Journal of Experimental Psychology:
Learning, Memory, and Cognition
2004, Vol. 30, No. 1, 65–82

0278-7393/04/$12.00 DOI: 10.1037/0278-7393.30.1.65

Strategy Execution in Cognitive Skill Learning: An Item-Level Test of Candidate Models

Timothy C. Rickard
University of California, San Diego

This article investigates the transition to memory-based performance that commonly occurs with practice on tasks that initially require use of a multistep algorithm. In an alphabet arithmetic task, item response times exhibited pronounced step-function decreases after moderate practice that were uniquely predicted by T. C. Rickard's (1997) component power laws model. The results challenge parallel strategy execution models as developed to date and they demonstrate that the shift to retrieval is an item-specific, as opposed to task-general, learning phenomenon. The results also call into question the entire class of smooth speed-up functions as global empirical learning laws. It is shown that overlaying of averaged item fits on averaged data can provide a sensitive test for model sufficiency. Strategy probes agreed with strategy inferences that were based on step-function speed-up patterns, supporting the validity of the probing technique.

A basic question in the study of cognition is whether and when two mental processes can operate in parallel. Recently, much of the research in this area has focused on various aspects of long-term memory retrieval (Carrier & Pashler, 1995; Compton & Logan, 1991; Fernandes & Moscovitch, 2002; Logan, 1988, 1992; Logan & Delheimer, 2001; Logan & Schulkind, 2000; Nino & Rickard, 2003; Nosofsky & Palmeri, 1997; Palmeri, 1997, 1999; Rickard, 1997, 1999; Rickard & Bajic, in press; Rickard & Pashler, 2003; Rohrer, Pashler, & Etchegaray, 1998; Ross & Anderson, 1981; Wenger, 1999). This work is related more generally to the topics of attention and executive processing. If memory retrieval cannot take place in parallel with another retrieval or with some other cognitive process, then a performance bottleneck is implied and a decision mechanism is required when participants must make a choice. The results so far are mixed; some studies suggest that memory retrieval can proceed in parallel with other processes, whereas others do not.

One goal of this article is to address the related issue of whether a memory recall strategy can be executed in parallel with a multistep algorithmic strategy in problem-solving and cued-recall tasks. Here and elsewhere in the literature, the term *strategy* is used merely to denote a unique series of one or more mental steps toward a solution and does not necessarily have direct implications regarding intention or awareness.

An example task for which both of these strategies are available is single-digit arithmetic. To work a problem like 4 + 5, children can execute a counting strategy, such as starting with four and counting on five fingers. After sufficient practice, they can also access the answer through direct memory retrieval, bypassing the algorithm all together (e.g., Siegler, 1988; Siegler & Shrager, 1984). There are numerous other examples of tasks that can be performed by either strategy after sufficient practice, both in the laboratory (Delaney, Reder, Staszewski, & Ritter, 1998; Jenkins & Hoyer, 2000; Logan, 1988, 1992; Palmeri, 1997; Reder & Ritter, 1992; Rickard, 1997, 1999; Rogers, Hertzog, & Fisk, 2000; Schunn, Reder, Nhouynanisvong, Richards, & Stroffolino, 1997; Touron, Hoyer, & Cerella, 2001) and in educational and professional settings. In everyday life, mnemonic mediation, which people use frequently as a bootstrapping method for memory recall (Richardson, 1988), appears to give way to a more direct retrieval pathway after retrieval practice (Crutcher & Ericsson, 2000; Rickard & Bajic, 2003). In word reading, children who learn using a sequential, multistep phonics approach may be able to make a transition to single-step, whole-word processing after sufficient practice (e.g., Liang & Healy, 2002). If retrieval cannot take place in parallel with execution of the algorithm for at least some of these tasks, then important theoretical questions will arise as to the cognitive mechanism by which the strategy choice is made.

A second theoretical issue being addressed is whether the shift to retrieval is item specific (i.e., occurs independently for each item for each subject) or is task general, leading to a near immediate cascade of shifts for all items once the first shift to retrieval occurs. Most recent theorists (e.g., Logan, 1988; Palmeri, 1997; Rickard, 1997) have assumed item specificity, and the failure of retrieval-based performance to transfer to new items in most studies supports this position. However, in the only study so far to investigate this issue within the set of practiced items, Haider and Frensch (2002) concluded that shift to retrieval is a task-general phenomenon. A clear resolution to this basic theoretical issue is needed before existing models can be advanced further.

The literature review below describes candidate theories as they have been developed and fit to data to date, and the experiment is designed to directly test those models.

The Instance Theory of Automatization

Logan (1988) made the influential proposal that execution of algorithm and retrieval strategies proceeds independently and in

This research was supported by National Institute of Mental Health Grant R29-MH58202 to Timothy C. Rickard. I thank John Wixted for his comments on an earlier version of this article, Jude Mitchell and Danke Xie for their programming assistance, and Carmen Pulido for her help with data collection.

