11. M. Noda, K. Yoon, G. A. Rodan, D. E. Koppel, *J. Cell Biol.* **105**, 1671 (1987).
12. E. Kimball and J. E. Coligan, *Contemp. Top. Mol. Immunol.* **9**, 1 (1983); P. J. Bjorkman *et al.*, *Nature* **329**, 506 (1987).
13. J-i. Miyazaki *et al.*, *J. Exp. Med.* **163**, 856 (1986).
14. J. McCluskey *et al.*, *J. Immunol.* **136**, 1472 (1986).
15. Cells for photobleaching were plated onto glass cover slips 1 to 5 days before use. L^d antigens were labeled with fluorescein- or tetramethylrhodamine-(Molecular Probes, Eugene, OR) conjugated Fab fragments of the monoclonal antibody (MAb) 28-14-8, or with the fluorescein-conjugated IgG of the same MAb. Tetramethylrhodamine conjugates were purified over SM-2 BioBeads (Bio-Rad) to remove noncovalently bound rhodamine [E. Spack, Jr., *et al.*, *Anal. Biochem.* **158**, 233 (1986)]. The conjugates were centrifuged at 100,000*g* for 40 to 60 min before use. The conjugates labeled cells expressing L^d or the cross-reacting D^b MHC antigens and did not label L cells, which express $H\text{-}2^k$ antigens, D^k and K^k. $H\text{-}2^k$ antigens of L cells were labeled with fluorescein-conjugated Fab or IgG of MAb 11-4.1. Both labels gave the same *D* values. FPR measurements of labeled cells were made at 18° to 20°C with a 40× objective and a numerical aperture of 1.3, giving a spot radius (e^{-2}) of ~0.8 μm. Our computer-controlled photobleaching microscope has been described elsewhere (*16*).
16. M. Edidin and T. Wei, *J. Cell Biol.* **95**, 458 (1982).
17. M. L. Wier and M. Edidin, *ibid.* **103**, 215 (1986); B. Bierer *et al.*, *ibid.* **105**, 1147 (1987); P. Salas *et al.*, *ibid.*, in press.
18. E. Yechiel and M. Edidin, *ibid.* **105**, 755 (1987).
19. Y. C. Lee, R. R. Townsend, M. R. Hardy, in *Biochemical and Biophysical Studies of Proteins and Nucleic Acids*, T-b. Lo, T-y. Liu, C-h. Li, Eds. (Elsevier, Amsterdam, 1984), pp. 349–360.
20. P. J. Bjorkman *et al.*, *Nature* **329**, 506 (1987).
21. We thank K. Ozato for supplying the glycosylation-deficient mutants, J. McCluskey and D. Magulies for supplying the truncated L^d mutant, and T. Wei for technical assistance. Supported by NIH grant AI-14584 to M.E. M.W. was supported by an NIH training grant in immunology to K. Ishizaka. Contribution 1416 from the Department of Biology.

23 June 1988; accepted 8 September 1988

A Graph-Dynamic Model of the Power Law of Practice and the Problem-Solving Fan-Effect

JEFF SHRAGER, TAD HOGG, BERNARDO A. HUBERMAN

Numerous human learning phenomena have been observed and captured by individual laws, but no unified theory of learning has succeeded in accounting for these observations. A theory and model are proposed that account for two of these phenomena: the power law of practice and the problem-solving fan-effect. The power law of practice states that the speed of performance of a task will improve as a power of the number of times that the task is performed. The power law resulting from two sorts of problem-solving changes, addition of operators to the problem-space graph and alterations in the decision procedure used to decide which operator to apply at a particular state, is empirically demonstrated. The model provides an analytic account for both of these sources of the power law. The model also predicts a problem-solving fan-effect, slowdown during practice caused by an increase in the difficulty of making useful decisions between possible paths, which is also found empirically.

