Computational Intelligence, Volume 13, Number 4, 1997

SIMULATED ANNEALING AND THE GENERATION OF THE OBJECTIVE FUNCTION: A MODEL OF LEARNING DURING PROBLEM SOLVING

JONATHAN CAGAN*

Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh

KENNETH KOTOVSKY*

Department of Psychology, Carnegie Mellon University, Pittsburgh

A computational model of problem solving based on significant aspects of human problem solving is introduced. It is observed that during problem solving humans often start searching more or less randomly, becoming more deterministic over time as they learn more about the problem. This two-phase aspect of problem-solving behavior and its relation to learning is one of the important features this model accounts for. The model uses an accelerated simulated annealing technique as a search mechanism within a real-time dynamic programming-like framework upon a connected graph of neighboring problem states. The objective value of each node is adjusted as the model moves between nodes, learning more accurate values for the nodes and also compensating for misleading heuristic information as it does so. In this manner the model is shown to learn to more effectively solve isomorphs of the Balls and Boxes and Tower of Hanoi problems. The major issues investigated with the model are (a) whether such a simulated annealing-based model exhibits the kind of random-to-directed transition in behavior exhibited by people, and (b) whether the progressive discovery of the objective function, even when given very little or poor initial information, is a plausible method for representing the learning that occurs during problem solving and the knowledge that results from that learning.

Key words: problem solving, human problem solving, simulated annealing, reinforcement learning, Tower of Hanoi Puzzle, Balls and Boxes Puzzle, Chinese Ring Puzzle.

1. INTRODUCTION

This paper presents a model of problem solving that simulates significant aspects of human problem-solving behavior while also functioning as a computational model for problem solving. The model is best described as a hybrid model in that it combines computational approaches that are similar to a number of real-time reinforcement machine learning approaches (Sutton 1988; Barto, Sutton, & Watkins 1990; Barto, Bradtke, & Singh 1995) while at the same time attempting to model some important aspects of human performance and to do so while making cognitively plausible assumptions about many of the basic information processing mechanisms and limitations that operate in the model. The model proposes that in many problem situations, search within a new situation is initially largely random, but becomes more deterministic as problem solving progresses and, in particular, as one learns more and more about the search space. The model assumes that some landmarks might exist that can reasonably be expected to represent task-domain knowledge the problem solver brings to the task. As one moves through the search space, the indicators help one adjust their knowledge about the space, thereby learning an increasingly more accurate evaluation function that is operated on during the search process.[1] The model thus progressively develops a representation of the problem space while also searching it for the goal using a stochastic optimization technique. In particular, the computational model employs an accel-

*Order of authorship is alphabetic.

Address correspondence to the authors at Carnegie Mellon University, Pittsburgh, PA 15213; e-mail: jon.cagan@cmu.edu; kotovsky@cmu.edu

[1]Properly, this evaluation function should be called a "subjective" function because it represents the model's current and evolving understanding of the value of states in the space. An instance of this subjective function at any time is the "objective" function that is used within the optimization framework of the simulated annealing algorithm. We will refer only to the objective function in the remainder of this paper, in keeping with the standard usage of the term in the optimization literature.

erated version of the stochastic optimization technique of simulated annealing (Kirkpatrick, Gelatt, & Vecchi 1983), used in conjunction with objective function modifications based on knowledge obtained via moves between two neighboring states. In regard to the objective function modifications, the model is similar to the temporal difference technique described by Sutton (1988). This paper in particular investigates the following issues:

- the extent to which simulated annealing is a useful description of significant aspects of human problem-solving behavior, and in particular, the transition from randomness to more directed behavior
- the extent to which the progressive discovery of the objective function during search is a plausible and useful technique for acquiring and representing knowledge gained during problem space exploration
- the extent to which the technique can work to gradually enable more and more efficient solutions of the problem without giving the system large amounts of a priori knowledge of the task-domain within which it will operate. One question addressed here is whether such a simple model of learning and knowledge representation can learn to solve problems that are reasonably difficult for humans.

The model operates in a task environment defined as a graph of linked nodes, the links radiating from any node representing the set of legal moves available from that node. Additionally, each node has an associated objective function value that initially may be unrelated to the actual value (proximity to the goal) of that state. The model attempts to move from a start state to a goal, updating as it does so the objective function of each node it encounters. The updating is based on its discovering adjacency relations among the nodes it traverses.

This work represents an attempt to determine the functionality of a model with a modest and cognitively realistic degree of knowledge about the domain it is operating in, i.e., the minimal amount of knowledge that any problem solver attempting to solve that particular problem might reasonably be expected to know about the environment being searched. Examples of such knowledge might include information attainable from the problem statement, such as major subgoals that exist on the way to a solution or, possibly, reasonable expectations about states in the immediate vicinity of the goal. This knowledge is represented in the form of particular values of the objective function at those states. In addition, a problem solver brings some general strategies or heuristics that might lead to expectations about the value of particular states. One example would be the heuristic of downhill search (or hill-climbing), whereby a value is placed on the superficial appearance of progress toward the goal. That people possess such a heuristic is a reasonable assumption about the initial state of human subjects' general heuristics, even if those heuristics are often erroneous and lead to a misleading metric of progress. In these latter, erroneous cases, which usually arise in what are called *detour problems*, an interesting issue for the model is whether it can recover from the false downhill search-induced movement to local minima,[2] and find a path to the goal. In any case, the problem description presented to the model at times includes some initial knowledge state values that are either known or assumed, and that might be expected to either help or impede the solution process. We show that in either situation the model consistently learns a correct objective function such that search for the goal becomes increasingly quick and efficient.

