Journal of Experimental Psychology: General
1997, Vol. 126, No. 3, 288–311

0096-3445/97/$3.00

Bending the Power Law: A CMPL Theory of Strategy Shifts and the Automatization of Cognitive Skills

Timothy C. Rickard
University of California, San Diego

The shift with practice from use of generic, multistep problem-solving strategies to fast and relatively effortless memory-based strategies, was explored in 2 experiments using pseudoarithmetic tasks. A complete transition to the memory strategy occurred by about the 60th exposure to each problem. The power law of practice did not hold in the overall data for either the mean or the standard deviation of response latency, but it did hold within each strategy (algorithm or retrieval). Learning was highly specific to the practiced problems. These results constitute the 1st clear demonstration of a skill for which the power law does not apply overall. The results do not support the instance theory of automatization (G. D. Logan, 1988) but are consistent with an alternative component power laws (CMPL) theory that assumes that because of intrinsic attentional limitations, only 1 strategy can be executed at a time.

One of the fundamental processes of human skill acquisition is the strategy shift with practice from use of generic, multistep procedures to direct retrieval of answers from memory (Ashcraft, 1992; Lemaire & Siegler, 1995; Logan, 1988; Reder and Ritter, 1992; Rickard & Bourne, 1996; Siegler, 1988). Examples are numerous in both the natural environment and the laboratory. Foreign vocabulary learning (Crutcher, 1989), spelling (Siegler, 1986), acquisition of linguistic rules (Bourne, Healy, Rickard, & Parker, 1997; Healy & Sherrod, 1994), and visual numerosity judgments (Lassaline & Logan, 1993; Palmeri, 1997) can all reflect this type of strategy shift. Basic single-digit arithmetic is probably the most familiar example. During initial stages of learning, children often use counting procedures that can require 10 s or longer to execute. With sufficient practice, however, they learn to retrieve answers to individual problems directly from memory. By adulthood, the direct-retrieval strategy typically yields answers in about a second (Siegler, 1988).

This article evaluates two candidate accounts of adult skill acquisition, strategy shifting, and the development of automaticity in these and related skill domains, with a current focus on mental calculation.[1] The models under comparison make diametrically opposing claims about two fundamental properties of human information processing. The instance theory of automatization (Logan, 1988) claims that direct retrieval from memory is automatic and executable in parallel with a variety of other more complex processes, such as multistep procedures or algorithms. The alternative model developed in this article assumes that memory retrieval is strongly dependent on attention and that only one retrieval event can be completed at any given time. The theory thus precludes parallel completion (but not parallel initiation) of two or more memory-retrieval events and, by extension, of memory retrieval and a multistep algorithm in which memory retrieval is involved in one or more of the steps. An empirically grounded resolution of this issue is central to development of a complete model of memory and skill acquisition. A finding that direct retrieval and algorithmic strategies are executable in parallel in tasks like mental arithmetic suggests that a variety of other complex thought process might also be executable in parallel. Alternatively, demonstration that strategy execution is a one-at-a-time phenomenon establishes an important boundary condition on the extent and nature of parallel human information processing and highlights the importance of programmatic research that explores the mechanisms of strategy choice and the factors influencing their operation (e.g., Anderson, 1993; Lemaire & Siegler, 1995; Reder & Ritter, 1992).

A second difference between the instance theory and the alternative introduced in this article involves assumptions about memory representation. According to the instance theory, each problem-solving episode results in an independent record, an instance, and each instance completes independently from other instances during subsequent performance. The alternative model proposed later makes the opposing claim that the type of memory that is operating in

This research was conducted in partial fulfillment of doctoral degree requirements. Support was provided by Army Research Institute Grant MDA-903-93-K-0010 and through a postdoctoral fellowship from the Cognitive Neuroscience Section, Medical Neurology Branch, National Institutes of Health.

Special thanks to Lyle Bourne and Alice Healy for their many thoughtful comments. Special appreciation to Gordon Logan for his helpful comments.

Correspondence concerning this article should be addressed to Timothy C. Rickard, Department of Psychology, University of California, San Diego, 9500 Gilman Drive, La Jolla, California 92093-0109. Electronic mail may be sent via Internet to trickard@ucsd.edu.

[1] The theory of Lemaire and Siegler (1995) provides good accounts of similar strategy-shift phenomena in children's arithmetic. However, that model to date has not been applied to the aspects of performance that are the focus of this article, and thus its predictions are not treated in this section. However, I consider implications of the results for that model in the General Discussion.

skill domains is best understood as a prototype representation for each item, which extracts and stores aspects of performance episodes that are common across repetitions and that are crucial for subsequent skilled performance. The effect of practice is effectively to strengthen a prototype representation for each item. Although this second difference between the theories is not tested directly in this article, general empirical support for a model that assumes strengthening of a prototype at least establishes the viability of this alternative form to representation. I begin with a brief overview of the power law of practice, which plays a fundamental role in the empirical predictions of both models.

The Power Law of Practice

Power-function speedup with practice has been observed across a wide variety of tasks, including retrieval of facts from memory (Pirolli & Anderson, 1985; Rickard, Healy, & Bourne, 1994), repeating sentences (MacKay, 1982), proving geometry theorems (Neves & Anderson, 1981), learning editing routines (Moran, 1980), rolling cigars (Crossman, 1959), and evaluating logic circuits (Carlson, Sullivan, & Schneider, 1989). In fact, power-function speedup appears to be so ubiquitous that Newell and Rosenbloom (1981) conferred to it the status of a scientific law. The power function is a member of a large class of functions that predicts a negatively accelerating rate of speedup as a function of practice. That is, it predicts substantial speedup from trial to trial during early stages practice but progressively less speedup from trial to trial during later stages. In formal terms,

$$\mathrm{RT} = a + b(N + p)^{-c}, \qquad (1)$$

where RT is the response time required to do the task, N is the number of practice trials, and a, b, c, and p are parameters. The number of previous learning trials is represented by p. The term $b(N + p)^{-c}$ goes to zero as N goes to infinity, and thus the parameter a represents the asymptotic RT. The parameter b is the difference between the RT on the first trial and the RT at asymptote, and c is a rate parameter that determines how quickly the RT approaches asymptote. A simplified two-parameter version of the power function that ignores previous learning and the asymptote fits RT data extremely well in most circumstances (see Newell & Rosenbloom, 1981).

Equation 1 is linear when plotted in log–log coordinates provided that the asymptote is first subtracted. Thus,

$$\log(\mathrm{RT} - a) = \log(b) - c[\log(N - p)].$$

This log–log linearity can be a powerful diagnostic tool in evaluating how closely data conform to a power function. Often substantial and systematic deviations from linearity in log–log plots can be detected visually even when statistical regressions fits yields r^2 values of .95 or higher. Thus, in evaluating power-function fits to data, both statistical measures and visual inspections of log–log plots are of diagnostic value (Newell & Rosenbloom, 1981).[2]

The power law has been an important empirical constraint influencing the development of a variety of skill theories, including those of Anderson (1983, 1993), Cohen, Dunbar, and McClelland (1990), Logan (1988), MacKay (1982), and Newell and Rosenbloom (1981), and it is generally believed to hold for any task domain. There is, however, only limited empirical evidence that the law holds for tasks exhibiting a transition from algorithm to retrieval. Indeed, as I discuss later, the available data hint at the possibility that the law does not always hold in overall data for this task domain. One of the purposes of the current research is to collect new data that more decisively addresses this question.

The Instance Theory of Automatization

Logan's (1988) instance theory of automatization (see also Compton & Logan, 1991; Logan, 1990, 1992) incorporates three basic assumptions. First, it assumes that encoding into memory is an obligatory, unavoidable consequences of attention. Second, it assumes that retrieval from memory is an obligatory, unavoidable consequences of attention. Third, it assumes that each encounter with a stimulus is encoded, stored, and retrieved separately. This last assumption makes the theory an instance theory of memory, which contrasts it with a variety of strength-based theories of memory processes (e.g., Anderson, 1983; Cohen et al., 1990; MacKay, 1982).

Three additional assumptions allow for derivation of a quantitative model that can be applied directly to data from tasks exhibiting a transition from algorithm to retrieval (see Logan, 1988, for a detailed discussion). First, the algorithm and each memory instance are assumed to compete in parallel, and independently, on each trial. The process that finishes the race first controls the response. Second, each episode, or instance, has the same distribution of finishing times that does not change with practice. Third, the algorithm has a separate distribution of finishing times that does not change with practice. The memory strategy comes to dominate the race as practice proceeds because, as more memory episodes accrue, the probability that one of them will win the race steadily increases.

Using a combination of formal mathematical proofs and Monte Carlo simulations, Logan (1988) showed that the instance theory predicts that the speedup in RT, as well as the reduction in standard deviation (*SD*) follows a power function of practice and that the rate parameters for the speedup in RT and reduction in *SD* are the same. Expressed as equations, the instance theory's predictions for the RT and *SD* are

$$\mathrm{RT} = a_1 + b_1(N^{-c})$$

$$SD = a_2 + b_2(N^{-c}).$$

[2] Fitting on the log–log scale selectively attenuates large RTs, and thus the later practice trials are given greater weight than the early trials.

Logan (1988, Experiment 4) tested the instance theory using an alphabet arithmetic task. In this task, a problem is presented for verification (e.g., E + 5 = J, true or false?), and the answer is determined by whether or not the right-side letter corresponds to the letter "down the alphabet" from the left-side letter as indicated by the numerical addend. Thus, E + 5 = J is true. In the Logan (1988) experiment, each participant received 72 blocks of practice, across 12 sessions, on 10 true and 10 false problems at each of four levels of addend size (2, 3, 4, and 5), for a total of 80 problems per block. The instance theory fits to that data set were reasonably good overall, as shown in Figure 9 of Logan (1988). However, on closer examination, it is clear that the fits underestimate the RTs and *SD*s during the middle portion of practice and overestimate these values toward the end of practice. This trend is weak for addend sizes of 2 and 3 but is clear for addend sizes of 4 and 5. Logan (1988) acknowledged these deviations but argued that they do not constitute a serious problem for the instance theory for two reasons. First, no existing model of skill acquisition predicts the deviations (because all current theories predict power-function speedup), and thus evidence against the instance theory is also evidence against the other models. Second, some participants reported at the end of the experiment that they used special mnemonics to deal with the problems with addends of 5. Logan proposed that participants shifted to using mnemonics between the fourth and fifth sessions of practice and that the use of mnemonics resulted in more efficient, or more memorable, traces, with a faster associated-RT distribution. Logan (1988) suggested that a modified version of the instance theory that incorporates this assumption can account for deviations from the power functions observed in the addend 5 alphabet arithmetic data.

The Component Power Laws Theory

The instance theory is the first principled alternative to process-based approaches to automaticity (which assume that speedup with practice reflects essentially more efficient processing of a single strategy). The new theory proposed in this article does not take issue with Logan's (1988) fundamental insight that automaticity is, at least in some contexts, best understood in terms of a strategy shift from algorithm-based to memory-based performance. Rather, it differs with respect to important assumptions about the underlying processes and representations that mediate this transition. Central assumptions of the instance theory are that algorithm and retrieval strategies are executed in parallel and independently of one another and that memory consists of a set of independent instances. In this section, an alternative CMPL theory is introduced, which makes the contrasting claims that either the algorithm or retrieval strategies, but not both, are selected at the outset of each trial and that a prototype representation for each item is strengthened with practice. These two assumptions lead naturally to the unique predictions that the power law of practice does *not* hold in the overall data for either the RTs or the *SD*s, but does hold generally within each of the component strategies. The connectionist (in the simple sense of nodes with connections) simulation model described later is motivated largely as a sufficiency demonstration that the fundamental assumptions of strength-based learning and nonparallel strategy execution can indeed lead naturally to these predictions.

Architecture

The architecture of the CMPL simulation model is described in the context of the *pound arithmetic task* used in Experiment 1. Prior to practice, solving these problems requires execution of a simple three-step arithmetic algorithm. Consider for example the problem 4 # 17 = ?. As the first step of the algorithm, the left-side number is subtracted from the right-side number (17 − 4 = 13). Second, 1 is added to the result of Step 1 (13 + 1 = 14). In the third and final step, the result of Step 2 is added to the right-side number (17 + 14 = 31). A basic assumption of the model is that pound arithmetic and related algorithms are a string of purely sequential memory-retrieval events, in which each step of the algorithm is a single retrieval event. The direct retrieval strategy is also treated as a retrieval event that is qualitatively equivalent to the retrieval event associated with execution of one step of the algorithm. As described in detail later, the model claims that on every trial there is a competition between the first step of the algorithm and direct retrieval strategy. Strategy choice in the model boils down to a choice process between these two single-step retrievals. The CMPL model thus makes the fundamental assumption that two retrievals cannot be completed in parallel (note, however, that the model does assume that multiple candidates for retrieval are initially activated in parallel). There is independent evidence from other experimental paradigms in support of this claim (Carrier & Pashler, 1995).