THE POWER LAW OF PRACTICE (*1*), one of the few solid psychological learning phenomena, states that the speed of performance of a task will increase as a power of the number of times the task is performed. In one model, problem solving can be viewed as the search for a path through a directed "problem-space" graph, where nodes represent states of the problem or facts in memory and edges represent operators that move between states (*2*). Solving the problem involves finding a path from the initial state to the goal state by means of the available operators. Learning in this model corresponds to changes in either the specific topology of the graph or the decision procedure used to decide which operator to apply at a particular step when there is more than one edge emanating from a node. Many sorts of changes in method and operators can be modeled as changes in the topology of the problem-space graph, including restructuring and method selection. In this report we use computer experiments to show that this learning model exhibits the power law and that the phenomenon can be explained analytically by a theory based on graph dynamics. Our theory further predicts a problem-solving "fan-effect" in which performance becomes slower as more operators are learned in certain situations (*3*). This prediction is also empirically validated by our simulations.

Xerox PARC, Palo Alto, CA 94304.

The simple problem that we will use to explore learning phenomena, the "bit game," is analogous to many real problems. A problem state in the bit game is a *B*-bit binary vector (such as 01010). For the sake of concreteness we will use a 5-bit vector (*B* = 5) in most cases. A "trial" begins with an arbitrary initial state, say 00000. The player (a computer) searches for some other arbitrary vector (the goal state), say 11111, by successively applying operators that change the contents of the state vector. Each operator is composed of 1 to *B* elements indicating a particular bit in the vector that should be flipped if it matches in the current state. Operators specify only the bits in the state that actually change and can be written as "pattern → result" pairs, with question marks (?) where the operator pattern says nothing about a particular bit position. For instance, the operator ?1?1? → ?0?0? will take the state 11010 to 10000 or the state 11111 to 10101 but will not apply to the state 00000 because the bits indicated in the pattern do not match this state. As a result of the question mark "don't care" bits, operators vary in their generality. For instance, each of the two-element operators, such as 0?1?? → 1?0??, apply to eight different states (in this case 00100, 00101, 00110, 00111, 01100, 01101, 01110, and 01111).

We begin playing a particular bit game with all of the (2*B*) 1-bit operators (10, in the case of a 5-bit game). This set forms a *B*-dimensional hypercube and ensures that

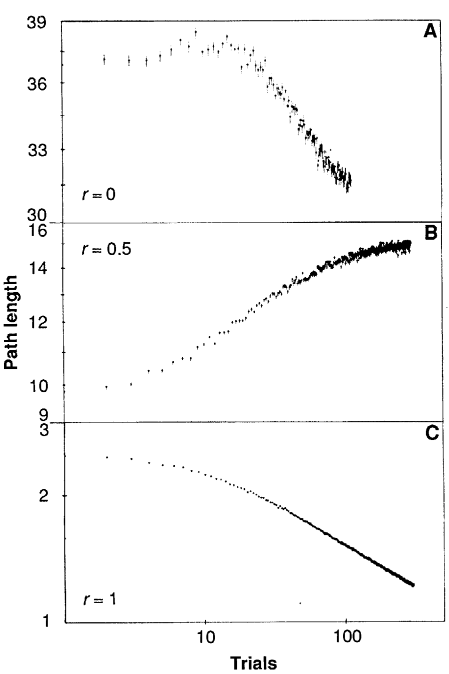

Fig. 1. Log-log plots of solution rates for the 5-bit game as a function of the number of trials: (**A**) the random walk; (**B**) mediocre decision procedure; (**C**) optimal decision procedure. All points are averaged over 16,384 observations.

there is at least one path between any two nodes. When there are several applicable operators for a particular state, a decision procedure is required in order to choose among them. The number of edges traversed on a trial (that is, the number of steps required to find the goal state from the initial state) measures performance. For simplicity, we assume that all operators contribute equally to performance, although operators with variable cost could be used to model problems of specific sorts. A series of trials, beginning with a common initial problem-space and with learning between each trial, will be called a problem-solving "run."

We separately simulated operator addition and decision-procedure improvement in the bit game. Operator learning takes place after each trial is completed in the style of SOAR chunking (*4*). Specifically, we add the operator that most generally summarizes the solution obtained in the trial. For instance, suppose we begin with the game, 01110 → 10101, and find a solution. The "subproblem solving" for this game is summarized by adding the operator that solves this particular game in one step. In this case the new operator is 01?10 → 10?01. Notice that the ? element of this operator appears because the third bit did not change between the initial and goal state in this trial, and so this new operator connects two pairs of states in the problem-space, that is, it adds two (directed) edges to the problem-space graph.