[2]Note that this paper focuses on problem *minimization*; the inverse discussion would hold true for maximization. *Downhill searching* and *hill-climbing* are thus equivalent heuristics in the respective representations of search as minimization or maximization.

The approach taken in this paper represents search as an optimization problem and uses a standard optimization method as the search technique. This approach is somewhat minimalist in that not only does it use a simple optimization mechanism but also a knowledge representation system that is not dependent on the strategic use of a highly elaborated knowledge base. In contrast, other methods of solving optimization problems such as monotonicity analysis (Papalambros & Wilde 1988; Agogino & Almgren 1989; Cagan & Agogino 1987), activity analysis (Williams & Cagan 1996), as well as standard numerical gradient-based approaches (Vanderplaats 1984) take more strategic approaches. The knowledge representation used in the current work contrasts with PDP approaches (McClelland & Rumelhart 1986; Rumelhart & McClelland 1986) in relying on a uniquely assigned value for each state in the problem space that is only affected by interactions with immediately adjacent states rather than a more distributed representation. The work further stands out in not relying on knowledge-rich representational systems (e.g., Feigenbaum 1977) nor powerful AI search techniques (e.g., Rich 1983).

This work uses on-line machine learning techniques to learn about the evaluation space. In particular, the method is a type of real-time dynamic programming (Barto, Bradtke, & Singh 1995), employing a $\lambda = 0$ temporal difference process (Sutton 1988) where state value updates are based on a one-step look ahead mode. The limited one-step look ahead mode is critical to this work, not for the efficiency of the machine learning technique, but rather for its relevance to cognitive processes. Our model invokes processes that are at least broadly similar to or compatible with those seen in the behavior of human subjects, in particular by not positing an unrealistic (for humans) ability to store short-term information. Other approaches, such as $\lambda = 1$ mechanisms that approach more off-line methods where no reinforcement is applied until the goal state is found, place very large demands on memory for the pathway taken to reach the goal on each iteration. These methods include those that depend on the retention of complete move records and state visit frequencies to update knowledge of the environment (Fulcher 1992). Similarly, counter-based, recency-based, and error-based methods (e.g., Thrun 1992; Sutton 1990; Thrun & Möller 1992), while very effective from a machine learning perspective, are also viewed as too demanding of cognitive resources. These methods, although able to take advantage of the computational power of the machine to be quite efficient in their ability to learn the evaluation functions, differ in not being focused on human problem solving. Although humans may be able to learn more than pairwise relations, it is extremely unlikely that they can learn and store complete move pathways while engaged in active problem solving.

Our learning mechanism differs from the literature in that our approach is meant to be consistent with human cognitive processes, not solve the machine learning problem with maximal efficiency. We do, however, borrow features from the machine learning literature. Simulated annealing is a Markovian process; our method is most similar to, and takes many features from, Q-Learning (Watkins 1988; Watkins & Dayan 1992), which works within controlled Markovian domains. Like Q-Learning, the value at each state represents the current approximation of the correct objective (or evaluation) function, the evaluation function is updated during each move, and the experimentation strategy uses probabilities to determine the move to take.

However, we introduce a unique simulated annealing method for control of the experimentation strategy. Not only is the method used to make decisions as to whether to make a move or not, but the aggressiveness of the technique is coupled with the learning framework. We introduce an *accelerated* annealing schedule that, over successive iterations, becomes more aggressive in its transition from random to deterministic search. We show that learning about the problem space is not enough; the probabilistic tendency to violate that acquired

knowledge and move to higher energy (or evaluation) level states, seen to be useful early in problem solving, must be sharply curtailed through the accelerated annealing in order for the learning to be maximally effective. However, without the learning of the objective function the acceleration of the annealing does not result in an improvement in performance, and can even be detrimental to the model's behavior. Further, a typical use of simulated annealing optimizes a fixed objective function; in this work the technique is searching over a *constantly changing* objective function. We argue and support in this paper that although a pure simulated annealing mechanism is unlikely to model the complete cognitive process, the concept behind the accelerated annealing mechanism could very well be a part of the human cognitive mechanism in problem solving.

Another difference between our method and the literature is that, rather than updating the current state to the new state in some predetermined way, both the current and new states are updated based on percentages of the difference between the values of the states. This aspect of the model is thus similar to the delta rule used in parallel distributed processing models of cognition (McClelland and Rumelhart, 1986) but with the absence of an overt teacher. The perspective is that humans learn proximity of states, and thus our model learns that two states are near *each other*. Since the initial value of the states may be incorrect or approximate, both states are adjusted toward each other rather than assuming that the new state has a more correct value and thus the current state updated toward it. Rather than using the Bellman equation, the updates do not represent the number of moves to the goal but rather the relative position of one state with respect to the other; while search is taking place the updates occur independent of where the goal state lies.[3]

To the extent that the work reported herein attempts to model significant aspects of human behavior, and to represent its learning from experience without reliance on cognitively unrealistic amounts of constantly revised knowledge, or strategic decision making at each step, even at the sacrifice of computational efficiency, it represents something of a hybrid between a machine learning/AI based model and a cognitive simulation. To illustrate our approach, we focus on two problems within which human performance has been thoroughly explored in the problem-solving literature: the Tower of Hanoi Puzzle and the Balls and Boxes Puzzle (an isomorph of the Chinese Ring Puzzle). We will first describe each of these problems, discuss their generality, and examine previous empirical studies of human subjects' approaches to their solutions. From these results we propose a model of problem solving. To simulate this model we introduce our computational model using simulated annealing and objective function adjustments that progressively refine the objective function. We illustrate the computational model on both the Tower of Hanoi and the Balls and Boxes Puzzles, where the program consistently learns an effective objective function over several iterations while at the same time decreasing the number of moves needed to reach the goal state. This is true even when it is given no information or even misleading information about the true evaluation function.