The strategy-choice process, and subsequent execution of the first step of the algorithm or of the direct-retrieval strategy, is the focus of the model diagram in Figure 1 and of the immediately following discussion. Straightforward extension of the model to account for RTs and *SD*s for all steps of a multistep algorithm are discussed subsequently. The top node in Figure 1 represents a general goal for solving a problem. This node has excitatory connections with two nodes at the subgoal level, one for executing the first step of algorithm (a subtraction in this example) and another for executing a direct retrieval from memory. The two subgoal nodes in turn have excitatory connections to long-term memory nodes at the problem level for executing either a subtraction or the direct-retrieval strategy. Also connected to the problem-level nodes are inputs from the external stimuli. The model assumes that all nodes at the problem level that are consistent with some known interpretation of the attended information (i.e., the external stimuli and information in working memory) receive activation via this pathway, which is independent of activation received from nodes at the subgoal level. Thus, problem-level nodes receive both bottom-up or perceptually driven activation and top-down or goal-driven activation. There are

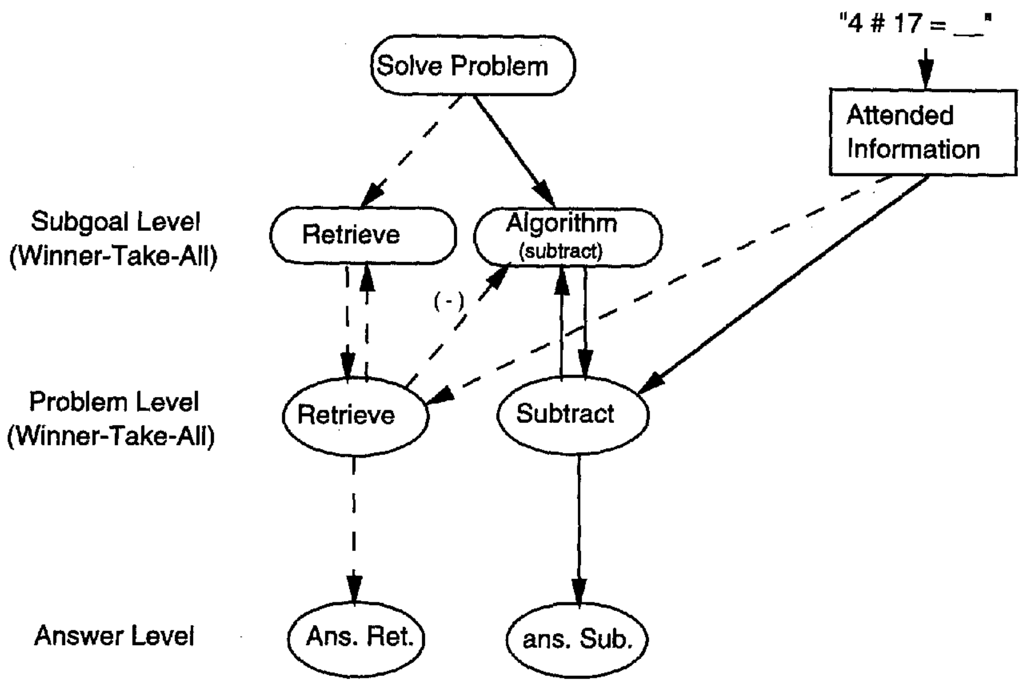

Figure 1. A diagram of the Component power laws network for the first step of the algorithm and the retrieval strategy for the pound arithmetic problem 4 # 17 = ?. Ans. Sub. = answer for subtract; Ans. Ret. = answer for retrieve.

also reverse excitatory connections from the problem nodes to the subgoal nodes. Thus, nodes as the subgoal level also receive both bottom-up and top-down activation. There are excitatory connections from each problem node to the corresponding answer nodes. There is also a single inhibitory connection from the direct memory-retrieval problem node to the algorithm first-step subgoal node. Finally, the model embodies a global, nonassociative winner-take-all inhibition that operates independently at the subgoal and problem levels and that is discussed in more detail later.

Learning and Process Assumptions

The necessary representations and connections for performing each step of the algorithm are assumed to exist at the outset of practice (solid lines in Figure 1). Strengths of these connections are assumed to take positive values that depend on the extent of previous experience with the component steps. Nodes and connections for the direct-retrieval strategy are assumed to be established on the first trial of practice for each problem (hatched lines). Connections among all nodes are assumed to increment in strength as a result of practice, according to the following rules. First, if the algorithm is the selected strategy on a given performance trial, all connections for both the algorithm and retrieval strategy are incremented for that problem. Second, if the retrieval strategy is selected, only strengths corresponding to the retrieval strategy are incremented. Finally, connection strength value, *st*, is assumed to be a negatively accelerating function of number of practice trials, *tr*, on which strengthening occurs of the form

$$st = 1 - c1^{tr^{c2}}. \qquad (2)$$

If the parameter $c1$ is set to a value just below 1.0 and $c2$ is set to a value of about 0.5, then *st* increases gradually from an initial value of 0 to a value of 1.0 with infinite practice.[3]

Activation of each node in the network is a function of input to that node from other nodes and of the number of cycles that the model has iterated (with synchronous updating) on a given performance trial. For a given target node of interest, *n*, this activation takes the form

$$a_{n,i} = 1 - [1 - \operatorname{sum}_j(st_{j,n} \times a_{j,i-1})]^i, \qquad (3)$$

[3] There are psychologically plausible mechanisms by which the strengthening function might take on the form of Equation 2. As one example, assume that some finite number of neural connections are available to be strengthened in support of any given association and that strengthening of any one of them follows Equation 2 with $c2$ set to 1.0 (i.e., strengthening of each connection is exponential). Also, assume that because of unspecified random factors, the strengthening value, $c1$, varies among the connections. Most connections strengthen very slowly, but a few strengthen quickly. Finally, assume that total strength is simply given by the sum of the strengths of the individual connections. Under these conditions, the overall strength as a function of practice can be closely approximated by Equation 2.

where $a_{n,i}$ is the activation of node n on Cycle i, j is a summation across all nodes in the network, $st_{j,n}$ is the strength of the connection between each node and the target node (self-connection strengths are all fixed at zero, and connections between all nodes that are not directly linked in Figure 1 have fixed strengths of zero), and $a_{j,i-1}$ is the activation of each node in the summation index on the cycle immediately preceding Cycle i. Thus, activation of any given target node of interest is an exponential function of the number of processing cycles, i, which is in turn modulated by the connection strengths and activation levels of nodes feeding into that target node.[4] As elaborated later, activation is further modulated by the winner-take-all competition at the subgoal level, which insures that one node within each level eventually reaches activation approaching 1.0 and that all other nodes within each level are suppressed to activation of zero.

Strategy-Selection Processes

Strategy selection in the model involves a dynamic interaction among nodes at the subgoal and problem levels. Initially, when strengths for the direct-retrieval strategy are weak, the subgoal for the algorithm first step (in the example in Figure 1, the subtract subgoal) and the corresponding problem node (in the example, the subtract problem node for 17 − 4) both reach the activation threshold at which the within-level winner-take-all inhibition sets (the *inhibition threshold*) first, forcing suppression of subgoal and problem nodes for the direct-retrieval strategy. Activation then continues to accumulate at the algorithm first-step answer node without competition from the direct-retrieval answer node.

With more practice, however, the connections strengths among nodes corresponding to the direct-retrieval strategy (including the inhibitory connection from the retrieval problem node to the algorithm subgoal node) become stronger. Eventually, this fact allows the retrieval subgoal to reach inhibition threshold first at the subgoal level, forcing suppression of algorithm subgoal activation. At this point the algorithm problem node no longer receives top-down activation from the algorithm subgoal node, thus placing it at a disadvantage relative to the retrieval problem node (which has both top-down and bottom-up input). In most cases, when the algorithm subgoal activation is suppressed, the retrieval problem node reaches the inhibition threshold for the problem level first, forcing suppression of the algorithm problem node. Activation of the direct-retrieval answer node then accumulates without competition from the algorithm answer node. Note that the inhibitory connection from the retrieval problem node to the algorithm first-step subgoal node has an important function of allowing the direct-retrieval strategy to win the competition even if the relevant strengths are weaker for that strategy.

In most cases, the winning problem node corresponds to the winning subgoal node (i.e., if the algorithm subgoal node wins, then the algorithm problem node also wins). However, in some cases, the retrieval node wins the competition at the subgoal level even through the algorithm (subtract) node still wins at the problem level. This effect occurs because the bottom-up input to the algorithm problem node (i.e., the input from the external stimuli) can in some cases be strong enough that the algorithm wins the competition at the problem level in spite of the bias against that node because of the absence of any top-down input from the subgoal level. In this unusual case, the model is designed to immediately shift the activated node at the subgoal level from the retrieval subgoal to the algorithm subgoal. This switching process is included because it is most natural to assume a cognitive system that avoids or corrects anomalous configurations such as concurrent activation of subgoal and problem nodes that do not match. If one assumes that setting and execution of a goal is a consciously accessible process, then this feature of the model predicts that participants occasionally set an initial goal to retrieve the answer but then experience a shift to the algorithm strategy because retrieval fails. This experience is in fact reported in informal protocols, especially on trials immediately preceding the initial retrieval trials for a given item. This is exactly the point during learning at which the CMPL model predicts problem node-driven subgoal shifting.

RT Assumptions for Memory Retrieval

By substituting Equation 2 into Equation 3, activation at the answer level for either the algorithm or the retrieval node can be expressed as

$$a_{\text{ans}} = 1 - [1 - a_{pi-1}(1 - c1^{(tr)^{c2}})]^i,$$

where a_{pi-1} is the activation, on the immediately preceding cycle, of the problem node that is connected to the answer node of interest. Solving for i, and replacing i with i_r, the number of cycles required for the activation of the answer node, a_{ans}, to an reach a response-threshold value, a_t, yields

$$i_t = \log(1 - a_t)/\log[1 - a_{pi-1}(1 - c1^{(tr)^{c2}})]. \quad (4)$$

A reasonable simplifying assumption for the moment is that activation of the problem node (a_{pi-1}) is approximately 1.0 when activation of the answer node reaches response threshold. Given this simplification, Equation 4 reduces to

$$i_t = \log(1 - a_t)/\log(c1^{(tr)^{c2}}),$$

which can be written as

$$i_t = [\log(1 - a_t)/\log(c1)](tr)^{-c2}.$$

Taking the log of both sides yields

$$\log(i_t) = \log[\log(1 - a_t)/\log(c1)] - c2[\log(tr)]. \quad (5)$$

[4] Equation 3 is only stable provided that sum $\{st_{j,n} \times a_{j,i-1}\}$ is less than 1.0. In the simulations reported later, this term never approaches this value.

Equation 5 is an exact power function, expressed in log–log terms, with an intercept of $\log[\log(1 - a_t)/\log(c1)]$ and a slope of $c2$.

Each processing cycle is assumed to correspond to some constant increment of real time. Thus, for a given retrieval event (either a step of the algorithm or direct retrieval of the answer), the CMPL model predicts power-function speedup with practice. Note that this prediction depends on the validity of the assumption that $a_{pi-1} = 1$ when $a_{\text{ans}} = a_t$. In practice, a_{pi-1} is slightly less than 1.0 at this point. Possible consequences of violating this assumption is addressed later in the *Simulations* section.

To model noise, the threshold-response value, a_t, is assumed to fluctuate as a beta distribution (Hogg & Craig, 1978) from trial to trial. The beta distribution has a domain between 0 and 1, thus assuring that the threshold never falls outside of the range of possible activation allowed in the model. A fluctuating response threshold is reasonable as a proxy for the effects of lapses of attention, varying levels of motivation, as well as a variety of other intrinsic noise effect.

Algorithm Assumptions

The algorithm is treated as a string of memory-retrieval events, each of which is qualitatively identical to the process of retrieving the answer directly. The guts of algorithm execution are thus identical to those of direct retrieval from memory. This treatment of algorithms is almost certainly still incomplete (see, for example, Carlson & Lundy, 1992), but as should be evident in the following discussion, it has some important predictive advantages over approaches that treat the algorithm as a distinct and undifferentiated process.

When activation of the answer node corresponding to the first step of the algorithm reaches response threshold, the model assumes that the attended information is updated to include this newly retrieved information. Also, any information that was attended to for execution of the first step of the algorithm but that is no longer needed for retrieval of the second step is assumed to be dropped from attentional focus. The retrieval event corresponding to the second and any subsequent steps of the algorithm then takes place analogously to that for the first step. The time to execute a multistep algorithm is assumed to be a simple additive function of the time needed to execute each component step. Note that mechanisms by which the system parses and selects new attended information on execution of each step of the algorithm are not explicitly accounted for the current model. Rather, it is simply assumed that the appropriate information is available in working memory to execute each step. Also, note that the model predicts that if the retrieval strategy does not win the competition on the first step of the algorithm, it does not win on any subsequent steps.

The sum of a series of power functions that all have identical rate parameters is another power function with the same rate parameter and a scaling parameter (the parameter b in Equation 1) that is the sum of the scaling parameters of the power functions for the individual steps of the algorithm. Thus, assuming that the rate-parameter values for the algorithm steps are very similar for each step, at least on average (see simulations given next), then the CMPL model predicts that the power function should describe speedup with practice not only for the retrieval strategy but also for the algorithm strategy.

An Empirically Motivated Constraint on the Values of c1 and c2

Equation 2 embodies two strength parameters, $c1$ and $c2$. Note that $c1$ is a unique component of the intercept, and $c2$ is the slope of the power-function prediction (in log–log coordination) for speedup in RT (Equation 5). There also turns out to be a strong positive correlation between the slope and intercept (larger intercepts correspond to steeper slopes) in log–log regression fits of the individual-item RT data for each participant within a given strategy. For example, these values correlate around .9 on average for both algorithm and direct retrieval data in Experiment 1 of this article.

Candidate models ultimately need to provide an account for this empirical relation between the intercept and slope at the item level. Although the CMPL model does not predict this correlation, the parameters $c1$ and $c2$ in the CMPL model provide a natural framework for accommodating it. Specifically, the CMPL model can be constrained such that

$$\log[\log(1 - a_t)/\log(c1)] = x1 + x2(c2),$$

where $\log\ [\log(1 - a_t)/\log(c1)]$ is the predicted log-RT intercept, $c2$ is the predicted log-RT slope, and $x1$ and $x2$ are slope and intercept parameters for the linear relation between the slope and intercept for the log RTs. Solving for $c1$ in terms of $c2$ yields

$$c1 = 10^{[\log(1-a_t)/10^{(x1+x2\times c2)}]}.$$

This constraint defining $c1$ in terms of $c2$, (the slope in log–log plots) is incorporated into the simulations discussed next.

Simulations

Four issues are addressed in the following simulations. First, does the fact that the activation level of the winning problem node tends to be slightly less than 1.0 when the activation level of the winning answer node reaches response threshold compromise the strategy-specific power-function predictions of the model? Second, does the strategy-choice mechanism generate a strategy shift from algorithm to retrieval, and does the model produce within-strategy power-function speedup even in the context of a strategy shift? Third, does reduction in the *SD* with practice within each strategy follow a power function and, if so, how do the parameters of the power function for the *SD*s relate to those for the RTs? Finally, does collapsing data across multiple items and participants that have different values for the learning-rate parameter, $c2$, in any way alter or compromise the quantitative predictions regarding either the strategy transition or the RTs and *SD*s of the component strategies?

Simulation for a single item. At the outset of practice, the connection strengths between the subgoal, problem, and answer nodes for the algorithm are assumed to have values greater than zero to represent previous learning effects. All other connection strengths are initially set to zero. Activations are set to zero at the outset of each trial, with the exception of the external information and the *solve problem nodes,* which are assumed to have constant activation levels of 1.0 throughout each trial. In the current version of the model, the global inhibition has a simple all-or-none property: When the more active of the subgoal nodes equals or exceeds a value of .3, the less active node is set to zero. Similarly, when the more active of the problem nodes equal or exceeds .6, then the less active problem node is set to zero. The response threshold, a_p, is set to fluctuate from trial to trial according to a beta distribution with parameters alpha = 16 and beta = 4. This distribution has a peak probability density at .8, and falls off sharply in both directions such that it rarely produces values below .7 or above .9.

To demonstrate basic properties of the model at the individual item level, 100 trials of practice on a single item were generated 1,000 times, and RTs and *SD*s were computed for each practice block. In this simulation, the learning-rate parameter, $c2$, was set to 0.5 for all connections, the scaling parameters for the relation between $c2$ and $c1$ were set to $x1 = 1$ and $x2 = 2.3$, respectively, and the previous learning, p, was set to 25 trials. A three-step algorithm was assumed. An algorithm with more (or fewer) steps yields larger or smaller algorithm RTs and *SD*s, but has no effect on either the number of trials necessary to make the transition to retrieval or on the quantitative results for the retrieval strategy itself.

