth goes to infinity. There are also some results that attempt to explain the empirical fact that the Shannon algorithm is nevertheless quite successful. But there is no rigorous discussion of how the Shannon algorithm might be modified to overcome the search-depth pathology.

What goes wrong with the Shannon algorithm? First, since it does not recognize that its guesses may be mistaken, it does not use the available information optimally. If, for instance, the guesses for the terminal nodes were 1,1,1,–1, from top to bottom, then it would seem obvious that A is a better choice than B, but in this case the Shannon algorithm treats A and B indifferently. Second, the Shannon algorithm does not attempt to determine which of the nodes should be most profitably examined, but instead always examines the deepest nodes it can reach. Deeper search means looking at more nodes, but if one then restricts one's attention to only the deepest nodes in the tree, searching deeper is no longer synonymous with using more information.

A Little Mathematics

We wish to resolve our puzzle about foresight and calculation by mathematical proof, and so without being overly technical, we describe here the theoretical framework in which we are carrying out our investigations. We begin with an arbitrary tree of nodes, at each of which we assign a player to move. The Zermelo values of the nodes are unknown to the players, but they can guess the value at a node (and perhaps other things) by observing it. In this context we can give mathematical formulas for the optimal choice of move, given any set of observations, and also the formula prescribing the best node to examine first. Recall that this constitutes the first step in our methodology. Afterward, we will describe the progress we have made with the second step, which is to extract general principles from the mathematics. Of crucial importance to the second step is that in the first step, we consider only those models in which spurious statistical correlations do not prevent sensible decision making. For example, in general some apparently irrelevant node might be perfectly correlated with the decision a player should make. We will not attempt to create a theory to handle such situations. By assuming the "Markov property," as explained below, we rule out such anomalies. As we will see, the Markov property is a reasonable description of many decision-making contexts.

To make our analysis mathematical, we must first come up with a rule for assigning values to the terminal nodes of the tree. For example, one often-used method is to assign values to terminal nodes randomly and independently. This approach has the feature that trees are "grown" backwards, namely from the terminal nodes to the roots, leading to results that are quite unnatural. We grow the trees from the root upward, using the Markov property.

To do this in a realistic manner, we suppose that positions come in several types. The type of a position tells us its true value along with several other things, which might include information about how difficult the true value is to guess and what types of positions it is likely to lead to. This is in accordance with the way good chess players tend to think about positions. They might, for example, look at a position and say something like: "this position is difficult to evaluate, but it seems clear that White's chances lie in a Kingside attack, while Black must seek counterplay in the center," or "this type of position tends to lead to very dull play in which neither side can gain an advantage."

We typically make the following assumption: once we know the type of a position x, our beliefs about the types of the successors of x are not affected by any knowledge about the types of any nodes that are not successors of x. This kind of assumption is often called the "Markov property." It is a reasonable approximation to what happens in games like chess, where once you observe all you can about a position, your predictions about the successors of that position should not be affected by knowledge or lack of knowledge about the types of positions you reached along the way, or by the types of positions you could have reached if you or your opponent had made a different earlier move.

Since even experts sometimes read a situation incorrectly, we want to include in our model some way in which errors occur, that is, to "quantify our ignorance." We assume that for any given set of position types, the probability that we will guess that the immediate successors of some position x have those types depends on two things: the actual types of the immediate successors of x, and the guess that we made for x itself (if such a guess was made). This assumption is similar to the Markovian assumption made earlier. It includes as an interesting special case the possibility that all guesses are made independently of one another, as in the example illustrated in the Figure 1.

Some General Principles

We will give here five principles that can be deduced from our mathematical formulas. Afterwards, we will talk about how the Shannon algorithm fares with respect to these principles, and we will mention several guidelines used by human decision makers that seem to be in agreement with our principles.