Beyond showing generality of the model for solving various problems, we choose both problems because they illustrate different aspects of the model. The Balls and Boxes Puzzle has the simpler problem space of the two but is nonetheless quite difficult for people to solve. Due to the linear nature of its space, we are able to compare our model's solution to move records from humans solving the problem; as well this problem readily illustrates an important aspect of people's problem-solving behavior—the *final path behavior* discussed below. The Tower of Hanoi problem has a larger, more complicated solution space. This

[3]Other work (e.g., Sabes, 1993; Baird 1995) has updated states toward each other by adjusting both sides of the Bellman equation to guarantee convergence, not to improve a cognitive-based model.

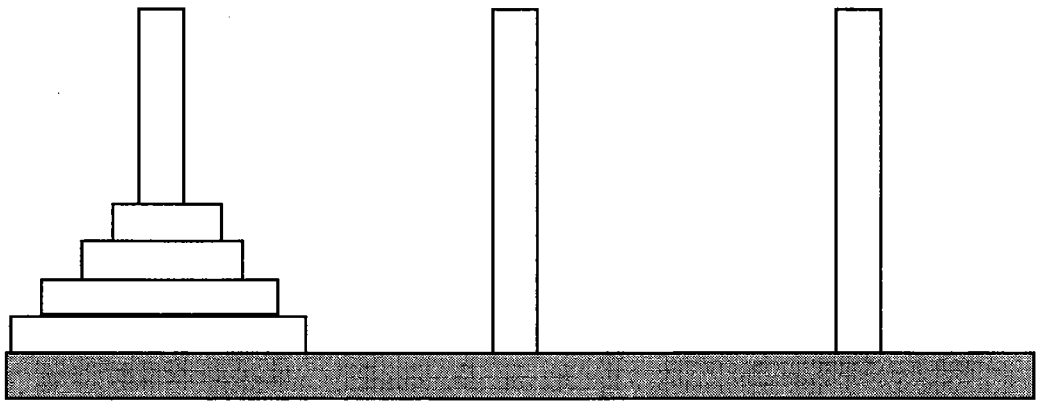

FIGURE 1. Tower of Hanoi four-disk problem. Four disks start on the leftmost peg and are to be moved to the rightmost peg.

problem better illustrates the learning performance of the model in larger spaces and provides a framework to test the learning and search aspects of the model. The Tower of Hanoi results will be shown first to explore the capabilities of the model, followed by the Balls and Boxes results to better compare the model to records of the performance of humans solving the same problem.

2. TOWER OF HANOI PUZZLE

The first problem used in the modeling work reported here is the four-disk Tower of Hanoi problem (see Figure 1). The problem consists, in the external representation that is presented to subjects, of a series of disks placed in descending size order, on one of three pegs (the "start peg"). The goal of the problem is to move the stack of disks to another of the three pegs, the "goal peg." Moves are subject to the restrictions that only one disk may be moved at a time, if a peg contains more than one disk only the smallest may be moved, and a larger disk may never be placed on a smaller disk.

The problem can be defined in terms of the possible "states" of the problem or puzzle and the legal moves that convert one state to another. In these terms, a problem is represented by a graph in which problem states are nodes and legal moves are links joining these nodes. This representation of the problem space as a series of nodes connected by links designating legal moves is the form of the problem space that is searched by our computer model. This mapping can be thought of as an external search space that depicts all possible legal problem configurations or knowledge states, and the connections between them. It is in this search space that the computer models we report on search for a solution. The "objective function" (see note 1) is the set of values associated with the nodes in this space. Unlike the human subjects, the computer does not have any representation of the surface features of the problem in the form of disks on pegs or, in the Balls and Boxes problem, balls in boxes.

The number of moves in the minimum solution path is $2^n - 1$, where n is the number of disks in the Tower of Hanoi. Similar relations between number of disks (balls) and minimum solution path length hold for the Balls and Boxes problem to be described below. The structure of the problem space in the Tower of Hanoi problem is, however, more complex due to the