The results are shown in Figure 2 for the RTs and Figure 3 for the SDs. The best fitting two-parameter power functions are also shown for the algorithm and retrieval RTs and *SD*s, which are presented in units of log cycles. For the algorithm fits, 25 trials of previous learning was assumed, in correspondence with the 25 trials of previous learning stipulated in the simulations. For this sample item, the algorithm is selected for the first 39 trials, and memory retrieval is selected thereafter. The fits are essentially exact, confirming the component power-law predictions at the item level for both the RTs and *SD*s in the context of a strategy shift. Note that although the algorithm data follow a power function, they are not linear in the log–log plot. This effect is due to the previous learning of 25 trials and is to be expected for the algorithm strategy according to the CMPL model. Note also that the rate-parameter estimates ($c2$) for he RT and *SD* fits within each strategy are identical, corresponding almost exactly to the value of 0.5, which was also the learning parameter selected for this simulation. This finding provides further confirmation that the actual running simulation conforms very closely to the idealized mathematical derivation at the individual item level.

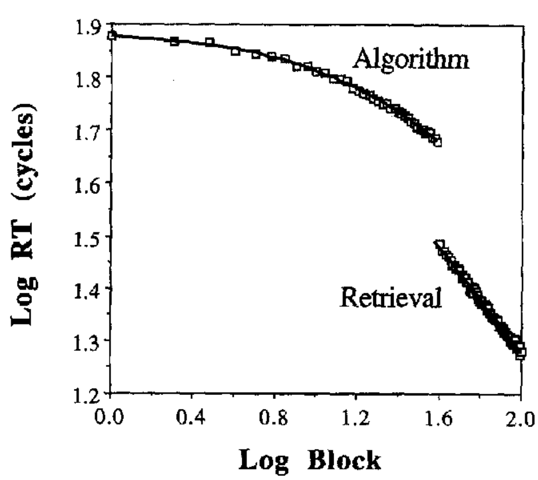

Figure 2. Simulation results for response times (RTs) for 1,000 items across 100 trials in which each item has identical parameter values.

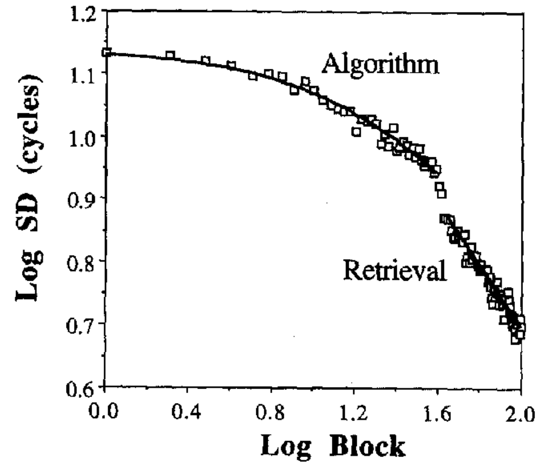

Figure 3. Simulation results for standard deviations (*SD*s) for 1,000 items across 100 trials in which each item has identical parameter values.

Simulation for multiple items and participants. A second simulation was performed to explore possible distortions caused by collapsing data over items and participants with differing values for the learning parameter, $c2$. An experiment with 18 participants, each of whom practices on a set of 12 problems, was simulated (the conditions of Experiment 1). The same parameter settings were used as for the single-items simulation, with the exception of the learning-rate parameter, $c2$, which was generated for each item from a beta distribution with the alpha and beta parameters both equal to 5. This distribution yields an expected value for $c2$ of .5, with most observations occurring between the range of .3 to .7. To compute RTs for each strategy, the results for each item for each simulated participant were logged, and then data were averaged over items and then over participant. *SD*s for each strategy were computed first for each participant and were then logged and averaged across participants.[5]

The strategy-transition results are shown in Figure 4. The first strategy transition took place on Block 3, and the last transition occurred on Block 69. As shown in Figures 5 and 6, the component power-function predictions hold to a close approximation across most of the training interval for both the RTs and *SD*s. However, there are three important differences between the multi-item and single-item results. First, the rate-parameter values for the RT and *SD* within each strategy are no longer identical in the multiple-item simulation. Rather, for both strategies, the rate parameter takes slightly larger values for *SD*s than for the RTs. This effect reflects different results of collapsing over multiple items on RTs and *SD*s. In this simulation, there is a between-items RT difference that results in a between-items component of the *SD*, which was not present in the single-items fits. Because of the linear constraint-relating intercept and slope for the log RTs that is incorporated into the model, this between-items component of the *SD* decreases with practice. Thus, the decrease in *SD* with practice in the multi-item simulations reflects both the intrinsic within-item component, which follows an exact power function, plus an additional between-items component, which is also an approximate power function across the range of practice simulated but which decreases at a faster rate than does the within-items component. In combination, these two components of the *SD* result in a reduction in *SD* with practice that is still nearly an exact power function but that always has a rate-parameter value that is slightly greater than that for the corresponding RTs within the same strategy.

Second, for both the RT and *SD* for the retrieval strategy, there is a concave downward deviation of the data from log − log linearity during roughly the first half of the strategy-transition interval. Because of these distortions, power-function fits for the retrieval strategy shown in the figure are limited to trials beyond the halfway point of the practice interval during which the strategy transition occurred. This point is indicated by the change from solid to hatched lines in the retrieval fit. The hatched line is simply an extrapolation from the solid-line fit. The algorithm fit is also based only on the data prior to the halfway point of the strategy transition.

For the RTs, the concave downward deviation from the power-function fit for the retrieval strategy reflects the fact that items with higher retrieval strengths initially shift to retrieval earliest, and they also have the fastest RTs (because RT is a direct function of strength). Thus, although the power function holds at the item level within each strategy, the average of the retrieval trials during roughly the first half of the strategy-transition interval has faster RTs than would be predicted on the basis of an extrapolation of the power-function fit to the average of the retrieval trials after the midpoint of the strategy transition. The power function holds for the average algorithm data for almost the

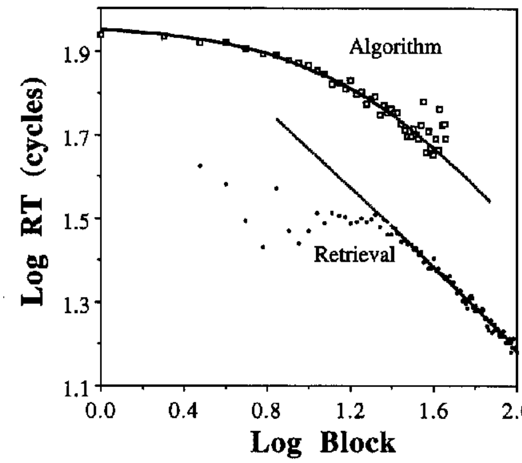

Figure 5. Results for response times (RTs) for both the algorithm and retrieval strategies for a simulation of 12 items for each of 18 participants over 100 blocks of practice in which the value of the parameter *c*2 varies according to a beta distribution with mean of .5.

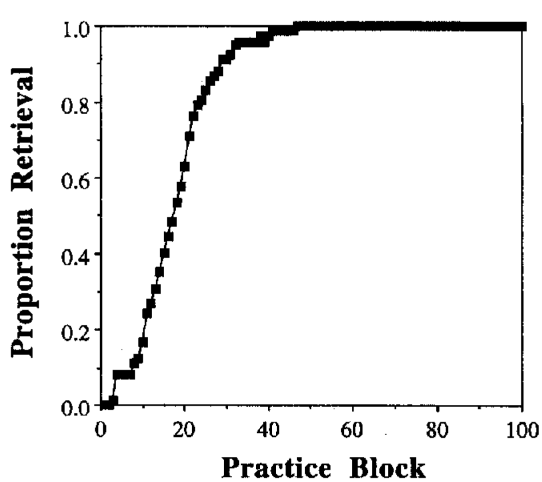

Figure 4. Results for proportion of retrieval trials for a simulation of 12 items for each of 18 participants over 100 blocks of practice in which the value of the parameter *c*2 varies according to a beta distribution with mean of .5.

[5] Logging at the individual-items level and then averaging is mathematically equivalent to taking the geometric mean (i.e., multiplying at the item level and taking the *n*th root, where *n* is the number of items) and then taking the log of the result. It is possible to prove that if the power function holds for the individual-item data, it also holds for the geometric mean of the data (provided that asymptote effects on the fits are negligible and can be ignored). In contrast, taking the arithmetic mean and then logging does not guarantee a power function except in the unrealistic special case in which the rate parameter for each individual item is identical. However, note that Wixted (personal communication, January 9, 1997) demonstrated through simulation that arithmetic averaging nevertheless preserves the power-function form almost exactly, provided (a) there are no strong ceiling or floor effects in the data, (b) there are more than just a few observations being averaged, and (c) the distribution of parameter values of the averaged power functions do not have extremely large variance (a condition that probably holds in most real data sets).

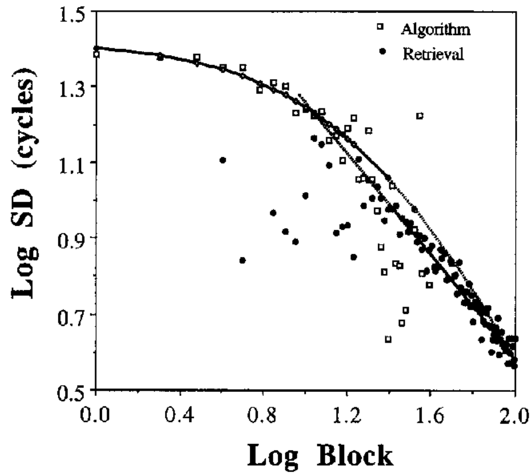

Figure 6. Results for standard deviations (*SD*s) for both the algorithm and retrieval strategies for a simulation of 12 items for each of 18 participants over 100 blocks of practice in which the value of the parameter $c2$ varies according to a beta distribution with mean of .5.

entire simulated practice interval. Only for the last few algorithm trials is there a slight trend for the regression model to underpredict the simulated algorithm data.

The analogous concave downward deviation from log–log linearity for the *SD*s is also a result of the fact that higher strength items make the transition to retrieval first. Recall that the within-items *SD* decreases as a power function of practice (see *Simulation for a single item*). This decrease is a direct result of corresponding increases in strength with practice. Thus, if higher strength items make the transition to retrieval first, the within-items (as well as the between-items) component of the *SD* for those items is smaller than would be expected if all items were solved by retrieval at that same point during practice.

A Regression Model Based on the CMPL Simulations

The simulations validate a relatively simple mathematical version of the CMPL model that can be fit to data using standard least squares regression techniques. The regression model embodies the following constraints:

1. The power function holds for both the RTs and *SD*s for the algorithm across essentially the entire practice interval. A different power function holds for the RTs and *SD*s for the retrieval strategy after about the halfway point of the transition interval.
2. The previous learning parameter of the generalized power function (Equation 1) can take positive values for the algorithm for both the RTs and *SD*s. Previous learning should be set to zero for the retrieval RTs and *SD*s. Even though asymptotes for RTs must be in principle some positive value, they can be assumed to be zero with no detectable decrement in quality of fit, given the reasonable assumption (for most experimental tasks) that the RTs do not approach asymptote within the practice interval over which data are collected (see Newell & Rosenbloom, 1981).
3. Rate parameters of the best fitting power functions for the RT are smaller than those for the *SD* for both the algorithm and the retrieval strategies.

These predictions are expressed in the following equations and inequalities:

$$\text{RTalg} = b1(tr + p)^{-k1}$$

$$SD\text{alg} = b2(tr + p)^{-k2}$$

$$\text{RTret} = b3(tr)^{-k3}$$

$$SD\text{ret} = b4(tr)^{k4}$$

$$k1 < k2$$

$$k3 < k4,$$

where $b1$ through $b4$ are intercept parameters, $k1$ through $k4$ are rate parameters, and p is the amount of previous learning for the algorithm.

Finally, note that, according to the CMPL model, overall RT for a given practice block is governed by the mixture equation $\text{RT} = P(\text{RT}_{\text{algorithm}}) + (1 - P)(\text{RT}_{\text{retrieval}})$, where $\text{RT}_{\text{algorithm}}$ and $\text{RT}_{\text{retrieval}}$ are the practice functions describing the algorithm and retrieval means for that practice block and P is the proportion of trials on the which algorithm is selected. Similarly, the overall variance on a given practice block is governed by $\text{VAR} = P(\text{VAR}_{\text{algorithm}}) + (1 - P)(\text{VAR}_{\text{retrieval}}) + P(1 - P)(\text{RT}_{\text{algorithm}} - \text{RT}_{\text{retrieval}})^2$. However, the CMPL currently does not strongly constrain P as a function of practice block (because the parameter $c2$ could in fact vary according to any number of possible distributions), and thus it is not possible to fit the model to the overall RT or the overall variance (or *SD*). For this reason, the simulation results as well as fits of the model to data are described solely in terms of RTs and *SD*s for each separate strategy.

Experiment 1

A pseudoarithmetic task, pound arithmetic, was used to provide a direct empirical comparison of the two models. Two types of pound arithmetic problems were constructed using a simple arithmetic series in which the third element of the series is the difference between the first two elements, plus 1, added to the second element. For example, the third element of the specific number sequence 9, 15, ?, is [(15 − 9) + 1] + 15 = 22. In Type 1 problems, the third element of the series was unknown (e.g., 9 # 15 = __). In Type 2 problems, the second element of the series was unknown (e.g., 9 # __ = 22). Problems were presented in a traditional arithmetic format (as in the example above) with a blank holding the place of the missing element, and with the # symbol used to hold the place of the arithmetic symbol. Participants were taught a three-step algorithm, as shown above, for solving Type 1 problems and a related four-step algorithm for solving Type 2 problems.

After the practice phase, participants were tested on the exact problems seen during practice (no-change problems), on type-change problems (i.e., a Type 1 problem seen during practice was presented as a Type 2 problem at test), and on new problems not seen during practice. The type-change problems at test allow exploration of the specificity of the problem representation that is formed during practice. As is evident in Figure 1, the CMPL model assumes a unidirectional association between problem and answer nodes and thus predicts that learning that occurs during the practice phase should not transfer to new problems or even to type-change problems, at test. A comparison of the new problems condition at test with performance at the beginning of practice should also allow determination of whether there was any general algorithm speedup during practice that was not directly related to speedup on individual problems. General speedup in the algorithm is not predicted by the CMPL model (although problem-specific speedup clearly is predicted). Neither general nor problem-specific algorithm speedup is accommodated by a strict interpretation of the instance theory as developed in Logan (1988).