1. *The proximity principle.* Think about the immediate consequences of your actions first! More precisely, if y and y' are both successors of x. and if y' is also a successor of y, then y is more informative about x than is y'. The relationship can sometimes be reversed if we are sufficiently certain of our guesses about y' and/or sufficiently uncertain of our guesses about y. In such a case, the "clear sight principle" comes into play, as explained below.

2. *The relevance principle.* Examine the most promising alternatives first! Or, in the context of games, continue your investigations first along the apparent intended lines of play. In particular, if you want to learn about x by looking at one of the immediate successors of x, look at the node y that is *a priori* most likely to be chosen by the player moving at x.

   If y and y' both succeed x, and y' is deeper in the tree than y, but *does not* succeed y, it might seem that we should first examine y. But this will not be true if along the path from x to y, the moves along the path to y' are more or less "forced" (i.e., the alternatives at each node on the path are obviously bad for the player moving at that node,) and if y' seems like it might be a good position for the player who is currently trying to make the decision.

3. *The family principle.* Make decisions that lead to many good options, rather than relying on one excellent possibility. Having many successors nodes with reasonably good prospects is better than having a few successors with great prospects. Formulating this principle in precise mathematical terms leads to some interesting questions of statistics that, to our knowledge, have never been addressed. Namely, suppose you observe two sets (or "families") of numbers, and want to choose the set that contains the largest number (the "best child"). If your observations are subject to error, how should you choose? The family principle says that you need to consider the apparent combined strengths of all the children in a family, rather than just that of the apparently best child. There are two distinct reasons for this. The first is that for the family that has a greater number of "promising" children, it is more likely that one of them will actually be much better than your observations led you to believe. The second is that for a family with one very promising child and many mediocre children, the fact that you observed many mediocre children makes it more likely that your estimation of the apparently outstanding child is too high.

4. *The stability principle.* Consistency increases confidence. If our guesses were accurate, we would guess values along the intended path that changed very little. Thus, if there is a terminal node whose predecessor nodes look consistently good from beginning to end, we should prefer that over a terminal node whose predecessor nodes appear to fluctuate in value, which in turn is preferable to a terminal node with consistently bad predecessors. If there is a lot of fluctuation in the values along the path to a terminal node in our search, we may want to increase our confidence in its evaluation by searching deeper.

5. *The clear sight principle.* If possible, base your decisions on clear-cut results. Our formulas show that observations with the lowest error probability typically receive the most weight. In the extreme case, when all of the error probabilities are very small, the Shannon algorithm might work well. We must emphasize the "very small" here, because this principle is essentially in conflict with the previous principles, and it doesn't take much in the way of observation errors to make them dominate. Judea Pearl shows in his book, under certain special assumptions on the decision tree, the most important of which is independence of the values of the terminal nodes, that if the proportion of absolutely clear positions in the tree goes to 1 as the tree gets deeper, then the search-depth pathology goes away, and basing your decision on the values of the terminal nodes is a good idea. This principle is the reason he gives for the successes of the Shannon algorithm. The Shannon algorithm should be modified to include some assessment of the size of the error probability for each node that it looks at.

Not surprisingly, there are several folk maxims that sound like the principles just given. For example, the proximity and stability principles both are related to the phrase "the end never justifies the means," while the family principle sounds like "don't put all your eggs in one basket." This last phrase is also a warning about reliance on the clear sight principle in cases where we think our guessing errors are too unlikely to worry about ("I've got a sure thing in the fifth race..."). "Follow your nose" could be interpreted as the proximity principle, and "go with a winner" is clearly a form of the relevance principle. "A bird in the hand is worth two in the bush" is a version of the clear sight principle. The mathematics confirms, but also gives precision, to the proverbs of everyday experience.

Good chess players also seem to be aware of the principles we have given. Grandmasters typically do not choose a move based on looking far ahead. Instead, they

usually choose one or two plausible moves based on looking only one move ahead (the proximity principle), and then investigate those moves further by following what look like the most likely continuations (the relevance principle). In certain circumstances, they do look ahead to try to find positions whose value is obvious (the clear sight principle), so that they can avoid making blunders or falling into traps, or so that they can possibly find forcing moves that lead to a clearly won position. They have a keen sense of what types of positions will make available many good continuations for them (the family principle), and they are suspicious of lines that seem to end well, but are reached by way of positions that appear unsafe (the stability principle).