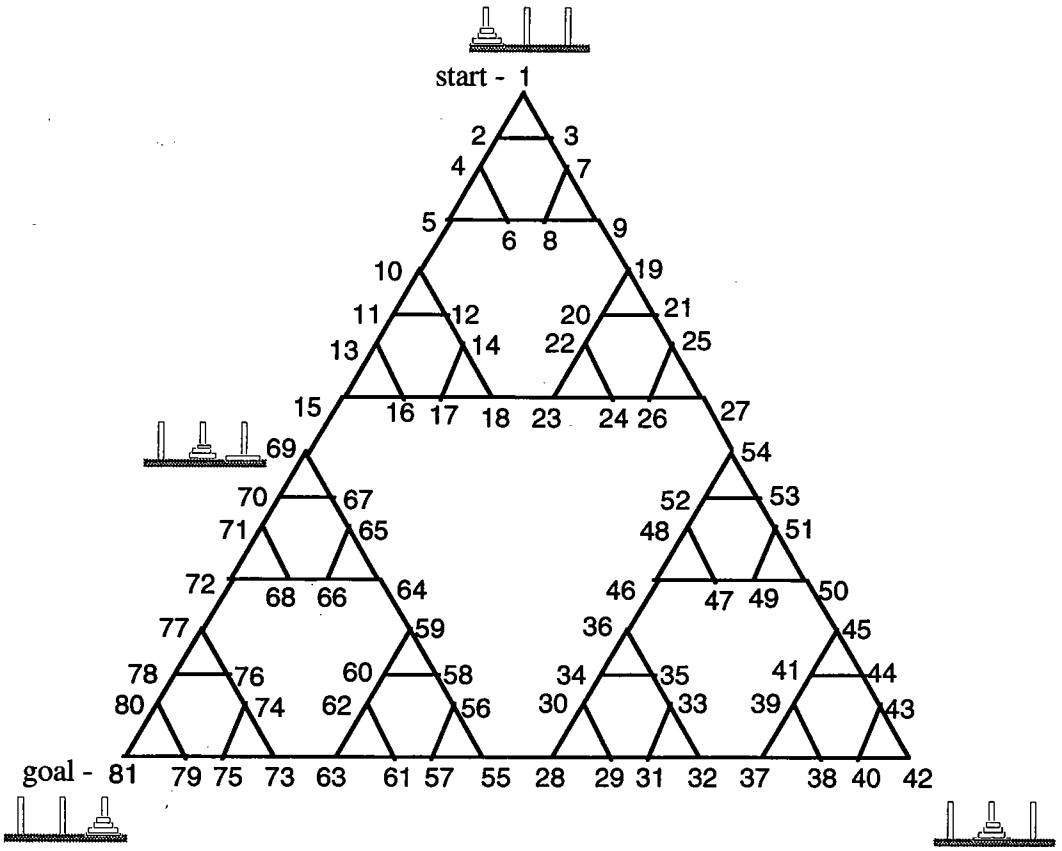

FIGURE 2. Tower of Hanoi four-disk problem problem space. Numbers label the states for identification. The typical starting position is position 1, with the goal position at 81. The most direct path traverses positions 1, 2, 4, 5, 10, 11, 13, 15, 69, 70, 71, 72, 77, 78, 80, and 81.

fact that there is a choice of three moves at all but three of the nodes in the search space (the three nodes where all the disks are piled on one peg), rather than two choices at each choice point as will be seen with the Balls and Boxes/Chinese Puzzle. The problem space of the four-disk version of the Tower of Hanoi problem is depicted in Figure 2. The space consists of 81 different states and the minimum solution path is 15 moves. As we can see from that figure, the move choices at almost all nodes consist of two possible new moves and the retraction of the most recent move (the return to the immediately prior state). The only exceptions to this are the three "corner" states where all of the disks are stacked on one peg and only two move choices are possible. The problem can be solved for any number of disks, and has the property that an n-disk problem can be viewed as consisting of an $n - 1$ disk problem (moving all but the largest disk off the start peg), a move of the largest disk to the goal peg, and then another $n - 1$ disk problem as the stack is moved back on top of the largest disk to complete the solution. The solution of the typically represented Tower of Hanoi Puzzle requires violations of downhill search in that it is not possible to solve the problem by simply moving disks from the start to the goal peg; it is thus termed a "detour" problem.

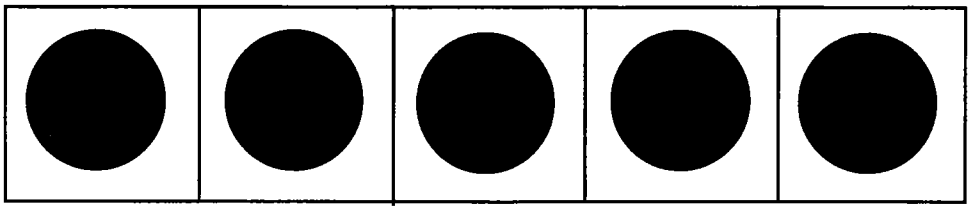

FIGURE 3. Balls and Boxes five-ball problem. The problem is shown as depicted at the start with all five balls in their respective boxes, with the goal being to remove them.

3. BALLS AND BOXES PUZZLE

The second problem explored, the Balls and Boxes Puzzle, has been used in a number of previous empirical studies (Kotovsky & Simon 1990; Reber & Kotovsky 1992) as has its much more difficult isomorph, the classic Chinese Ring Puzzle, which has been described by Ruger (1908), Afriat (1982), and Kotovsky and Simon (1990).[4]

The task, as presented to experimental subjects, is to remove five balls from five boxes. (The balls correspond to the disks of the Tower of Hanoi problem.) A move consists of inserting or removing a single ball to or from its respective box. The rules governing moves are that a ball is only free to move if the ball to its immediate right is in its box and none of the balls farther to the right are in theirs. The only exception is the rightmost ball which is always free to move. At the usual start of the problem all five balls are in their boxes and the goal is to get them all out. The situations are displayed on a CRT, and when presented to subjects, the moves are made by manipulating a mouse. This particular problem has five balls and boxes, with the usual start state having all five balls inside their boxes, the goal being to remove all of the balls. The problem is depicted in Figure 3.