Method

Participants. Twenty-one participants from an introductory psychology course participated in the experiment for credit. Two of these participants were dropped because they failed to attend all of the practice sessions. An additional participant's strategy-probing data revealed that no transition to retrieval occurred during the course of practice. Thus, a total of 18 participants attended all sessions and showed a transition to retrieval with practice. The data from the single no-transition participant were also preserved for separate analysis. All participants were tested on IBM-type personal computers, programmed with the Micro Experimental Language (MEL) software (Schneider, 1988).

Apparatus and materials. Three subsets of 6 pound arithmetic problems were constructed. Within each subset, there was 1 problem with each of six left-side numbers (3–8), and there was at most 1 problem with each of nine middle numbers (11–19), and at most 1 problem with each of 18 right-side numbers (18–35). Three master sets of 12 problems were then created, one from each of the two-way combinations of the three subsets of six. Six experimental problem sets were then created, two from each master set (see Appendix A). One of the two problem sets created from each master set had one subset of 6 problems written as Type 1 problems (e.g., 4 # 17 = __), and the other subset was written as Type 2 problems (e.g., 3 # __ = 36). The other problem set reversed the problem types (e.g., a Type 1 problem became a Type 2 problem). Each participant solved problems from only one experimental problem set during practice. Thus, each participant saw 12 problems during practice, 6 Type 1 problems, and 6 Type 2 problems. Either three or four participants were given practice on each of the six problem sets. During subsequent immediate and delayed transfer tests, all participants solved all 18 problems presented as both Type 1 problems and Type 2 problems.

Procedure. The experiment involved six sessions, the first three on Monday, Wednesday, and Friday of 1 week, two additional sessions on Monday and Wednesday of the following week, and a final session on the Wednesday 6 weeks after the fifth session. Each session lasted 40 to 60 min. Participants were tested in groups of up to four. At the beginning of the first session, the participants were given an example sheet describing the algorithms for Type 1 and Type 2 problems and an example problem worked out step by step for each problem type. The experimenter worked these example problems on a blackboard, with the participants following along using the example sheet. The participants were then given six problems (three Type 1 problems and three Type 2 problems) to work independently using paper and pencil (these problems were different than those used in the main experiment). When the participants completed the problems, the experimenter checked the results for accuracy and made corrections where necessary, making it clear to the participant what the errors were and what they should do differently to correct them. From this point on, participants performed the task independently at their own computer without the benefit of pencil or paper, although they were allowed to take the algorithm sheet with the example problems with them to the computers. For the remainder of the first session, participants performed nine blocks of problems using the computer, where each block was one exposure to each of the 12 problems, randomly ordered, in the participant's practice set. Problems were presented one at a time in the middle of the screen. Participants entered the two-digit answer using a number keypad on the right-hand side of the computer keyboard. They were instructed to work as fast as possible while being accurate. They were told that they could rest briefly between blocks of problems. Latencies were collected from the onset of the problem to the pressing of the first digit of the answer (the initiate RT) and from the pressing of the first digit of the answer to the pressing of the second digit of the answer.

Following one third of the problems, participants were probed for the strategy that they used. On these trials, a screen with three options was displayed below the problem after they pressed the second digit of the answer. The options instructed the participant to press a special key marked *A* if they used the algorithm that they were taught to solve the problem, to press a key marked *R* if they retrieved the answer directly from memory (retrieval of 2 × 4 = 8 was used as an example of what was meant by direct retrieval), and to press a key marked *O* if they used some other strategy that did not correspond closely to either of the other options. Across every set of three consecutive blocks, each problem was probed once. Four problems were probed per block. Problems probed on each block were randomly determined, subject to the preceding constraints. The participant's strategy response, as well as the latency from the onset of the strategy options screen to the pressing of the response, were collected.

The second, third, fourth, and fifth sessions consisted of 15, 21, 24, and 21 blocks of problems, respectively, presented on the computers as described previously. The transfer test was given immediately after the practice segment of the fifth session. The test consisted of 3 blocks, each block consisting of one exposure to each of the 18 problems shown as both types, for a total of 36 problems per block. Thus, there were three test conditions for both Type 1 and Type 2 problems: a no-change condition; a type-change condition, in which a Type 1 problem during practice (e.g., 4 # 17 = __) was presented as a Type 2 problem (e.g., 4 # __ = 31), and the reverse; and a new problems condition in which number combinations not seen during practice were presented. During the test, participants were probed after every problem, in the manner described above. The delayed-transfer test was given during the sixth session and was exactly the same as the immediate-transfer test, with the exception that no additional practice was given prior to the delayed test.

Results and Discussion

Primary analyses included only the 18 participants who reported a strategy transition with practice. Results for the single no-transition participant are discussed at a later point. Results for Type 1 and Type 2 problems were remarkably similar. There were no reliable problem-type differences in terms of error rate, rate of transition to retrieval, or RTs. Thus, all analyses reported later were collapsed across this variable. Overall error rates were .109, .065, .055, .029, and .019 in Sessions 1, 2, 3, 4, and 5, respectively. A within-subjects analysis of variance (ANOVA) with a single factor of session (1–5) indicated a reliable decrease in error rate across session, $F(4, 17) = 11.1$, $p < .001$. All subsequent analyses were performed on data from correctly solved problems.

The strategy-probing results are shown in Figure 7, collapsed over participants and problems and over consecutive three-block sequences across which each problem was probed once. Practice was successful in creating a transition to retrieval. By about Block 60, retrieval was the reported strategy on nearly all trials. There were relatively few "other" responses, a result that is consistent with the CMPL prediction that pure algorithm and retrieval strategies were the only two strategies that are used in this task. For 90 of the 216 items across the 18 participants (41%), the transition was a step function. The algorithm was used for an initial number of trials, and retrieval was used exclusively thereafter. For the remaining items, the transition was not a step function, although in the majority of these cases there were very few blocks of practice between the first retrieval response and the last algorithm response.

Practice: Instance theory fits. Figure 8 shows the log RT and log *SD* averaged across participants and problems, plotted as a function of log block. Also shown in these figures are the best fitting power functions as predicted by

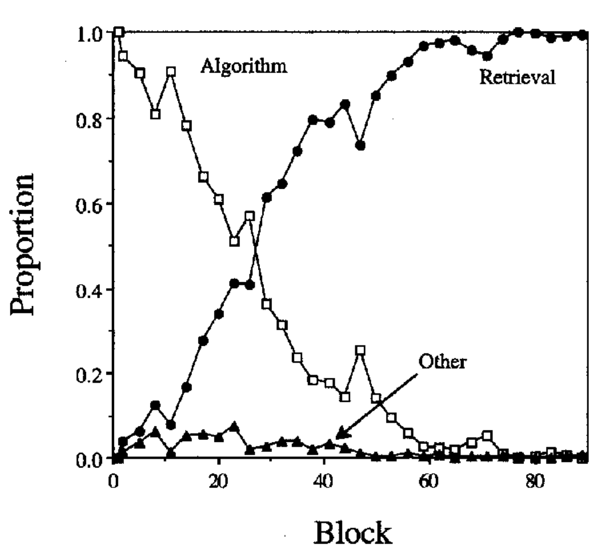

Figure 7. Proportion of algorithm, retrieval, and other responses as a function of practice in Experiment 1.

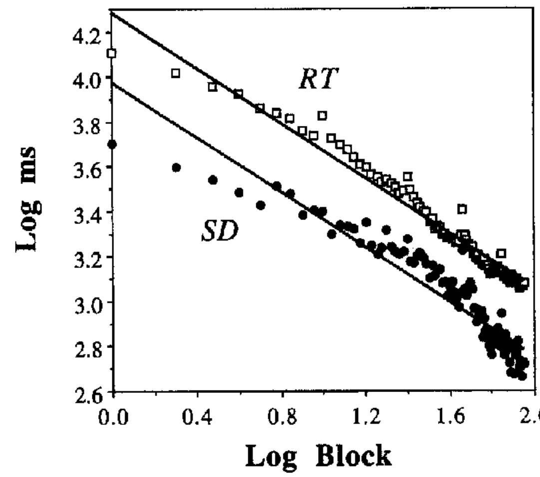

Figure 8. Instance theory fits to the response time (RT) and standard deviation (*SD*) data of Experiment 1.

the instance theory. Overall, r^2 for the combined RT and *SD* fits was .94. Note the systematic deviations of the observed from the predicted values for both the RT and *SD*, analogous to those observed by Logan (1988) for the alphabet arithmetic task. In the early stages of practice, the predicted values substantially overestimate the actual values (by a full second or more). During the middle stage of practice, the predicted values again overestimate the observed values. By the end of practice, the predicted values again overestimate the observed values. Also, as with Logan's (1988) alphabet arithmetic data, the deviations from linearity are more extreme for the *SD*s than for the RTs.

The instance theory prediction that the rate parameters for the RTs and *SD*s are the same (Logan, 1988) was tested by fitting three-parameter power functions, which included a parameter for the asymptote, separately to each participant's RT and *SD* data. Sixteen of the 18 participants showed steeper rate estimates for the RT ($M = -.505$) than for the *SD* ($M = -0.435$), a difference that is strongly reliable by a binomial sign test ($p < .01$), disconfirming this prediction of the theory (but see Logan & Etherton, 1994, for a discussion of conditions under which this prediction might not be expected to hold). Note that the CMPL model makes the prediction that the rate estimate for the *SD*s are greater than that for the RTs. However, this prediction applies only within each individual strategy. The result presented here for the overall data is thus not inconsistent with that model.

Practice: CMPL fits. Fitting the CMPL model requires a methodology that allows trials on which the algorithm was selected and trials on which retrieval was selected to be evaluated separately. In the CMPL model, one and only one of the strategies is selected for each trial. The model also assumes activation of a unique subgoal representing the selected strategy, and it is reasonable to assume that these representations are consciously accessible and reportable. Thus, the model predicts that participants should be able to reliably report which of the two strategies was employed on a given trial.

One approach to dividing the trials by strategy is to examine only the data on which strategy probes were collected. However, because strategy probing took place for a given item only once every three trials, this approach would eliminate two thirds of the data. An alternative is to take advantage of the fact that most items showed an abrupt strategy transition to retrieval as a way to group trials that were not probed into those that with a high probability involved the algorithm or with a high probability involved retrieval. Those trials can then be added to the data on which there were strategy probes. To provide a systematic and objective basis for making this grouping, a logistic function was fit to strategy-probe data separately for each item for each participant. The logistic function has the form

$$p(\text{ret}) = 1 - 1/\{1 + \exp[(BL - g)/h]\},$$

where p(ret) is the predicted probability that the retrieval strategy is used, and g and h are scaling parameters, and BL is the practice block. To fit logistic functions to the strategy data, strategy-probe results were coded with a value of 0 if "algorithm" or "other" response was given and with a value of 1 if a "retrieval" response was given. (The same analysis ignoring "other" responses yielded equivalent results.) Thus, for each item for each participant, the data consisted of up to 30 zeros and ones across 90 blocks of practice (the value was not always 30 because trials on which the response was incorrect were eliminated from the analysis).

The following filtering procedure was then employed for selecting algorithm trials. First, the practice block corresponding to predicted retrieval probabilities of .01 (*BL*min) were computed based on the logistic fits to each item. All trials that occurred before *BL*min for a given item were then categorized as algorithm trials, with the exception of a small number of trials on which the retrieval strategy was explicitly indicated by the strategy-probing data. For block values greater than *BL*min, only trials on which strategy probing directly showed that the algorithm was used were selected. The filter for retrieval trials was exactly analogous to that for algorithm trials, but in the reverse direction, such that all nonprobed trials that occurred after *BL*max (retrieval probabilities of .99) for a given item were categorized as retrieval trials, with the exception of a small number of trials on which the algorithm strategy was explicitly indicated by the strategy-probing data.

Figure 9 shows the results for RTs and Figure 10 shows the results for *SD*s, plotted in log–log coordinates. Best fitting CMPL functions are also shown for both strategies in both figures. The statistical fit for the algorithm trials was limited to trials prior to the halfway point of the transition interval, and the fit for retrieval trials was limited to those past the halfway point of the transition interval. The overall r^2 for the combined fit to the RT and *SD* data was .94, equivalent to that of the instance theory. It is important to note also that the systematic visual deviations from the power function that were clear in the instance theory fits to the overall data are no longer present when data are fit separately by strategy.

Two additional patterns in the data were consistent with

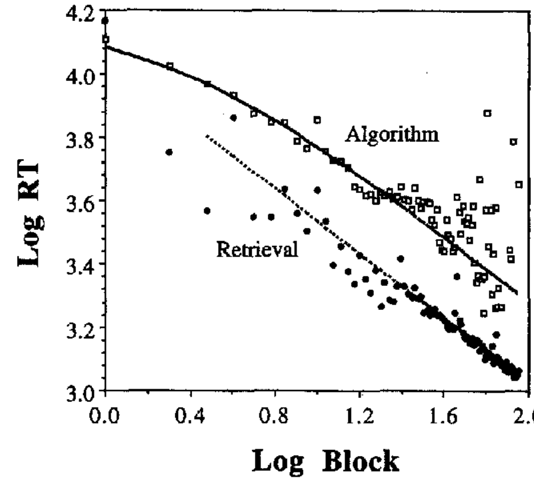

Figure 9. Component power law theory fits to the response time (RT) data for the algorithm and retrieval strategies in Experiment 1.

the predictions of the CMPL model. First, there was a slight concave downward deviation from the power functions fits in the RT and *SD* data for the retrieval strategy prior to about the halfway point of the strategy transition. Second, the rate estimates for the *SD* fits were greater for both the algorithm and retrieval strategies than they were for the corresponding RT fits, although this effect was not statistically reliable.

In summary, the CMPL model provides fits that are equivalent to those of the instance theory in terms of r^2 and superior in terms of the visual correspondence with the data. Further, the CMPL accomplishes these fits with a smaller number of datum points on average for each mean RT and *SD* and thus in the context of more intrinsic noise. It is important to note, however, that more free parameters were required for the CMPL fits (nine) than were required for the

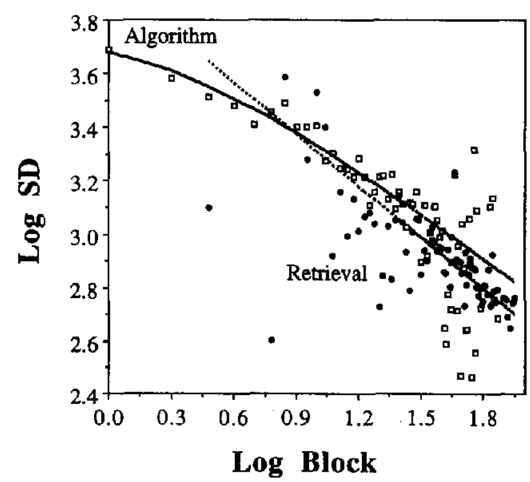

Figure 10. Component power law theory fits to the standard deviation (*SD*) data for the algorithm and retrieval strategies in Experiment 1.

instance theory fits (five). In Experiment 2 the two theories are compared in a context that reverses this free-parameter inequity.