Decision procedures can be arbitrarily complicated algorithms with changes leading to entirely different problem-solving behavior. For a given organization of operators and choice of initial and goal states, certain decision procedures will be more effective than others. Decision procedures generally change radically only in the face of some new insight into the problem structure. Without such an extreme change it only makes sense either to slowly vary the parameters controlling the decision procedure in order to try to hill-climb into a best solution mode or to vary them randomly, hoping to discover a good decision procedure serendipitously. We consider changing between a "poor" decision procedure and an "optimal" one. The optimal decision procedure finds the fastest way to the goal. On the other hand, the poor decision procedure is a random walk. As operators are added to the problem-space through learning, it becomes more densely connected. Hence one has to do less searching to find a path leading to the goal, but it is also easier to get off the path.

To explore the range of decision procedures that lie between optimal problem solving and a random walk, we use a simple descriptive model of the effectiveness of the decision procedure in which, at any node during the search for the goal, each unproductive edge is eliminated with probability r. Improvements in the decision procedure correspond to an increase in r and change the problem from an exponential random search to a linear drift toward the goal. Note that $r = 1$ corresponds to a perfect decision procedure in which search and backtracking are never required, whereas $r = 0$ corresponds to a random walk on the graph.

In order to implement a decision procedure incorporating this parameter for the bit game, we first find all applicable operators from the current state and then order them by asking for each operator how many bits would be correctly set (for the desired goal state) if this operator were actually applied. We then separate these into the "good" ones (those that minimize the Hamming distance to the goal) and all the remaining "bad" ones (those that do not minimize this distance). Next each operator from the bad set is removed from consideration with probability r. Finally, we choose one operator at random from the union of the remainder of the bad set and all the optimal operators. When $r = 1$, all of the bad operators will be deleted, leaving only the good ones. When $r = 0$, all of the bad operators are left in the set, making the decision procedure a random search. It is important to note that even when $r = 1$, this implementation is only heuristic—it only approximates optimal problem solving. Actual optimal solution paths can only be found by exhaustively exploring the graph beforehand, a very lengthy computation. However, this heuristic comes very close to the optimal path (with $r = 1$) in the bit game.

Figure 1 shows how operator learning affects performance of the 5-bit game in the three main decision-procedure effectiveness regimes: optimal (Fig. 1C, $r = 1$), mediocre (Fig. 1B, $r = 0.5$), and random (Fig. 1A, $r = 0$). In all cases we randomly chose a start and goal state, solved the problem according to the indicated decision procedure, and recorded the performance. Recall that learning takes place after each trial by adding the operator that most generally summarizes the solution path just found. The possible games are uniformly distributed among the 2^{2B} bit configurations. All experiments were run on a 16384-processor Thinking Machines Connection Machine (CM-2).

Notice, first, that the results in Fig. 1, A and C, are consistent with the power law—they appear approximately straight on a log-log plot. More interesting, however, is the fact that adding edges improves performance in the optimal and random cases but initially degrades performance in the regime of the mediocre decision procedure. This is the problem-solving fan-effect, wherein learning hurts performance rather than helps it. Nevertheless, the absolute performance ranges are, as expected, best for the optimal strategy, worst for the random strategy, and medium for the mediocre strategy. Thus, in order to actually improve as a result of learning, one must start with a moderately good decision procedure or else, as learning takes place, one must improve the decision procedure in addition to learning new operators.

The power law and fan-effect are observed in many situations (*1, 3, 5, 6*) although the quantitative details of their forms will differ for each different task. In order to understand the general nature of these phenomena, we now show how learning that results from the addition of edges in a problem-space graph or improvements in the decision procedure lead in some cases to a gradual reduction path length with a corresponding gradual improvement in performance that is a power of the number of trials (the power laws), and in other cases to a gradual increase in path length (the fan-effect). Recall that a problem-space can be modeled as a graph with n nodes representing various problem states and with edges representing instances of possible operators. We are interested in the learning behavior for situations involving a large number of states and typical problem-spaces rather than any specific one (*7*). This leads us to consider typical examples of the class of all problem-spaces of a given size, namely, random graphs, where the initial operators are distributed at random and where new edges are added independently of one another. When large graphs are involved, this model is mathematically equivalent (*8*) to one in which every edge between a pair of states exists with independent probability p. As new operators are learned during the trials, p will correspondingly increase. In order that all nodes are almost surely reachable from any given node, p should be greater than $(\ln n)/n$ (*8*).