The problem space for the puzzle is shown in Figure 4, which depicts the ball array that corresponds to each state in the problem space. To illustrate the move contingencies, the last five states illustrate the moves required to remove the final three balls from their boxes under the standard problem rules.[5] As always, the rightmost ball is free to move. In state 5, it is removed, producing state 4. In state 4, the leftmost ball is free to move given that only the ball to its immediate right is in its box, and it is removed, producing state 3. Next, the rightmost ball is replaced, producing state 2 from which the middle ball can be removed, given that it then has only the ball to its immediate right in the box. Notice that the first of this pair of moves (replacing the rightmost ball) could be misinterpreted as moving the subject in the wrong direction since it adds balls whereas the goal is to remove them. This necessary violation of downhill search is discussed below when the subjects' move records are analyzed. The last two moves consist of removing the middle ball and finally the rightmost one which achieves the goal (state 0). The minimum number of moves required to solve the

[4]Although the Chinese Ring Puzzle, a manipulation or "tavern" puzzle in which a metal bar must be extricated from five confining rings, is much more difficult than the Balls and Boxes Puzzle, the problem spaces are identical; thus the isomorphism.

[5]The same discussion and move sequence can be used for the rings on the bar by substituting "ring" for "ball" and "on or off the bar" for "in or out of the box".

entire problem is 21 or 31, depending on the start state as illustrated in the figure[6] (state 21 is typically used as the start state as it has all five balls in their boxes).

4. HUMAN PROBLEM SOLVING RESULTS

Typical solution paths (from state to state) obtained from human subjects solving the Balls and Boxes problem are illustrated in Figure 5. In that figure we present a set of move records for human subjects solving the puzzle. As can be seen in that figure, the subjects typically make a large number of moves with little or no net progress toward the goal, and then suddenly move rapidly and accurately to the goal. This dichotomous behavior has been labeled "exploratory" and "final path" (Kotovsky & Simon 1990). Interestingly, the seemingly insightful transition to the final path generally is not accompanied by significant verbalizable knowledge or true insight into the problem, and if given the same problem again, subjects again resort to problem solving including another exploratory and final path period. Learning is evidenced by a shortened exploratory period, but nothing approximating total understanding is evidenced on the second solution attempt (Reber & Kotovsky 1992; Kotovsky & Simon 1990). Another finding is that the linearity of the search space is not discovered by the subjects (even when they solve the puzzle multiple times) and thus the problem is not trivialized by that feature.

A similarly dichotomous exploratory/final path solution process has been found for subjects solving some difficult isomorphs of the shorter three-disk Tower of Hanoi Problem, with people typically solving the problem by wandering in the problem space for some time and then traversing the entire solution path length and solving the problem in the last 15% of the time (Kotovsky, Hayes, & Simon 1985). However, in the Tower of Hanoi problem, there are additional mechanisms that are likely to lead to the final path behavior beyond those found in the Balls and Boxes problem. The Balls and Boxes problem is characterized by subjects' inability to verbalize or even recognize strategic information, even after successful solution, while the Tower of Hanoi problem isomorphs yield much more verbalizable information about higher-level strategic knowledge such as move planning and subgoaling (Kotovsky et al. 1985; Kotovsky & Simon 1990; Reber 1993). The implicit or nonconscious nature of the learning that occurs in the Balls and Boxes problem that allows movement onto the final path within a problem and faster solution of a second problem is an important characteristic of the problem and is particularly well suited to the type of model we propose. The additional, higher level processes found in the Tower of Hanoi are not the focus of the current modeling effort, although any implicit learning that occurs there (or in other problems—see Broadbent & Berry 1988) is.

Major empirical results include the following (Kotovsky et al. 1985; Kotovsky & Simon 1990; Reber 1993):

- Despite the simple linear structure of the problem search space, the problems can be very difficult. The Balls and Boxes problems often took hundreds of moves to solve. The Chinese Ring isomorph was even more difficult. In one study, only 7% of the subjects were able to solve the problem in 1.5 hours.

[6]In general, for an n-ring problem, there are 2^n possible states in the search space, with a resultant minimum solution path of $2^n - 1$ moves. There are a number of similarities between this problem and the Tower of Hanoi problem (Section 2). Both problems are infinitely expandable. In both problems, the most restricted piece (ring or disk) is moved half as frequently as the next most restricted, which is moved half as frequently as the next, and so on. Finally, the minimum solution path length is the same as in the n-disk Tower of Hanoi problem, although the size of the search space is not.