Practice: RT results for the no-transition participant. A supplemental analysis was performed comparing the RT results for the 18 participants who reported a transition to retrieval with those of the single participant who reported using the algorithm almost exclusively throughout the five practice sessions (see Figure 11). RTs are collapsed across consecutive three-block sequences for the no-transition participant to reduce noise. The deviations from the power function that are clear for the transition participants are not evident at all for the no-transition participant. This result is as predicted by the CMPL model; because no strategy transition occurred for this participant, no deviation from power-function speedup should be present. Note also that although the no-transition participant was one of the fastest at solving problems initially, his performance at the end of practice was the slowest among all 19 participants. This effect provides confirming evidence for the claim based on the strategy-probing data that no strategy transition occurred for him. The instance theory model as developed in Logan (1988) assumes a constant distribution of algorithm finishing time, and cannot account for the speedup in pure algorithm execution time for this participant (provided one accepts that the strategy-probing results for this participant are valid).

Test. Results of the delayed test were generally consistent with those of the immediate test. These results are described in Rickard and Bourne (1995) and are not discussed further here. The proportion of trials on which each of the three strategies was used in the three immediate test conditions is shown in Figure 12, collapsed across blocks. No-change problems exhibited a high proportion of retrieval responses, which is not surprising given the complete transition to retrieval indicated for these problems during practice. In contrast, the algorithm was reported in most cases

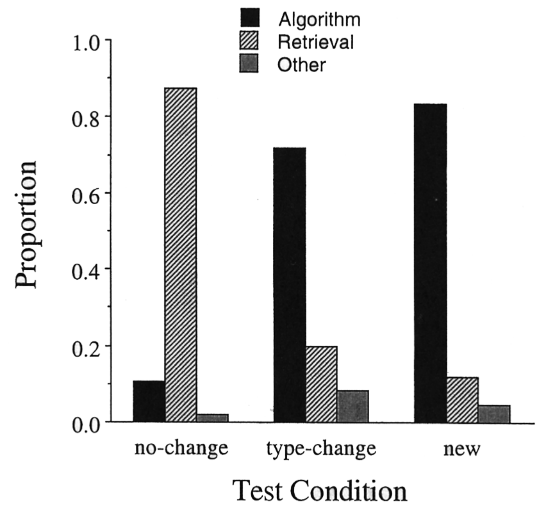

Figure 12. The proportion of trials on which each of the three strategies was reported in the three immediate test conditions of Experiment 1, collapsed across block.

for new and type-change problems. A planned contrast on the proportion retrieved comparing the no-change condition with the other conditions was highly significant, $F(1, 17) = 323$, $p < .001$, but a second planned contrast comparing the type-change and new problems conditions was not reliable, $F(1, 17) = 3.08$, $p = .088$.

Error proportions and RTs at test showed similar results. The overall error proportions (collapsed across blocks and strategies) for the no-change, type-change, and new problems conditions were .024, .250, and .284, respectively. The large difference between no-change problems on one hand and type-change and new problems on the other hand was strongly reliable, $F(1, 17) = 53.8$, $p < .001$, but the difference between type-change and new problems was not, $F(1, 17) < 1$. The RTs are shown in Figure 13 as a function of

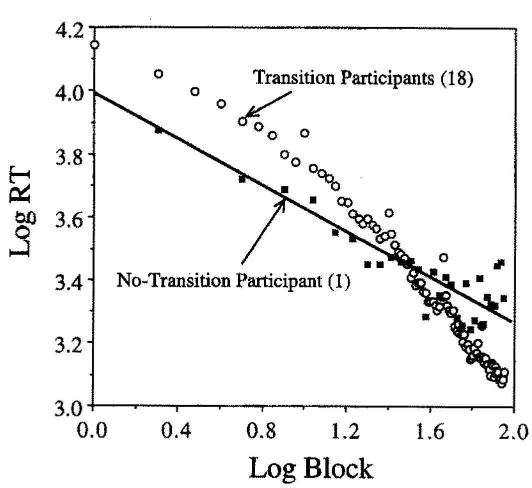

Figure 11. Response time (RT) results for the single no-transition participant in Experiment 1, with the best fitting power function (including asymptote parameter).

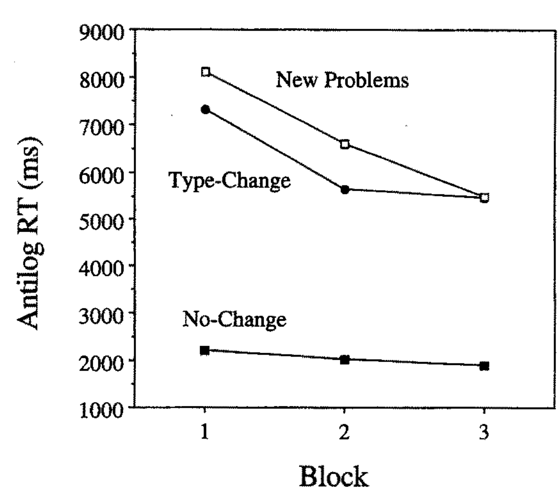

Figure 13. Response times (RTs) for the three conditions of the immediate test of Experiment 1, collapsed across block.

block and test condition. An ANOVA revealed reliable effects of test condition, $F(2, 17) = 42.6$, $p < .001$; and block, $F(2, 17) = 6.68$, $p = .004$; and of the interaction between these variables, $F(4, 17) = 5.82$, $p < .001$. The interaction reflects greater speedup across the blocks of test in the new problems and type-change conditions than in the no-change condition. Contrasts showed that RTs in the no-change condition were reliably faster than in the type-change and new problems conditions, $F(1\ 17) = 84.05$, $p < .001$, but there was no evidence of any difference between the type-change and new problems conditions, $F(1, 17) = 1.21$, $p = .28$.

In summary, consistent with the assumptions of the CMPL model, the transition to retrieval was quite specific to the problems on which participants practiced. Even reversed versions of the practiced problems (type-change problems) benefited little if at all from practice (for related transfer results in standard arithmetic, see Rickard and Bourne, 1995; Rickard and Bourne, 1996; and Rickard, Healy, and Bourne, 1994). These findings lend support to the claim that problem and answer nodes are distinctly represented and that the association that is strengthened with practice is unidirectional from the problem to the answer. Note that instance theory can also predict the observed performance difference between no-change and new problems. It would also be able to account for the failure of practice to transfer to type change problems given the added assumption that instances are unidirectional.

The above transfer results, as well as similar results in standard arithmetic reported by Rickard et al. (1994; see also Rickard and Bourne, 1996) are inconsistent with a plausible alternative model of the representation of arithmetic facts proposed by Campbell (1997), which assumes that complementary problems in multiplication and division (e.g., 4 × 7 and 28/7) access the same representation. However, the results reported by Campbell do suggest some currently unidentified type of relatedness between complementary arithmetic problems.

Two additional effects in the test data are worth considering. First, there was a substantial increase in proportion of algorithm responses for no-change problems at test compared to the last session of practice (from less than .01 during the last session of practice to about .1 during the immediate test). There was also a reliable increase in RTs for no-change problems at test compared to the last session of practice, even when considering only test problems on which participants reported using the retrieval strategy (approximately 2,000 ms at test versus around 1,200 ms at the end of practice). Analogous increases in RT at test were observed by Rickard et al. (1994) for standard arithmetic. These performance decrements at test are not predicted by the current versions of either the instance theory or the CMPL model. It remains to be demonstrated whether either approach can be extended to account for them.

Second, a comparison of the algorithm RTs from practice with RTs for new problems at test provides a rough estimate of the amount of speedup that reflects general algorithm speedup, and the amount that reflects speedup in executing the algorithm for specific problems. The CMPL model predicts problem-specific speedup but no general speedup, and the instance theory as developed in Logan (1988) assumes no algorithm speedup of either type. If algorithm speedup is solely general, then the RTs for algorithm trials on the last few blocks of practice on which they were reported should be roughly the same as the RTs for new problems at test. Alternatively, if the algorithm speedup is solely problem specific, then RTs for new problems at test should not be different from RTs on the first block of practice. Algorithm RTs were around 13,000 at the beginning of practice. On the last practice block on which the algorithm was reported at least 10% of the time, algorithm RTs were around 3,600 ms. These compare to RTs for new problems on the first block of the immediate test of around 8,000 ms. Thus, there appears to have been both general speedup (not predicted by either model) and specific speedup (predicted only by the CMPL model) in algorithm execution with practice.

The general speedup effect is not particularly surprising in hindsight because two similar but distinct algorithms were learned by participants (one for Type 1 and one for Type 2 problems) to only a minimal proficiency prior to the first block of practice. On the first few practice blocks, it seems likely that many participants found it necessary to refer to the algorithm example sheets that were available throughout practice. This consultation of external information about the algorithm would be much less likely at test, providing a candidate account of the general algorithm speedup effects.

Experiment 2

In Logan's (1988) alphabet arithmetic experiment, overall plots of the addend 2 data showed only negligible deviations from power-function speedup and reduction in *SD*. As addend size increased, however, these deviations from log–log linearity were increasingly evident. These effects are consistent with the CMPL model for the following reasons. First, the CMPL predicts that strengthening of the memory-retrieval strategy occurs independently of any characteristics of the algorithm (such as addend size). Also, the strategy-choice process is the result solely of a local competition between the first step of the algorithm and the memory retrieval strategy. That is, the number of steps in the algorithm, or its global difficulty, does not directly influence the strategy-choice process. Given the reasonable additional assumption that the connection strengths associated with the first step of the algorithm are roughly equivalent across the set of problems constituting the three addend size groups, then the strategy-choice process should not be correlated with addend size according to the model. In combination, the predictions above lead to the additional prediction that the number of trials needed to make the transition to retrieval, as well as RTs and *SD*s for the retrieval strategy, should be equivalent for the three addend sizes.

However, the RTs and *SD*s for the algorithm strategy clearly increase as an additive function of addend size.

Thus, the CMPL model predicts that the distance between the algorithm and retrieval power functions for both the RTs and *SD*s increase with increasing addend size. Finally, if the overall data (i.e., data collapsed over strategies) are plotted, then it follows that deviations from log–log linearity are increasingly prominent for both RTs and *SD*s with increasing addend size. One purpose of Experiment 2 was to further test these predictions using the alphabet arithmetic task. To assure comparability of this experiment with that of Logan (1988; see also Compton & Logan, 1991), the task was constructed as a verification task (e.g., F + 3 = 1; true or false?).

A second motivation for this experiment was to compare the instance theory and the CMPL model under conditions in which the CMPL model requires fewer free parameters. This experiment involved training participants on equal number of problems with addend sizes of 3, 5, and 7. Under these conditions, 15 free parameters, 5 for each addend size, are required to fit the instance theory model described by Logan (1988). However, the CMPL model embodies several mathematical constraints that can be applied to these data that allow it to be fit with a total of only 10 free parameters. It is important to note that inclusion of the following constraints is required for the regression model to be an accurate representation of the predictions of the simulation model. First, as discussed earlier, the CMPL model requires that the retrieval RTs and *SD*s are identical for each level of addend size. Thus, only a single power function, with 2 free parameters (intercept and slope), is needed to fit retrieval RTs for all three addend sizes, and only a single 2-parameter power function is needed to fit retrieval *SD*s, for a total of 4 free parameters for the retrieval strategy. Second, because the CMPL model assumes that algorithms are a string of successive memory-retrieval events with additive characteristics, the power-function intercepts for the RTs for the algorithm strategy can be fit with only 2 parameters; 1 parameter for the intercept for the addend 3 problems, and a 2nd parameter representing the constant increment (in terms of raw RTs) in the intercept for each increment in addend size. So far, 6 free parameters have been committed. Third, again, because the algorithm is assumed to be an additive function of addend size, only a single-slope parameter is necessary for the algorithm RTs (given the reasonable assumption that the strength parameter, $c2$, is equivalent on average over all steps of the algorithm for a given problem). Fourth, again because the algorithm RT is assumed to be an additive function of addend size, the intercept for the algorithm *SD*s is constrained to be a constant proportion of the intercept for algorithm RTs over all addend sizes. Thus, only a single additional free parameter is needed to fit the intercept of the algorithm *SD*s. Finally, the slopes of the *SD*s for the algorithm are also constrained to be identical over different addend size, thus allowing a single slope parameter for fitting the slopes of the algorithm *SD*s.[6] These 9 free parameters, plus a 10th that represents previous learning for the algorithm strategy, are sufficient to fit the entire practice data set across all three addend sizes. The corresponding equations and inequalities are

$$\mathrm{RTalg3} = b1(tr + p)^{-k1}$$

$$\mathrm{RTalg5} = (b1 + 2x)(tr + p)^{-k1}$$

$$\mathrm{RTalg7} = (b1 + 4x)(tr + p)^{-k1}$$

$$SD\mathrm{alg3} = e(b1)(tr + p)^{-k2}$$

$$SD\mathrm{alg5} = [e(b1 + 2x)](tr + p)^{-k2}$$

$$SD\mathrm{alg7} = [e(b1 + 4x)](tr + p)^{-k2}$$

$$\mathrm{RTret3} = \mathrm{RTret5} = \mathrm{RTret7} = b2(tr)^{-k3}$$

$$SD\mathrm{ret3} = SD\mathrm{ret5} = SD\mathrm{ret7} = b3(tr)^{-k4}$$

$$k1 < k2$$

$$k3 < k4$$

where $b1$ is the algorithm-intercept parameter for addend = 3 problems, $b2$ and $b3$ are the intercept parameters for retrieval RT and *SD*, p is previous learning, $k1$ through $k4$ are the rate parameters, x is the raw RT increment associated with each additional step of the algorithm, and e is the proportionality constant relating RT and *SD* for the algorithm.

Method

Participants. Twenty-one participants from an introductory psychology course participated in the experiment for credit. Participants were tested on IBM-type personal computers, programmed with the MEL software (Schneider, 1988). Twenty-four problems (12 true and 12 false) were constructed (see Appendix B): 8 problems with the addend 3, 8 with the addend 5, and 8 with the addend 7. Four problems within each addend size were true, and 4 were false.

Procedure. There were four experimental sessions, the first three on Monday, Wednesday, and Friday of 1 week, and the fourth on Monday of the following week. Each session lasted 30–45 min. Participants were tested in groups of up to 4. At the beginning of the first session, the participants were introduced to the alphabet arithmetic task by way of one true and one false problem worked on a blackboard by the experimenter (neither of these problems were in the stimulus set). Participants then performed the task independently at their own computer. During the first session, participants performed 15 blocks of problems, where each block was one exposure to each of the 24 problems in the participant's practice set. Problems were presented one at a time in the middle of the screen. Participants entered *true* or *false* using specially marked adjacent keys on the numeric keypad. Participants were instructed to use either the pointer finger of both hands (one for true and one for false) or the pointer and index finger of one hand, whichever was more comfortable. The *true* and *false* keys were counterbalanced across participants. Participants were instructed to work as fast as possible while being accurate. They were told that they could rest briefly between blocks of problems.