Evidence from the Chess World

Since the Shannon algorithm is unaware of its own "bounded rationality," it ignores the principles just enunciated. As far as we know, in the 40 years since Shannon first introduced his method, there have been only three modifications that attempt to compensate for these limitations. All three of them have dramatically improved the play of computer chess. The first one, introduced early on, allows the positional evaluator to assign a whole range of values to positions, even though there are only three possible true values for any position (win, lose, or draw). The effect of this procedure is that positions which are clear wins or losses are given extra weight (because they are given very high or low values respectively) in the Shannon algorithm. This is in keeping with the clear sight principle.

The second innovation is a limited kind of stability analysis. It is typically used in chess when a piece capture, say, White queen takes Black knight, occurs at the end of a path in the tree of moves being searched. Immediately after this capture, the value of the position appears to suddenly have changed, in favor of White, since Black is missing a knight. But it is often the case that in such circumstances, White's queen can be taken by Black on the very next play. Since queens are usually worth more than knights, the position is, in fact, good for Black. This phenomenon is known as the "horizon effect," and is avoided by searching deeper in those positions where pieces (such as the White queen) are "hanging," or vulnerable to being captured. Note how this is a special case of applying the stability principle.

The third modification is known as "singular extension," and was first introduced by the Deep Thought team in 1987. It noticeably improved the play of Deep Thought, and is now used by all of the top chess-playing programs. Briefly, the idea behind singular extension is that whenever the program finds a move that is vastly better than all of its alternatives, it looks deeper to check itself. Singular extension can be understood directly from the relevance principle. If a move to y following x is dramatically superior to its alternatives, then the relevance of successors of y will consequently also be high, and thus worthy of further investigation.

All of the modifications described here were motivated more by ad hoc practical considerations than by any deep understanding of general principles. There has been no attempt at all to be systematic, yet the three most important improvements in the Shannon algorithm can all be understood on the basis of our general principles. We hope that our work will lead to further improvements.

Where Do We Go From Here?

We have derived a set of general principles for optimal decision-making in complex environments. As a concrete application we investigated the game of chess, and we have tried to show that our principles explain all of the major improvements in computer chess programs (aside from those connected with computing speed). We further suggest that they roughly correspond to principles used by good human players.

We also believe that the practical application of our principles to chess play has not yet been exhausted. The clear sight principle and the relevance principle have indeed been partly embodied in the strongest programs. But the stability principle is used only in a very attenuated form, as a reason for further search-depth when the material is unstable. The application could be enormously enhanced by looking ahead after any position whose positional evaluation is unstable along the path leading up to it. The principle could be applied even without further search depth, by using the guessed values of the predecessors of a position to alter our guessed value for the position itself. The proximity principle guarantees that unless the positions are getting easier to evaluate, early evaluations are more reliable anyway. The family principle suggests that we should let our evaluation of a position depend somewhat on the guesses we make for its neighbors. The precise mathematical form of this will depend on the kind of ignorance that is present. Furthermore, implementation of a procedure based on the family principle is probably less compatible with the Shannon algorithm than any of the other suggestions we have made. But we believe that after much research, there may be something of practical value here as well.

As far as the world outside of chess goes, it seems that people have been at least vaguely aware of our principles all along. But by giving them a somewhat more precise formulation, we have illustrated how mathematics can refine and clarify, and even improve, the way in which we make day-to-day decisions.

—John Geanakoplos and Larry Gray
June 27, 1991

John Geanakoplos, Yale University, was 1990–91 co-director of the SFI Economics Research Program, and Larry Gray, a professor at the University of Minnesota, was a participant in the program.