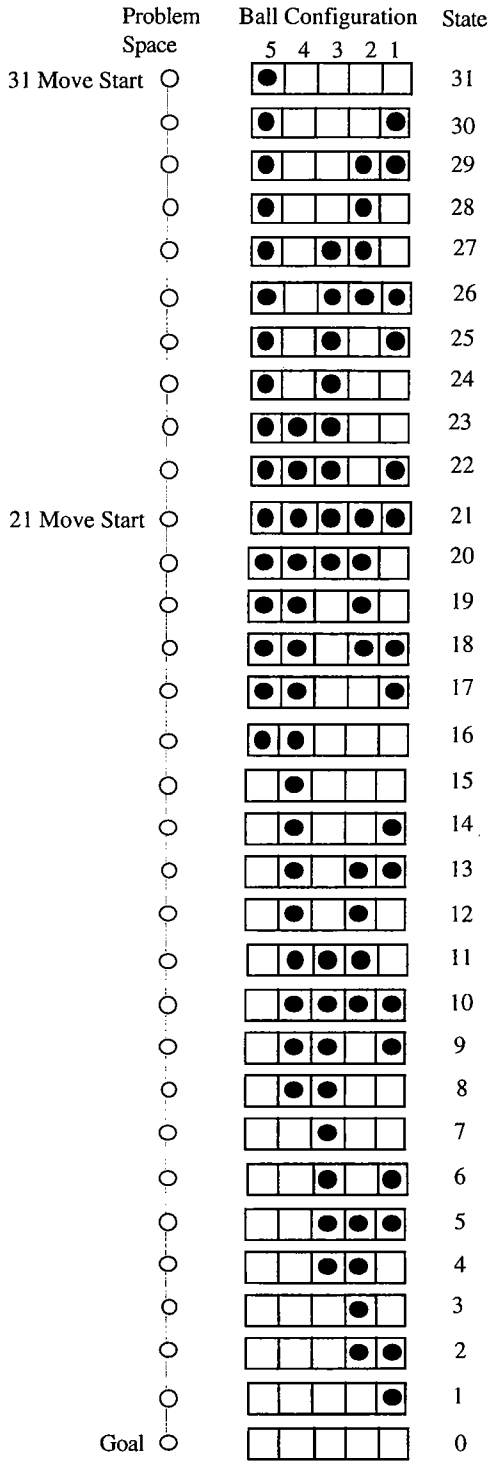

FIGURE 4. Balls and Boxes five-ball problem problem space. The linearity of the space as well as the 21-move starting position and goal are illustrated.

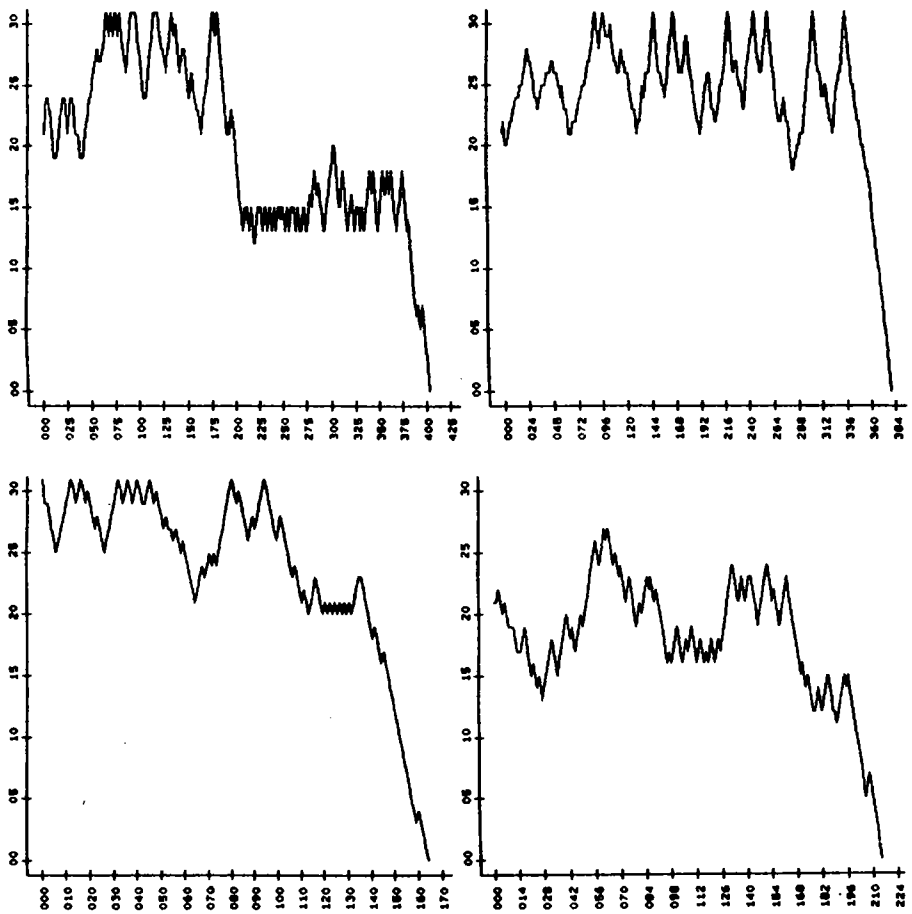

FIGURE 5. Balls and Boxes problem typical human move records. The distance from the goal is indicated on the ordinate and the move number on the abscissa (from Kotovsky & Simon 1990).

- The behavior of people solving the puzzle exhibits two phases, a long "exploratory" period in which subjects make many moves with no net progress toward the goal, followed by a rapid "final path" dash to a solution during which they move directly and unerringly to the goal. The exploratory behavior is characterized by frequent reversals of direction, a great deal of revisiting of previous visited states, minimal progress toward the goal, and, where the problem solver is not constrained from making them, a great number of illegal (problem rule-violating) moves.
- While the transition from exploratory to final path behavior seems to indicate a sudden insight into the solution, subjects are able to report very little understanding of the problem and are not able to articulate any of the rules that describe the problem or the structure of the problem space.
- Although the subjects are not able to report on or articulate the rules that describe the problem or state structure, when they solve the problem from the same initial state again, they solve it with the same characteristic move patterns but in fewer steps due to a shortened exploratory period.
- Isomorphs of the Tower of Hanoi problem that involve monsters passing globes back

and forth or changing their sizes (rather than disks moving from peg to peg as in the Tower of Hanoi) exhibit a similar exploratory–final path dichotomy. Subjects typically solve Monster problem isomorphs (traversing the total distance from start to goal after much nonprogress making "exploratory" behavior) in the final minute or two of a ten- to fifteen-minute solution attempt.