[6] These last two constraints do not strictly fall out of the mathematics because of the complicating factor of the between-items component of the variance, which is not an exact power function. However, simulation results show that in practice these constraints nevertheless hold almost exactly.

The participant's answer for each problem was collected. Strategy probes (algorithm, retrieval, or other) were collected on one third of the trials as in Experiment 1. The second, third, and fourth session consisted of 21, 24, and 27 blocks of problems, respectively.

Results and Discussion

True problems were solved slightly faster and slightly more accurately than false problems. These effects, however, did not enter into any interactions with other variables, and thus data were collapsed across the true–false distinction in all of the following analyses. Error rates for addend 3 problems were .058, .042, .044, and .045 in sessions 1, 2, 3, and 4, respectively. For addend 5 problems these values were .083, .099, .077, and .064, and for addend 7 problems they were .090, .072, .072, and .070. A 4 (session) by 3 (addend size) within-subjects ANOVA performed on the proportion of errors indicated a reliable increase in error rates with increasing of addend size, $F(2, 20) = 7.22$, $p = .002$. There was no reliable effect of session, $F(3, 20) = .48$, $p = .699$, and no reliable interaction of these two variables, $F(6, 20) = 1.77$, $p = .111$. All analyses reported later were limited to correctly solved problems.

The strategy probing results are shown in Figure 14, collapsed over participants, problems, and addend size. Practice appears to have been successful in creating a transition to retrieval. By about block 60, retrieval was the reported strategy on nearly all trials. As in Experiment 1, there were very few "other" responses, suggesting that there were no intermediate stages in which some third strategy was used. A within-subjects ANOVA performed on the overall proportion of retrieval responses with a single factor of addend size (M = .794, .791, and .799 for addend sizes of 3, 5, and 7, respectively) indicating that the number of trials needed to make the transition to retrieval was not influenced by addend size, $F(2, 20) < 1$. This finding is exactly as predicted by the CMPL model.

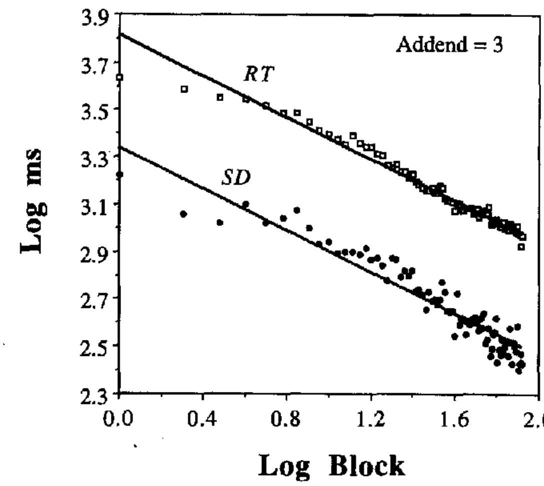

Figure 15. Instance theory fits to addend = 3 response time (RT) and standard deviation (*SD*) data from Experiment 2.

Instance theory fits. Figures 15, 16, and 17 show the overall log RTs and log *SD*s for the three addend sizes plotted as a function of log block. Also shown in these figures are the best fitting power functions as predicted by instance theory (Logan, 1988). Systematic deviations from the predictions are clearly evident for both the RT and *SD*. Also, as was the case in the alphabet arithmetic data of Logan (1988), and as predicted by the CMPL model, the deviations become larger with increasing addend sizes. The overall r^2 of the instance theory fit over all addend sizes was .95.

The instance theory prediction of identical values for RT and *SD* power-function rate parameters was evaluated separately for each addend size by computing the parameter estimates separately for each participant, as described in

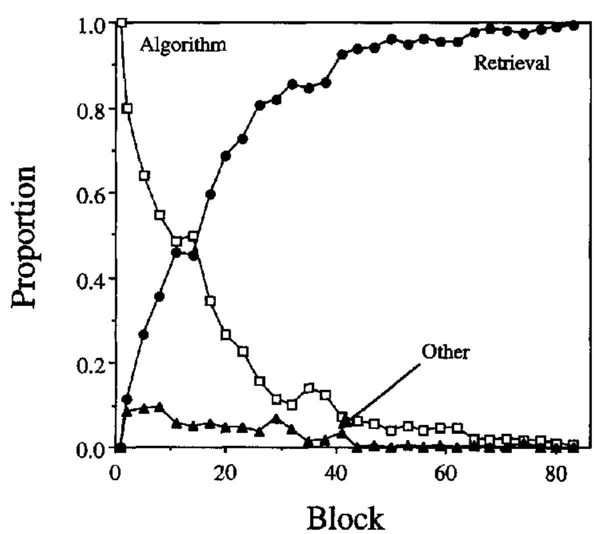

Figure 14. Proportion of algorithm, retrieval, and other responses as a function of practice in Experiment 2.

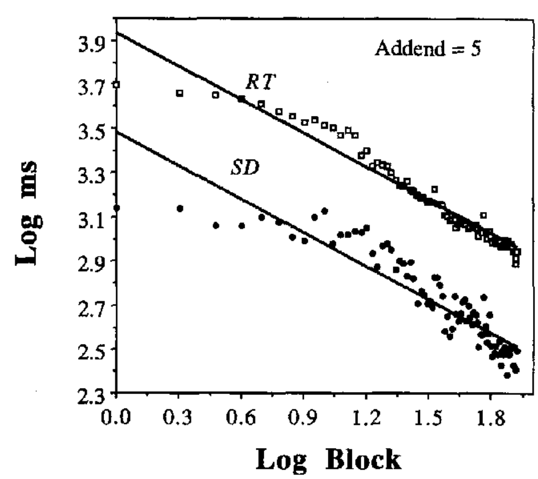

Figure 16. Instance theory fits to addend = 5 response time (RT) and standard deviation (*SD*) data from Experiment 2.

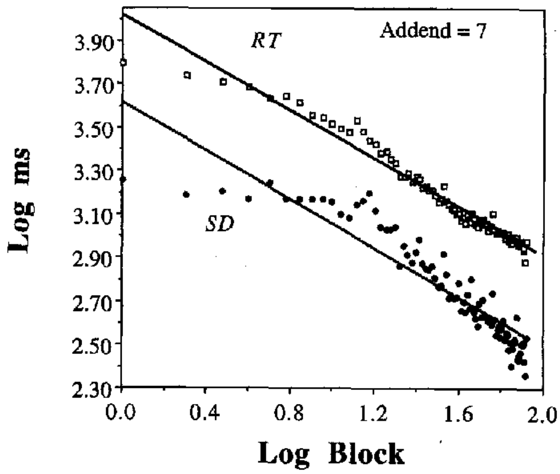

Figure 17. Instance theory fits to addend = 7 response time (RT) and standard deviation (*SD*) data from Experiment 2.

Experiment 1. For addend 3 problems, 15 of 21 participants had larger rate estimates for *SD* than for RT. However, for addend 5 and 7 problems, 15 and 14 of the participants, respectively, had larger rate estimates for the RT than for the *SD*. These effects for addend 3 and 5 problems were reliable by a binomial sign test ($ps < .05$). This pattern is analogous to that obtained by Logan (1988; Experiment 4) for alphabet arithmetic. In that experiment, rate estimates for the *SD*s were larger than for the RTs for both true and false problems with addend sizes of 2, 3, and 4, but the reverse was true for addend 5 size problems. Note also that the pound arithmetic task of Experiment 1, which had longer algorithm times than any of the alphabet arithmetic conditions discussed earlier, exhibited larger rate estimates for the RTs than for the *SD*s. Thus evidence from three experiments now suggests that the rate estimates resulting from power-function fits to the overall RTs and *SD*s are not necessarily the same, and they further suggest that the rate estimates for the RTs increases faster than that for the *SD* as algorithm difficulty increases. This interaction contradicts the strict instance theory (Logan, 1988) prediction that learning rates are identical for the RT and *SD* regardless of algorithm difficulty. Note, however, that it remains to be seen whether this finding is problematic for the instance theory approach more generally (see Logan and Etherton, 1994).

CMPL fits. RTs and *SD*s corresponding to the algorithm and retrieval strategies were identified using the filtering approach discussed in Experiment 1. Figure 18 shows the algorithm and retrieval RT results and the best fitting CMPL functions for each addend size. Figure 19 shows the results for the algorithm *SD*s, and Figure 20 shows them for retrieval *SD*s. The overall r^2 for the CMPL fit to the entire data was .97. As in Experiment 1, fits are limited for each

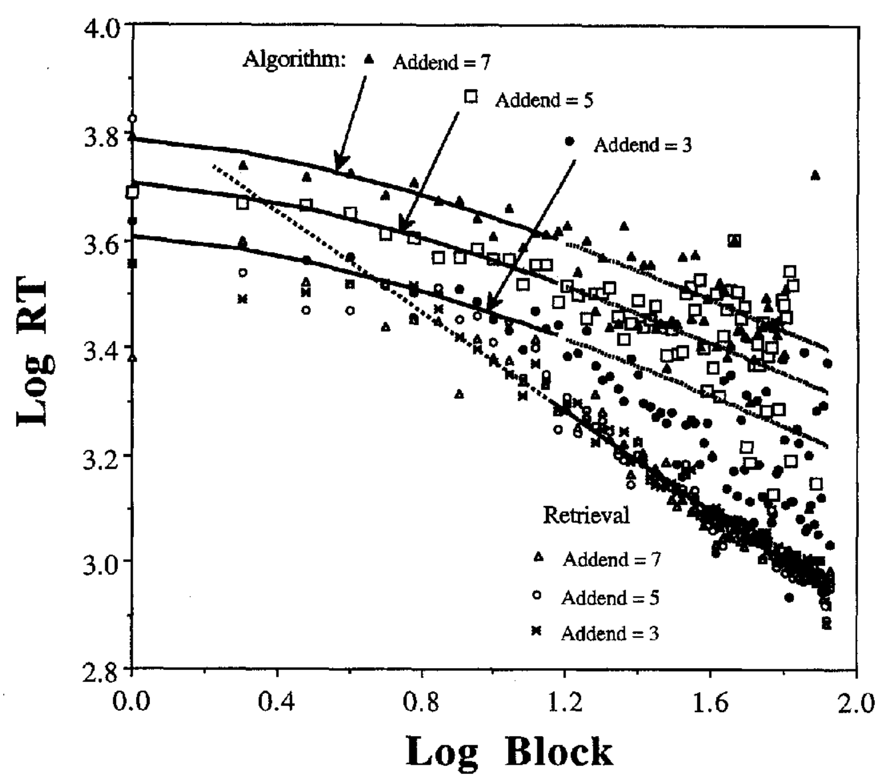

Figure 18. Component power law theory fits to the algorithm and retrieval response time (RT) data of Experiment 2.

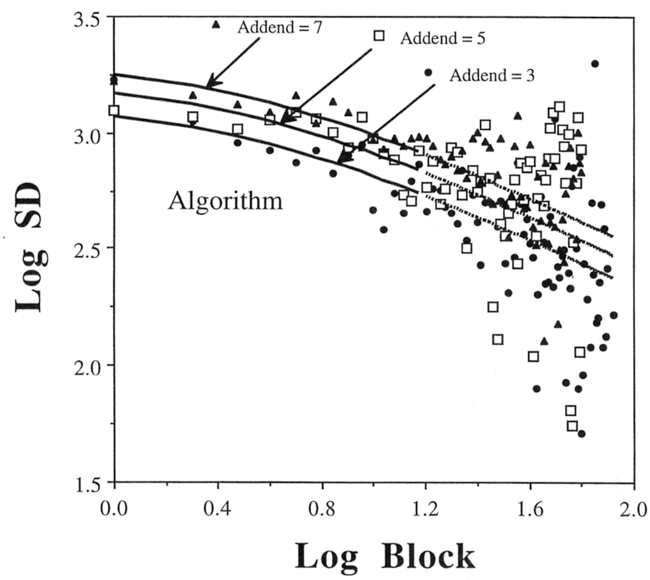

Figure 19. Component power law theory fits to the algorithm standard deviation (*SD*) data of Experiment 2.

strategy to the range covered by the solid lines shown in the figures. Hatched lines cover ranges for each strategy for which there were far fewer observations and thus far more intrinsic noise. Note that the systematic deviations from the predictions that are clear in the instance theory fits to the overall data are no longer evident in the CMPL fits.

As predicted by the CMPL model, there were concave downward deviations from the power-function fits to the RT and *SD* data for the retrieval strategy during the first few retrieval trials. Also as predicted by the CMPL model, slopes for the *SD*s were steeper than were those for corresponding RTs, and unlike Experiment 1, this effect was reliable for both the algorithm strategy, $F(1, 20) = 4.47$, $p < .05$, and the retrieval strategy, $F(1, 20) = 5.28, p < .05$.

One additional test was performed to further evaluate the CMPL prediction that retrieval RTs and *SD*s do not vary with addend size. Although the regression model that was fit to the data embodies this constraint and provided good fits, there could still be statistically reliable differences in retrieval RTs that were not evident in that analysis. To explore this possibility, an analysis of covariance (ANCOVA) with a continuous factor of log block and a categorical factor of addend (3, 5, or 7) was performed on the log RTs for retrieval. As expected, there was a reliable effect of log block, $F(1, 20) = 489$, $p < .0001$. However, there was no reliable effect of either addend, $F(2, 40) = .43$, $p = .66$, or of the interaction log block by addend, $F(2, 40) = .63$, $p = .54$. The same ANCOVA was performed on the log *SD*s for the retrieval strategy to investigate whether addend size predicted retrieval-based *SD*s. There was again a reliable effect of log block, $F(1, 20) = 69.5$, $p < .0001$, but there was no reliable effect either of addend, $F(2, 40) = .08$, $p = .92$, or of the interaction log block and addend, $F(2, 40) = .06$, $p = .94$. In sum, then, in line with the predictions of the CMPL model, there was absolutely no evidence that retrieval-based performance (whether indexed by RTs, *SD*s, or strategy-probing data) depended on addend size.

General Discussion

Two experiments provide new evidence in support of the general claim of Logan (1988) and Siegler (1988) that practice on skills that originally require execution of sequential algorithms can produce a strategy shift to direct memory retrieval (see also, Ashcraft, 1992). However, the results from both experiments also suggest that the CMPL model may provide a better account of the mechanisms underlying this strategy shift than does the version of the instance theory proposed by Logan (1988). In both experiments, the CMPL fits produced equivalent or higher r^2s than did the instance theory, and these r^2 values were obtained despite the fact that the CMPL model must be fit using only a subset of the available data. It is important to note that fitting the CMPL model required more free parameters than did the instance theory for Experiment 1. However, for Experiment 2 this inequity was reversed, and the CMPL model still provided higher r^2 values. Perhaps the most convincing evidence that favors the CMPL model over the instance theory is that it exhibited substantially fewer systematic visual deviations of the data from the predictions across all conditions of both experiments.