For this model we want to obtain an expression relating the expected number of steps, s, required to obtain a solution, to the values of r and p. We make a number of simplifications to the model which nevertheless retain its essential features. First, we assume that all nodes of the graph have the average number of links: $\mu = (n - 1)p$. Thus we are left with a regular graph consisting of n nodes with uniform branching ratio μ. Second, we assume that the cycles in the graph are long so that, in general, at any node there will be one edge that is one step closer to the goal while the others are one step farther away. In this limit, which applies when the graph is sparsely connected, the behavior will be similar to a walk on a tree. Because of the initially exponential growth in the number of nodes with distance, the initial and goal nodes will usually be separated by the diameter of the graph, which can be approximated as $D = \ln n/\ln \mu$.

With these approximations, at each node there is only one choice that gets closer to the goal state and $\mu - 1$ choices that move farther from the goal. However, the decision procedure eliminates each incorrect choice with probability r so there are (on average) effectively only $(\mu - 1)(1 - r)$ incorrect choices at each node. Thus the problem reduces to a bounded, one-dimensional random walk in which one starts at distance D from the goal and moves randomly until the goal is reached. This motion is constrained to remain within distance D of the goal and, at each step, to move toward the goal with probability

$$p_- = \frac{1}{1 + (\mu - 1)(1 - r)} \tag{1}$$

and away with probability $p_+ = 1 - p_-$. The average time required to reach the goal can be derived by standard techniques (*9*). It is given by

$$s = \frac{2}{(2p_- - 1)^2}\left[p_-^2 + p_-(D - 1) - (D/2) + \frac{(1 - p_-)^{D+1}}{p_-^{D-1}}\right] \tag{2}$$

This provides an explicit form for the expected behavior of s as a function of p (topology) and r (decision effectiveness) because the values appearing in Eq. 2, p_- and D, are expressed in terms of these two basic parameters.

When p_- is greater than p_+, the dominant behavior for large D is a drift toward the goal so that

$$s \approx \frac{D}{p_- - p_+} \tag{3}$$

When $p_+ = p_-$, Eq. 2 gives $s = D^2$, corresponding to symmetric diffusion. Finally, when p_+ is greater than p_-,

$$s \approx 2\left(\frac{1 - p_-}{p_-}\right)^{D-1}\left(\frac{1 - p_-}{1 - 2p_-}\right)^2 \tag{4}$$

which grows exponentially with D. We thus see a dramatic change in the nature of the search process as p_- passes through the critical value of 0.5.

Equation 2 produces the range of behaviors observed in the bit game experiments.

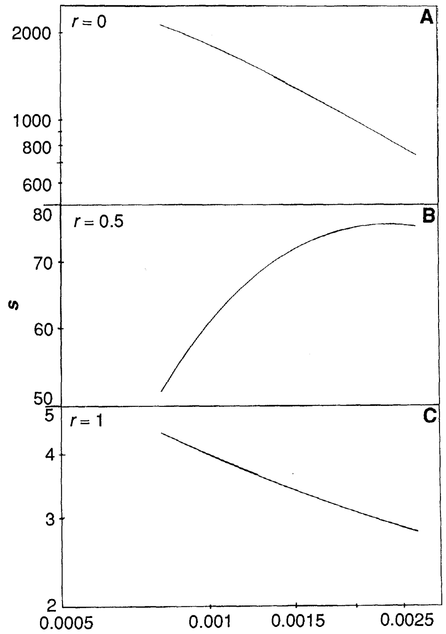

Fig. 2. Log-log plots of the theoretical predictions of s versus p for various r values, from Eq. 2: (**A**) the random walk; (**B**) mediocre decision procedure; (**C**) optimal decision procedure. In all cases n = 10,000.