Note that the exploratory phase can be viewed as almost random search. However, the prototypical abrupt change to the final path indicates that some learning has occurred that allows efficacious move-making. We propose that this happens through subjects learning pairwise move contingencies during the exploratory period in the Balls and Boxes Puzzle.[7] In addition, there is a further decrease in the randomness of the move choices that possibly results from a decreased reliance on the downhill search heuristic which leads them to prefer moves that remove balls from their boxes (Reber & Kotovsky 1992). The subjects thus become less stochastic in their approach to solving the problem, eventually achieving an error-free and unwavering final path to the goal. Finally, on subsequent solutions of the problem, although this learning allows them to eventually perform in a less random manner and thus more rapidly solve the problem, they do not start out that way, but rather exhibit another exploratory period, albeit a shortened one, a somewhat surprising result given the evidence of understanding provided by the preceding final path. Examples of the speedup from the first solution of a problem to the second solution of the same problem range from 420/205 (first solution/second solution) at the high end to 116/49 at the low end depending on the difficulty of the particular problem isomorph used (Kotovsky & Simon 1990; Reber 1993).

5. A MODEL OF PROBLEM SOLVING

We propose a computational model of problem solving based on the preceding observations. The model has the following characteristics: It is given the basic node-link structure of the problem space with minimal information about the relation of nodes in that space to the goal. Search within that space begins stochastically, becoming deterministic over time. When the model moves between two states, it updates the objective function values of those states to reflect the acquired knowledge that each is easily reachable from the other and thus must have similar values. When the problem is re-solved, the updated understanding of the objective function space is used and search progresses in a less stochastic manner. These operating characteristics of the model are broadly based on the behavior exhibited by human subjects, in particular the fact that over time the search becomes less exploratory and moves to final path—i.e., that people start out moving more or less randomly (although they might not feel that they are moving randomly) but after some time become much more directed toward the goal. The learning of pairwise contiguity relations (here represented by moving the energy levels toward each other when a move occurs between the two states) is based on the plausible assumption that people can learn that two states are close to each other when it is possible to move from one to the other in one move. Note again that this contiguity is bidirectional; both state values are adjusted rather than just the one from which the algorithm moves as is typically done in machine learning.

[7] In Monster problem isomorphs of the Tower of Hanoi, an additional mechanism comes into play that involves the working memory load imposed by surface representational features of the different isomorphs, and how they impact on planning pairs of moves. These mental model representational features that account for difficulty differences among the various isomorphs are not the focus of the current model, which operates by evolving a representation of the underlying structure of the problem space.

Our model uses the stochastic optimization technique of simulated annealing as a method of exploration. Simulated annealing is a zero-order optimization technique that begins in an essentially random manner but quickly becomes more deterministic while still maintaining some stochastic characteristics. The model uses simulated annealing to search for the goal in the problem space by moving between connected states. At the start of a model run the states are assigned values based on estimates of the a priori knowledge that humans solving the same problem might be expected to have. Thus each such state is initially described as being at one of three levels of certainty: "known", "assumed", or "unknown". A known state has a known and fixed evaluation value; an assumed state has a weakly determined value that is adjustable; an unknown state has an unknown value (which is arbitrarily set to be relatively high and is readily adjustable). Unless there is some plausible reason to assume that a human would have a priori knowledge about a state, it is set at the default initial value of "unknown". As the model moves between states, the values of the two states move toward each other, indicating that the two states are easily reachable from each other and thus must have evaluations near to each other. The relative amount of movement of the two states is based on their relative certainty level. Exhaustive search of the entire space is not desirable; however, if the search finds a state not previously visited then an unexplored region is indicated in which it may be desirable to search. This increase in uncertainty that results from being in an unfamiliar region is accomplished by a transitory increase in annealing temperature. After the goal is found, the program is restarted at the same initial point with the updated values of the objective function at each state maintained. Additionally, the rate of reduction in the annealing temperature (as discussed below) is accelerated, thereby reducing the randomness in the search of the now more familiar space. Each of the model's mechanisms is now more fully described.

5.1. Simulated Annealing

Introduced by Kirkpatrick et al. (1983), simulated annealing is a stochastic numerical optimization technique used to solve continuous, ordered discrete and multimodal optimization problems. The algorithm selects an initial state, which is evaluated through the objective function. In an annealing algorithm the result of the evaluation is known as the *energy*. A state is selected within some predefined neighborhood, here defined as the set of nodes linked to the current state, and it is evaluated. The energy of the new state is compared to that of the original state. If it is better in its evaluation then it is selected as the current state (a "move" occurs); if it is worse then there is still a probability that it will be selected as the current state. The probability of accepting a worse state is initially high (emulating random search) and progressively decreases to zero (emulating downhill search). This probability, Pr{new}, takes the form:

$$\Pr\{\text{new}\} = \frac{e^{-\frac{E_{\text{new}} - E_{\text{current}}}{T}}}{e^{-1}},$$

where E_{new} and E_{current} are the new and current evaluations (energies), T is a variable called *temperature*, and the denominator (e^{-1}) is used to normalize the initial probability to be 1.0.

The variable T (temperature) is defined so that Pr{new} is initially 1.0 and decreases after a large number of moves to 0.0, unless the problem is solved first. The decrease in temperature is laid out in a function called the annealing schedule. Simulated annealing is often used to solve problems where 100,000 or more iterations are required; sophisticated self-adaptive annealing schedules can be defined which adjust themselves based on a statistical analysis of their behavior (e.g., Huang, Romeo, & Sangiovanni-Vincentelli 1986). In our application between 15 and 1,000 moves are all that is required and reasonable and so a fixed "natural" or vanilla schedule is used: Here the temperature is multiplied by a constant reduction factor

that is less than, but close to, 1.0 at each temperature reduction. In our implementation, the temperature is reduced at every iteration. Thus the temperature at any iteration, i, is defined as:

$$T = T_{\text{initial}} * (\text{reduction\_factor})^i,$$

where T_{initial} is the temperature at the start of the run. In this implementation, the initial temperature is set to 1.0 and the reduction factor is initially set to .99 unless otherwise stated.