In addition to the good statistical and visual data fits, the CMPL model makes several unique predictions that were confirmed, including those of no shift from algorithm to retrieval for either type reverse or new problems at test in Experiment 1, of no deviation from power-function speedup for a participant who did not exhibit a strategy shift in Experiment 1, of steeper slopes for the *SD*s than for the RTs

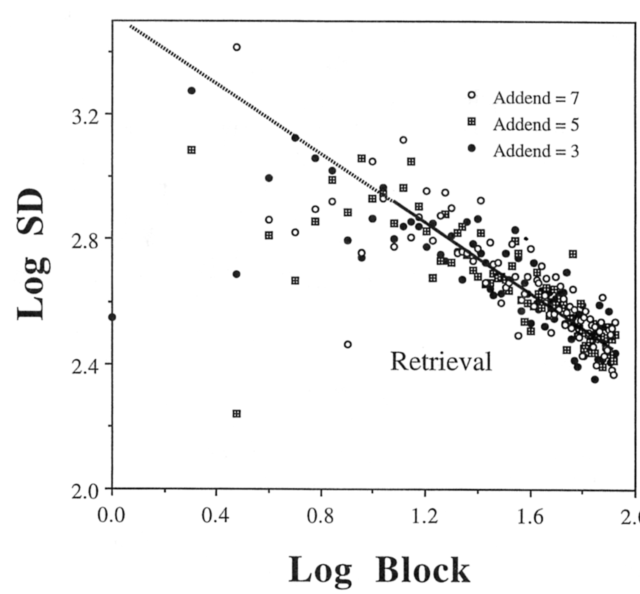

Figure 20. Component power law theory fits to the retrieval standard deviation (*SD*) data of Experiment 2.

within each strategy (reliable in Experiment 2 but not in Experiment 1), of concave downward deviations from the power function for the retrieval Experiment prior to the halfway point of the strategy transition, and of identical retrieval-based performance for the three addend sizes used in Experiment 2.

As discussed previously, Logan (1988) suggested a modified version of the instance theory to account for the deviations from log–log linearity in the overall plots of the RTs and *SD*s. This account assumes that at some point during practice, participants shift from a less efficient to a more efficient memory strategy. This account cannot be ruled out at present. However, it has not been explicitly tested to date by actually fitting a modified instance theory to the data. It is not immediately obvious that it provides a reasonable account of all aspects of the data, such as the shape of the deviation from log–log linearity for the RTs and *SD*s, the differences in rate parameter estimates for the overall RTs and *SD*s, or the equal RTs and *SD*s for the retrieval strategy over different addend sizes as observed in Experiment 2. Also note that if the hypothesis of a second transition to more efficient memory strategy is to be demonstrated convincingly, some independently validation that such a process occurs is needed. At the least, the presence of two strategy transitions to retrieval needs to be verified with strategy-probing data.

An important and unique prediction of the instance theory that has received support in previous research (Compton & Logan, 1991) is that participants, given the opportunity, sometimes choose strategy categories that appear to indicate that they do indeed execute both the algorithmic and retrieval strategies in parallel. Compton and Logan conducted two such experiments, using the alphabet task, that merit discussion in detail. In the main experiment, they gave participants three strategy report options after selected trials: (a) counted through the alphabet, (b) remembered the answer without counting, and (c) counted and remembered at the same time. In a follow-up experiment, they again gave participants the first two strategy report options just stated and also six additional options: (d) first counted and then got the answer by remembering, (e) tried to remember and then got the answer by counting, (f) tried to count and remember simultaneously and got the answer by counting, (g) tried to count and remember simultaneously and got the answer by remembering, (h) used a strategy that is not listed above, and (i) made a mistake or did not know how to solve the problem.

In the main experiment, participants chose Option a on 20% of the problems, Option b on 56% of the problems, and Option c on 24% of problems. Participants typically chose Option c (simultaneous counting and remembering) during the middle of the strategy-transition interval, and the associated RTs were in between those for the pure algorithm and the pure memory-retrieval strategy-response categories. These results are consistent with the instance theory interpretation that these strategy responses reflect concurrent execution of the algorithm and retrieval strategies. Such results may prove problematic for the CMPL model and need to be addressed in future research. However, there are several factors that should lead one to view the Compton and Logan (1991) results with at least some skepticism. First, in the follow-up experiment, where the eight strategy report options listed earlier were included, the percentage of simultaneous strategy reports (Options f and g combined) fell to 9%, and were not reported at all by about half of the participants. Each of the other strategy options in the follow-up experiment can potentially be accounted for by assuming some sort of serial strategy execution that is not inconsistent with the broad assumptions motivating the CMPL model.[7] Second, the percentage of trials on which participants reported Options a and b in the follow-up experiment was very close to the percentage of trials on which those options were reported in the main experiment. Also, the sum of the percentages for Options d through h in the follow-up experiment is very close to the percentage of trials on which participants chose Option c in the main experiment. These results suggest the possibility that beyond Options a (counted) and b (remembered), participants have difficulty accurately introspecting on their thought processes. That is, it may be that thought processes that in fact did not reflect simultaneous strategy execution but rather on which participants had difficulty accurately introspecting were grouped into Option c in the main experiment and that the analogous set of trials were distributed roughly evenly across Options d through h in the follow-up experiment. The possibility that participants have difficulty making accurate introspections on some trials is acknowledged by Compton and Logan (1991) who stage that, "Given the authors' experience with the task, it seems likely that subjects are not able to make find distinctions about the strategies they are using when the both count and remember on the same trial" (p. 156). In summary, although the Compton and Logan strategy-probing results are interesting and important, follow-up research seems warranted before making a strong conclusion that concurrent strategy execution is occurring.

Issues in Automaticity and Attention

The instance theory claims that automatic processing reflects direct retrieval of instances from memory. The CMPL model, in contrast, claims that there is a continuum from more goal-driven to more stimulus-driven retrieval from memory. For example, on the first few retrieval trials for a given item in the experiments described in the article, the CMPL model predicts that initial selection of the retrieval subgoal is necessary for the retrieval problem node to be able to win the competition (i.e., for retrieval to occur). Thus, retrieval in this case is strongly goal influenced. In contrast, consider the last few algorithm trials prior to the

[7] Note that the CMPL as presented in this article assumes either algorithm or retrieval execution and does not allow for the possibility of serial execution of (for example) retrieval followed by execution of the algorithm as a check. However, such sequential strategy execution would not be inconsistent more generally with the core claim of the model that algorithm and retrieval strategies are not executable concurrently.

transition to retrieval. As discussed in the *Simulations* section, the CMPL model predicts that on some of these trials the retrieval subgoal is selected initially but that the problem node for the first step of the algorithm is still selected, in turn forcing a shift of activation at the subgoal level from the retrieval to the algorithm subgoal in order to maintain internal coherence in the system. In this case the retrieval event associated with the first step of the algorithm is almost purely stimulus driven. Indeed, it occurs despite the initial goal to execute the direct retrieval strategy. This stimulus-driven retrieval in the CMPL model has one of the attributes typically associated with automaticity; namely, it can proceed to completion under some (unusual) circumstances largely outside of the control of attention (i.e., outside of the influence of goal-based processing). However, stimulus-driven retrieval in the model also has several properties that contrast from the model view of automaticity. In particular, (a) whether of not retrieval is automatic is determined not by the absolute strength associated with retrieval for that item but rather by the relative strength of that item to all other competing retrieval candidates, and (b) stimulus-driven retrieval does not reflect operation of a form of automatic memory retrieval that is qualitatively distinct from other memory processes. In other words, the same types of representations and connection pathways are involved in both stimulus-driven and in goal-driven retrieval (see Cohen et al., 1990, for related points in their connectionist model of the Stroop effect.).

The CMPL also differs from the instance theory in that it predicts that even relatively stimulus-driven, or automatic, retrieval cannot take place in parallel for two or more items (or for two or more interpretations of a given stimulus item). Multiple candidates for retrieval are activated in parallel during early stages of the retrieval. However, selection of one response always results in suppression of all other competing responses. This theoretical claim is consistent with dual task studies of the psychological refractory period (PRP) conducted by Pashler and colleagues (see Pashler, 1993, for a review), which show that response selection even for two very simple tasks cannot occur concurrently. For example, the decision of whether a tone is high or low does not occur concurrently with the visually based decision such as whether a stimulus is a letter or not, even when the response modalities for these two tasks do not interfere (i.e., when the response for the tone is verbal and that for the visual stimulus is manual). This result holds even after 2,500 trials of practice (Dutta & Walker, 1995). Other experiments in which one task was a tone task and the second task involved either retrieval of a paired-associate response from memory or recognition of a previously presented word yielded similar results (Carrier & Pashler, 1995). The results of Pashler and colleagues map naturally to the CMPL model if one assumes that their response-selection stage, at which the dual-task bottleneck occurs, corresponds roughly to processing within the subgoal and problem level of the simulation model.

The PRP results appear to be problematic for the instance theory. One reasonable counterargument, however, is that the memory tasks used by Carrier and Pashler (1995) are not automatic. Consider, for example, the possibility that recognition of words in the Carrier and Pashler (1995; Experiment 2) study involved explicit episodic retrieval of previous exposures to the words during training. The instance theory applies to implicit (i.e., not episodically mediated) retrieval (Logan, 1990) and thus would not necessarily be inconsistent with the finding that episodic retrieval cannot occur concurrently with the tone task. The critical PRP experiment needed to test the instance theory, in which participants are first given extensive practice on the memory task until there is clear evidence of automatized retrieval and are then tested using a PRP task, remains to be done.

The CMPL model treats attention in much the same way that Cohen et al. (1990) treat it in their connectionist model of the Stroop effect. In both models, attentional modulation is graded and is mediated through the same types of connections through which other nodes in the network are connected. One way in which the CMPL model differs from the Cohen et al. model is that the subgoal nodes (the rough equivalent of the Cohen et al. attentional units) for each individual strategy are not "clamped on" to an active state for the duration of each retrieval event. Rather, they accrue activation from the general task goal and from the stimuli as cycling proceeds. The assumption that subgoal nodes are clamped on is reasonable for the Stroop task. It would also be a viable alternative approach to the current tasks. One could argue that participants adopt a strategy of first selecting the algorithm subgoal prior to each trial, and then after sufficient practice, they switch to selecting the retrieval subgoal. In this version of the model, the subgoals would play a role that is strongly analogous to the modulatory role played by the attentional nodes in the Cohen et al. (1990) simulation. By leaving the subgoals inactivated at the onset of the trial in the simulations reported in this article, I effectively assume that participants, throughout practice, take a neutral stance with respect to strategy execution at the initiation of each trial. That is, they let the information from the stimulus, in combination with the relative strengths of pathways from the "solve problem" goal to the subgoals guide their strategy decision.

Strategy-Choice Processes

The question of whether or not retrieval and algorithmic processes can be executed in parallel is fundamental and relevant to many ongoing lines of research in cognitive psychology. If retrieval and algorithmic strategies are executed in parallel and independently, as assumed in the instance theory, then scheduling problems (Townsend & Schweickert, 1989) in some skill processes are automatically resolved (or, more accurately, simply do not exist). However, if as I claim in this article, strategy-choice processes are critical in such tasks, then many very important questions arise as to the mechanisms of strategy choice and the various factors which might influence it. A comprehensive cognitive theory that addresses these questions is important for development of human factors models applicable in a variety of real-world contexts.

The CMPL model makes the simple preliminary claim that only item-specific processes (i.e., strength of connections from the external stimulus items to the problem nodes) and strategy-specific processes (i.e., strength of connection from the general solve problem goal to the strategy subgoal), but no other factors, determine strategy choice. The item-specific factor is supported by the high degree of problem specificity in the transition to retrieval. Both factors also receive some support indirectly by the good overall fits of the CMPL model to the data sets. Both the item-specific and strategy-specific factors have previously been demonstrated, along with other factors, in an elegant series of studies by Siegler and colleagues in the domain of children's arithmetic (Lemaire & Siegler, 1995; Siegler, 1988). An important goal for future work is to determine the relative extent to which these factors along with other possible task, environmental, and individual difference factors influence strategy choice in adults.

The CMPL model makes strong and apparently unique predictions about the relative effects of what we term *local* and *global* item-specific algorithm difficulty on the strategy-choice process. According to the model, local algorithm difficulty (i.e., the difficulty of the first step of the algorithm) is relevant to the strategy-choice process. If the first step of the algorithm has a relatively high strength for a given problem, the transition to retrieval takes longer, other factors being equal. However, global difficulty, which is most naturally indexed by overall algorithm RTs, has no logically necessary correlation with the strategy-choice process, according to the model. The transition to retrieval should not depend on overall algorithm difficulty (e.g., the number of algorithm steps), provided that the difficulty of the first step of the algorithm is held constant. This is exactly the state of affairs with respect to the addend size variable in Experiment 2, and the data confirmed this prediction of the model. The generality of the effect merits further empirical investigation.

The CMPL model has many points of contact with the adaptive strategy choice (ASCM; Lemaire & Siegler, 1995) model of children's strategy choice. Both models assume a shift from algorithm to retrieval with practice and both assume a nonparallel strategy choice and execution process. The current results show that these assumptions appear to generalize to adults, at least for some tasks. The models differ, however, both in terms of empirical emphasis and in some of their core assumptions. The ASCM model has been applied to date primarily to account for an impressive variety of strategy shifts in children's learning. The CMPL model currently deals only with the strategy transition from algorithm to retrieval, and it focuses more on functional form of RTs and *SD*s of correct trials as they relate to practice. It remains to be seen whether the CMPL can be extended to cover a broader range of strategy-choice processes and whether the ASCM model can predict the phenomena that are the focus of this article, such as strategy-specific power functions for RTs and *SD*s.

As discussed above, the CMPL model assumes that all strategy choice reflects a local competition between two-candidate memory-retrieval events. In contrast, strategy choice in the ASCM model is influenced only by global properties of the algorithm, such as overall RT and error rate. Data from Siegler and colleagues (see Lemaire & Siegler, 1995) demonstrate that a model that focuses only on global properties can provide good accounts of strategy choice in children's performance (although global versus local factors in strategy choice have not to date been directly manipulated in their work). Thus, taken as a whole, the available data suggest that both local and global algorithm factors may be important in strategy choice. It will be important in future work to determine the relative influence of these two factors in various contexts.