For instance, when the decision procedure is weak (r near 0), p_- will be small (μ is at least as large as $\ln n$) and roughly equal to $1/\mu(1 - r)$. When $\mu(1 - r) >> 1$, one obtains

$$s \approx 2\frac{n}{\mu}(1 - r)^{\ln n/\ln \mu - 1} \tag{5}$$

In this case, when $r = 0$ (choices made at random), increasing the number of edges reduces the time to solve the problem. However, when $r > 0$ [but $\mu(1 - r)$ remains much larger than unity], there is a range in which increasing the number of links will result in a gradual increase in s, the expected number of steps required to solve the problem, because the smaller diameter of the graph is more than balanced by the increased difficulty of choosing the correct operator from among the larger number of choices. Specifically, s will increase as links are added when $|\ln(1 - r)| \ln n > (\ln \mu)^2$, which holds when r is not too close to zero, and there are many nodes and not too many edges. Conversely, when the decision procedure is strong (r near 1), the system's behavior is governed by the drift behavior of Eq. 3. When $\mu(1 - r)$ is much less than 1, Eq. 3 gives

$$s \approx \ln n/\ln \mu \tag{6}$$

Thus as long as the decision procedure improves sufficiently fast as new links are added, one obtains a power-law decay in $\ln \mu$. Figure 2 shows the behavior of Eq. 2 for increasing numbers of links in the three important r value regimes. In all cases the path length decreases or increases, corresponding to the experimental behavior.

We have shown that by applying the theory of graph dynamics to a problem-space viewed as a graph, we can capture, explain, and experimentally demonstrate the power law and the problem-solving fan-effect. Our approach should be contrasted with other theories of the power law and fan-effects. First, Anderson's ACT* model (*5*) obtains the power law by a rule-strengthening mechanism that itself operates according to a power law. Second, our approach is more general than the approach of Rosenbloom (*1, 6*), whose account of the source of the power law is restricted to addition of operators resulting from the chunking of problem-solving subgoals. We also incorporate improvements that result from adding operators and improvements in the decision procedure. Furthermore, we predict a power law for any sort of operator addition (in the appropriate decision-procedure regimes), whereas Rosenbloom predicted power laws only in the case of subgoal chunking.

REFERENCES AND NOTES

1. J. R. Anderson [*Language, Memory, and Thought* (Earlbaum, Hillsdale, NJ, 1976)], J. R. Anderson [*The Architecture of Cognition* (Harvard Univ. Press, Cambridge, MA, 1983)], and A. N. Newell and P. Rosenbloom [in *Cognitive Skills and Their Acquisition*, J. R. Anderson, Ed. (Earlbaum, Hillsdale, NJ, 1981), pp. 1–55] are seminal theories of this classical observation due to E. R. F. W. Crossman [*Ergonomics* **2**, 153 (1959)] and G. S. Snoddy [*J. Appl. Psychol.* **10**, 1 (1926)].
2. A. N. Newell and H. A. Simon, *Human Problem Solving* (Prentice-Hall, New York, 1972).
3. S. Minton, in *Proceedings of the 9th International Joint Conference on Artificial Intelligence* (Morgan Kaufmann, San Mateo, CA, 1985), pp. 305–310; S. Markovitch and P. D. Scott, in *Proceedings of the Fifth International Conference on Machine Learning* (Morgan Kaufmann, San Mateo, CA, 1988), pp. 459–465. This is analogous to the memory fan-effect: J. R. Anderson, *Cognitive Psychol.* **6**, 451 (1974); E. E. Smith, N. Adams, D. Schorr, *ibid.* **10**, 438 (1978).
4. J. Laird *et al.*, *Mach. Learn.* **1**, 11 (1986); R. E. Korf, *Artif. Intell.* **26**, 35 (1985).
5. See J. R. Anderson, *The Architecture of Cognition* in (*1*).
6. P. Rosenbloom, in *Universal Subgoaling and Chunking*, J. Laird, P. Rosenbloom, A. N. Newell, Eds. (Kluwer Academic, Boston, 1986).
7. B. A. Huberman and T. Hogg, *Artif. Intell.* **33**, 155 (1987); T. Hogg and B. A. Huberman, *Phys. Rep.* **156**, 229 (1987).
8. B. Bollobas, *Random Graphs* (Academic Press, New York, 1982).
9. N. G. van Kampen, *Stochastic Processes in Physics and Chemistry* (North-Holland, Amsterdam, 1981).

10. We thank J. Kephart and F. Ritter for detailed discussion of these issues and for comments on the paper. S. Card, J. Laird, A. N. Newell, P. Rosenbloom, and S. Stornetta contributed to our thoughts on this research. We thank the reviewers of *Science* who also provided many helpful comments.

5 July 1988; accepted 26 August 1988