The state space in the current implementation is represented by a connected graph where each node represents a feasible configuration of balls in boxes or disks on pegs and the links between nodes represent the legal moves. Such a move between any two nodes consists of moving one ball or one disk. A random move is selected from the available links and the move is taken based on the annealing criteria.

Note the characteristic of simulated annealing: the search starts with a large random component and then progressively becomes downhill search. Thus, wide exploration of the search space becomes focused into a local region where the solution is found. These same characteristics are seen in the solution paths from humans while solving the Balls and Boxes and Tower of Hanoi problems where the move-making starts out seemingly random and progresses into a delimited pathway. Our model attempts to capture this characteristic human behavior in two ways: one is simply the annealing temperature reduction and the other is the progressive smoothing of the search space as the relative proximity and values of nodes are learned. The learning in our model is due to the combination of these two mechanisms, fairly random behavior (a wide range of acceptance of candidate moves) early in problem solving due to the high annealing temperature, coupled with a decrease in temperature as more is learned about the structure of the problem space through alterations of the energy levels as moves are made. Both of these mechanisms represent reasonable assumptions about how humans may modify their behavior as they progress in a problem-solving episode.

5.2. Evaluation Adjustments (Learning about the Problem Space)

The exploration-learning capability of the model assumes three types of values for the objective function: known values, assumed values, and unknown values. This trifurcation was chosen to represent the likely variation that would be expected to exist in a human problem solver's level of certainty about different loci in the problem space. Although it is possible that people have an even more highly graded set of certitude levels, the above partitioning is a reasonable first assumption. Each state has associated with it an initial estimation of the value of the state. We assume that a problem solver typically knows he or she is starting at some distance from the goal; therefore the starting state (as well as most others) have a relatively high initial energy value, while that of the goal, which we assume is recognizable by the problem solver, has a very low energy value. In addition there might be a small number of "landmark" states between the start and the goal (such as a state known to be close to the goal) where the solver can be expected to know (or believe he or she knows on the basis of some general heuristic) either the approximate or exact value. This knowledge is instantiated by assigning an energy to each state and marking the state as "known", "assumed", or "unknown", depending on the level of certainty about its value, i.e., its energy. Known information is exact and does not change when that state is encountered; assumed and unknown values have increasing flexibility in the amount that they can change. An example of such a landmark state initial value assignment is implemented in one of the problems investigated, one version of the four-disk Tower of Hanoi problem. In that problem the only initial intermediate problem space energy function knowledge is that it is "good" to get the largest disk on the goal peg. The value of that state is initially set as "assumed"; all other states are set at "unknown" except the goal state which is always set at "known".

TABLE 1. Transition Matrix Representation of Learning (alteration of energy levels) When a Move is Made between Two States; Ratios Indicate Source/Destination.

		Destination		
		Known	Assumed	Unknown
Source	Known	0/0	0/.8	0/.9
	Assumed	.8/0	.25/.25	.1/.8
	Unknown	.9/0	.8/.1	0/0

The model learns about the space as it moves between states, changing both the value and classification of the states it encounters as well as the value of the states it moves from. In this learning, known states are more powerful than assumed states, which in turn are more powerful than unknown states. What this means is that the certitude of the value assigned to a state grants that state the power to modify other states and itself resist modification. Thus if there is a move from a known state to an unknown or assumed state, it modifies the value of the target state to bring it closer to the value of the known state. The complement is also true; i.e., a move from an assumed or unknown state to a known state moves the assumed or unknown state toward the known state. Finally, if a move is made from an assumed or unknown state to another assumed or unknown state, it not only modifies the value of the target state, but also modifies the value of the state it moves from. The assumption here is that the model learns that the two states are contiguous by moving from one to the other, and thus links them with similar energy values, given that it is easy (one move) to get from one to the other. In this way the model learns about the search space, representing its knowledge as a set of energy values associated with the states it has visited from previously known or assumed states. It thus progressively expands its knowledge of the space outward from states whose value it knows (partially or wholly) to states it encounters during the solution of the problem. This learning of pairwise relations is well documented in human behavior (Cohen, Ivry, & Keele 1990).

Again, the power of a state to change another contiguous state depends on the level of certainty of its knowledge of the energy function values for the states the model moves to and from. The difference between the energies associated with the source and destination states is reduced by a fractional amount, determined by the level of knowledge of the two states, according to the values given in Table 1. In this table the pair of numbers indicates what percentage the source (top) moves toward the destination and what percentage (of the energy difference separating them) the bottom (destination) moves toward the source. Thus, an assumed state moving to an unknown state moves the assumed source 10% closer to the unknown destination and the unknown destination moves 80% closer to the assumed source.

Initial unknown states are set to a high value, the premise being that start states are likely to be far from the goal. Once an unknown state is visited from a known or assumed state and its value adjusted, its classification is changed to "assumed". This is because there is now some information about its value, albeit incomplete or inexact information. The differences between adjacent states' values are thus reduced as the model makes moves. Note that the actual numbers used in the table are somewhat arbitrary; what is important is the relative certitude of the information for the model (or, correspondingly, for the human problem solver) indicated by the relative magnitudes of the values in the table.

Again note that the policy (i.e., learning method) used in this model contrasts with that typically used in machine learning. Here, the values of both the current and new states are