Generalization of the CMPL Model

It remains an open question how far basic predictions of the CMPL model generalize beyond the tasks explored in this experiment. It seems quite likely that it generalizes to other arithmetic and related tasks. Rickard (1994) discussed results of an arithmetic task by Carlson and Lundy (1992) that exhibited a transition toward memory-based performance with practice and also a concave-downward deviation from log–log linearity that is generally consistent with that expected by the model. There is also preliminary evidence that the model may generalize to other tasks. The data from a 10-finger task reported by Seibel (1963, and discussed by Newell & Rosenbloom, 1981) show deviations from log–log linearity that are characteristic of those predicted by the model. In that task, some subset of 10 lights was turned on for each trial, and the participant pressed the corresponding keys. It is plausible that participants mapped lights to the corresponding fingers in a consciously mediated way initially but later were able to more reflexively make their responses. That is, they may have undergone a form of transition from an algorithm to direct retrieval. As another example, in three experiments by Palmeri (1997), stimuli consisting of 6 to 11 dots were presented repeatedly for up to 20 practice sessions and 208 repetitions per item. I plotted these data, reported in Palmeri (1995), separately in log–log coordinates for dot patterns of each numerosity, for each item type, and for each experiment. The signature deviations from log–log linearity predicted by the CMPL model are clearly evident for patterns of 10 and 11 dots in all three experiments, and also as predicted, these deviations become less pronounced as the number of dots decreases.

It is also encouraging that the CMPL model has points of connection with a variety of other skill models, including but not limited to the instance theory (Logan, 1988) the theories of Siegler and colleagues (Lemaire & Siegler, 1995), and Anderson (1993), the Stroop model of Cohen et al. (1990), and arithmetic fact retrieval and interference model such as those of Campbell and Oliphant (1992), Rickard, Mozer, and Bourne (1992), and Rickard and Bourne (1996). A synthesis among such models should ultimately provide a comprehensive account of learning and performance in mental arithmetic and related task domains.

Implications for the Power Law of Practice

The power law of practice has been generally accepted to be true for overall speedup with practice for any task since the seminal paper of Newell and Rosenbloom (1981). The data presented in this article present the first empirically strong challenge to that claim (note these data were originally reported in Rickard, 1994). The CMPL model now makes explicit and testable process-based predictions concerning when the power law holds for both RTs and *SD*s in the overall task, when it does not, and what the functional relations between these two measures are when trials are evaluated separately by strategy.

The demonstration that the power law does not hold overall for tasks exhibiting a strategy shift from algorithm to retrieval raises the questions of why the power law nevertheless does appear to hold in many other skill domains (Newell & Rosenbloom, 1981). As an initial attempt to address this question, I propose three classes of skill-acquisition tasks to which the power law applies with varying degrees of accuracy. First, learning that reflects strengthening of a single memory-retrieval event, or of a string of sequential retrieval events, yields exact power-function speedup in expected RT even at the item level. The power function is thus fundamentally a property of memory-retrieval practice. Second, there are some tasks, such as those explored in this article, that exhibit marked and discrete shifts between algorithm and memory retrieval strategies. For these tasks, the power law is simply incorrect as an empirical law of overall speedup with practice (although it holds within each strategy and may also yield a good approximation when RT differences between the two strategies are small). Delaney, Reder, Staszewski, and Ritter (in press) have independently reached the same conclusion. Note that a theoretically motivated definition of exactly what constitutes a unique strategy is required to substantiate and test this proposal. Within the context of the CMPL model, we can define a strategy as simply a unique string of memory retrievals, executed in the service of some problem-solving goal, and typically identifiable through participant reports. Given this definition, the CMPL model makes the strong prediction of power-function speedup and reduction in *SD* within each strategy. Third, there is an additional class of skills for which qualitative process changes occur with practice and for which the power law does nevertheless hold, at least to a good approximation when data are aggregated over items and participants. This class of tasks appears to exhibit types of process transitions other than algorithm-to-retrieval shifts, which are gradual and piecewise at the item level and which have been shown by mathematical derivation and by simulation to give rise to approximate power-function speedup (e.g., Newell and Rosenbloom, 1981; Anderson, 1983). For these tasks, the power law does not necessarily hold for a single item, but it is a good approximation in many cases when data are averaged over items.

Finally, it is worth speculating on how the parametric properties of power-function fits might differ systematically for the three cases of the framework described earlier. Consider the possibility that there is a constant rate parameter associated with speedup in memory retrieval with practice for a given participant. The CMPL model then must predict that power-function rate estimates are identical for a given participant across all single-step as well as multistep memory retrieval tasks (i.e., for practice on all strategies, as defined earlier). In tentative support of this speculation, the fits of the CMPL model to the data from Experiment 1 and 2 would have suffered only negligibly had the rate parameters been constrained to be the same for the algorithm and retrieval strategies.

In contrast, differing tasks exemplifying Case 3 described earlier might show widely varying rate estimates even within a given participant. Some tentative support for this possibility can be found in the meta-analysis of power-function fits reported by Newell and Rosenbloom (1981). Ultimately, models of skill acquisition should be able to predict not only when a power function should hold in a given data set but also what the rough values of those parameters should be and how such parameters are constrained across various conditions. The CMPL model represents one candidate framework through which progress in this direction may be possible.

Conclusions

The new model of skill acquisition introduced in this article provides a clear, constrained, and empirically supported account of the strategy shift from algorithm-based to memory-based performance. I hope the reader finds the model both simple and compelling (i.e., CMPL and CMPLing), at least in terms of its broad theoretical claims. Of course, it may also be wrong. If this is the case, then the merit of the work has been to elucidate new and previously unpredicted empirical regularities in skill acquisition, to suggest boundary conditions for applicability of the power law of practice, and to focus attention on the value of considering theoretical approaches that preclude parallel execution of two or more strategies and that are grounded in the idea that repetition of identical items strengthens a generalized representation for each item.

References

Anderson, J. R. (1983). *The architecture of cognition.* Cambridge, MA: Harvard University Press.

Anderson, J. R. (1993). *Rules of the mind.* Hillsdale, NJ: Erlbaum.

Ashcraft, M. H. (1992). Cognitive arithmetic: A review of data and theory. *Cognition, 44,* 75–106.

Bourne, L. E., Jr., Healy, A. F., Rickard, T., and Parker, J. (1997). *Strategies, strategy transitions, and the strategic basis of performance.* Manuscript submitted for publication.

Campbell, J. I. D. (1997). On the relation between skilled performance of simple multiplication and division. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 23,* 1140–1159.

Campbell, J. I. D., & Oliphant, M. (1992). Representation and retrieval of arithmetic facts: A network-interference model and simulation. In J. I. D. Campbell (Ed.), *The nature and origin of mathematical skills* (pp. 331–364). Amsterdam: North-Holland.

Carlson, R. A., & Lundy, D. H. (1992). Consistency and restructing in learning cognitive procedural sequences. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 18,* 127–141.

Carlson, R. A., Sullivan, M. A., & Schneider, W. (1989). Component fluency in a problem-solving context. *Human Factors, 31,* 489–502.

Carrier, L., & Pashler, H. (1995). Attentional limits in memory retrieval. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 21,* 1339–1348.

Cohen, J. D., Dunbar, K., & McClelland, J. L. (1990). On the control of automatic processes: A parallel distributed processing account of the Stroop effect. *Psychological Review, 7,* 332–361.

Compton, B. J., & Logan, G. D. (1991). The transition from algorithm to retrieval in memory-based theories of automaticity. *Memory & Cognition, 19,* 151–158.

Crossman, E. R. F. W. (1959). A theory of the acquisition of speed-skill. *Ergonomics, 2,* 153–166.

Crutcher, R. J. (1989). *The Role of mediation in knowledge acquisition and retention: Learning foreign vocabulary using the keyword method.* Unpublished doctoral dissertation. University of Colorado, Boulder.

Delaney, P. F., Reder, L. M., Staszewski, J., & Ritter, F. E. (in press). The strategy specific nature of improvement: The power law applies by strategy within tasks. *Psychological Science.*

Dutta, A., & Walker, B. N. (1995, November). *Persistence of the PRP effect: Evaluating the response-selection bottleneck.* Paper presented at the annual meeting of the Psychonomics Society, Los Angeles.

Healy, A. F., & Sherrod, N. B. (1994, November). *The three pronunciation distinction: A local model of linguistic categories.* Paper presented at the annual meeting of the Psychonomic Society, St. Louis, MO.

Hogg, R. V., & Craig, A. T. (1978). *Introduction to mathematical statistics.* New York: MacMillan.

Lassaline, M. E., & Logan, G. D. (1993). Memory-based automaticity in the discrimination of visual numerosity. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 19,* 561–581.

Lemaire, P., & Siegler, R. S. (1995). Four aspects of strategic change: Contribution to children's learning of multiplication. *Journal of Experimental Psychology: General, 124,* 83–97.

Logan, G. D. (1988). Toward an instance theory of automatization. *Psychological Review, 95,* 492–527.

Logan, G. D. (1990). Repetition priming and automaticity: Common underlying mechanisms? *Cognitive Psychology, 22,* 1–35.

Logan, G. D. (1992). Shapes of reaction-time distributions and shapes of learning curves: A test of the instance theory of automaticity. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 18,* 883–914.

Logan, G. D., & Etherton, J. L. (1994). What is learned during automatization? The role of attention in constructing an instance. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 20,* 1022–1050.

MacKay, D. G. (1982). The problem of flexibility, fluency, and speed–accuracy trade-off in skilled behavior. *Psychological Review, 89,* 483–505.

Moran, T. P. (1980). *Compiling cognitive skill.* (AIP memo 150, available from Xerox Palo Alto Research Center, Palo Alto, CA 94304).

Neves, D. M., & Anderson, J. R. (1981). Knowledge compilation: Mechanisms for the automatization of cognitive skills. In J. R. Anderson, (Ed.), *Cognitive skills and their acquisition.* (pp. 57–84). Hillsdale, NJ: Erlbaum.

Newell, A., & Rosenbloom, P. S. (1981). Mechanisms of skill acquisition and the law of practice. In J. R. Anderson, *Cognitive skills and their acquisition* (pp. 1–55). Hillsdale, NJ: Erlbaum.

Palmeri, T. J. (1995). *Exemplar similarity and the development of automaticity* (Tech. Rep. No. 142). Bloomington: Indiana University, Cognitive Science.

Palmeri, T. J. (1997). Exemplar similarity and the development of automaticity. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 23,* 324–354.

Pashler, H. (1993). Doing two things at the same time. *American Scientist, 81,* 48–55.

Pirolli, P. L., & Anderson, J. R. (1985). The role of practice in fact retrieval. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 11,* 136–153.

Reder, L. M., & Ritter, F. E. (1992). What determines feeling of knowing? Familiarity with question terms, not with the answer. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 18,* 435–451.

Rickard, T. (1994). *Bending the power law: The transition from algorithm-based to memory-based performance.* Unpublished doctoral dissertation, University of Colorado, Boulder.

Rickard, T. C., & Bourne, L. E., Jr. (1995). An identical-elements model of basic arithmetic skills. In A. F. Healy & L. E. Bourne, Jr. (Eds.), *Learning and memory of knowledge and skills: Durability and specificity* (pp. 255–281). Newbury Park, CA: Sage.

Rickard, T. C., & Bourne, L. E., Jr. (1996). Tests of an identical elements model of basic arithmetic skills. *Journal of Experimental Psychology: Learning, Memory, and Cognition.*

Rickard, T. C., Healy, A. F., & Bourne, L. E., Jr. (1994). On the cognitive structure of basic arithmetic skills: Operation, order, and symbol transfer effects. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 20,* 1139–1153.

Rickard, T. C., Mozer, M., & Bourne, L. E., Jr. (1992). *An interactive activation model of numerical fact retrieval* (Tech. Rep. No. 92–15). Boulder, University of Colorado at Boulder, Institute of Cognitive Science.

Schneider, W. (1988). Micro Experimental Laboratory: An integrated system for IBM PC compatibles. *Behavior Research Methods, Instruments, and Computers, 20,* 206–217.

Seibel, R. (1963). Discrimination reaction time for a 1,023 alternative task. *Journal of Experimental Psychology, 66,* 215–226.

Siegler, R. S. (1986). Unities in thinking across domains in children's strategy choices. In M. Perlmutter (Ed.) *Minnesota Symposium on Child Development: Vol. 19. Perspectives on intellectual development* (pp. 1–48). Hillsdale, NJ: Erlbaum.

Siegler, R. S. (1988). Strategy choice procedures and the development of multiplication skill. *Journal of Experimental Psychology: General, 117,* 258–275.

Townsend, J. T., & Schweickert, R. (1989). Toward the trichotomy method of reaction times: Laying the foundation of stochastic mental networks. *Journal of Mathematical Psychology, 33,* 309–327.

Appendix A

Problem Sets Used in Experiment 1

Set 1	Set 2	Set 3
3 # 17 = __	3 # __ = 32	3 # 17 = __
4 # 12 = __	4 # __ = 21	4 # 12 = __
5 # 16 = __	5 # __ = 28	5 # 16 = __
6 # 19 = __	6 # __ = 33	6 # 19 = __
7 # 15 = __	7 # __ = 24	7 # 15 = __
8 # 13 = __	8 # __ = 19	8 # 13 = __
3 # __ = 20	3 # 11 = __	3 # __ = 34
4 # __ = 29	4 # 16 = __	4 # __ = 11
5 # __ = 34	5 # 19 = __	5 # __ = 30
6 # __ = 18	6 # 18 = __	6 # __ = 25
7 # __ = 12	7 # 12 = __	7 # __ = 32
8 # __ = 27	8 # 17 = __	8 # __ = 21

Set 4	Set 5	Set 6
3 # __ = 32	3 # 11 = __	3 # __ = 20
4 # __ = 21	4 # 16 = __	4 # __ = 29
5 # __ = 28	5 # 19 = __	5 # __ = 34
6 # __ = 33	6 # 18 = __	6 # __ = 31
7 # __ = 24	7 # 12 = __	7 # __ = 18
8 # __ = 19	8 # 17 = __	8 # __ = 27
3 # 18 = __	3 # __ =	3 # 18 = __
4 # 11 = __	4 # 11 = __	4 # 11 = __
5 # 17 = __	5 # __ =	5 # 17 = __
6 # 15 = __	6 # __ =	6 # 15 = __
7 # 19 = __	7 # __ =	7 # 19 = __
8 # 14 = __	8 # __ =	8 # 14 = __

Appendix B

Problems Used in Experiment 2

True	False
E+3=H	E+3=I
N+3=Q	N+3=R
H+3=K	H+3=L
K+3=N	K+3=O
J+5=O	J+5=P
G+5=L	G+5=M
P+5=U	P+5=V
M+5=R	M+5=S
L+7=S	L+7=T
I+7=P	I+7=Q
F+7=M	F+7=N
O+7=V	O+7=W

Received August 14, 1995
Revision received July 23, 1996
Accepted January 27, 1997 ■