Correspondence concerning this article should be addressed to Timothy C. Rickard, Department of Psychology, University of California, San Diego, La Jolla, CA 92093-0109. E-mail: trickard@ucsd.edu

parallel on each performance trial. Provided that the participant has both the algorithm and the retrieval strategy available, he or she will initiate both simultaneously, and the latency and accuracy of the winning strategy will determine the latency and accuracy of the stated response. Logan formalized this idea, among others, in his instance theory of automatization, which was developed through both mathematical analyses and simulation. In his 1988 article, Logan started with the assumption that the algorithm response times (RTs) are drawn from a truncated normal distribution, which has a mean and variance that does not change with practice.[1] He also assumed that on each performance trial for each item, a new, item-specific memory trace (which he termed an *instance*) is created in memory. On subsequent exposures to that item, previously encoded instances race independently, along with the single algorithm racer, to retrieve the response.

Logan (1988) pointed out that the Weibull distribution can take shapes that are reasonable for memory retrieval RTs (i.e., unimodal with a right skew) and, thus, can be used as a plausible distribution model for the retrieval latency of a given instance. He assumed initially that all accrued instances are drawn from the same Weibull distribution (I term this latency distribution for a single instance the *parent* instance distribution). In this special case, the statistics of extremes dictate that the winner of a race among the multiple instances on a given trial—excluding the algorithm racer for the moment—will continue to be Weibull distributed, at least up to high practice levels (see Colonius, 1995; Cousineau, Goodman, & Shiffrin, 2002; Logan, 1995, for further discussion). The location and variance of the distribution, however, decreases as the number of racing instances increases. This generalized Weibull distribution function is:

$$f(RT) = [c/(a \cdot n^{(-1/c)})][(RT - b)/(a \cdot n^{(-1/c)})]^{(c-1)} \cdot e^{[-\{(RT-b)/(a \cdot n^{(-1/c)})\}^{c}]}, \quad (1)$$

where *a*, *b*, and *c* are parameters, and *n* represents the number of previously encoded instances. This distribution corresponds to the parent-instance distribution when $n = 1$. Logan (1988, Appendix A) pointed out that the expected value of this distribution decreases as a three-parameter power function of practice,

$$\mu_{\text{ret}} = a + b * n^{-c}, \quad (2)$$

where μ_{ret} is the population mean RT for the retrieval strategy, and *a*, *b*, and *c* take the same values as in Equation 1. The parameter *c* controls the nonlinear rate of speed-up from the starting value ($a + b$; when $n = 1$) to the asymptotic value, *a*. The instance theory also predicts that the standard deviation (σ_{ret}) of the retrieval strategy RTs will decrease with practice as a three-parameter power function (Equation 2) and that the value of the rate parameter, *c*, should be identical for μ_{ret} and σ_{ret}. These expected value predictions hold for data both at the item and group (averaged) level. It also turns out that these predictions hold to a close approximation even if several of the assumptions outlined above are violated.[2]

There is no mathematical guarantee in the instance theory that the power function will continue to govern the overall mean RT (μ_{overall}) and standard deviation (σ_{overall}) once an algorithm racer is added into the mix. However, Logan (1988) showed through simulation that, if the mean algorithm latency is similar to the mean retrieval latency of the parent-instance distribution, then the three-parameter power function still describes μ_{overall} and σ_{overall} quite well throughout practice.

In his initial investigation of the instance theory, Logan (1988) explored several tasks, including a novel alphabet arithmetic task (e.g., "D + 4 = H, true or false?"). Prior to practice, the answer in that task is obtained by reciting down the alphabet, starting with the letter given on the left, the number of times indicated by the digit, and then comparing the letter arrived at with the candidate answer presented. Thus, the example above is true. Logan analyzed his data by first computing the sample mean RT and the standard deviation over items within each subject and then averaging both of these measures over subjects within each practice block (here and elsewhere, one practice block included one randomly ordered presentation of each item). His implicit assumption was that, if and only if the instance theory is correct at the item level (i.e., for each item for each subject), it will fit well to the average data. The group-level RTs and standard deviations were in fact reasonably well fit by two 3-parameter power functions that had identical rate parameters.

The Component Power Laws Theory

Rickard (1997) proposed an alternative model of strategy execution in skill learning that he termed the component power laws (CMPL) theory. Both that theory and the instance theory assume that there is a shift to memory retrieval with practice (for earlier study of this process in children, see Ashcraft, 1981; Siegler, 1988; Siegler & Shrager, 1984), but their other core assumptions are quite different, in some cases diametrically opposed. For example, the CMPL theory assumes that, instead of laying down a new instance on every trial, a single memory association is formed (on the first trial) and strengthened (on subsequent trials) for each item. Most important for current purposes, Rickard assumed that strategy execution cannot be completed in parallel. Rather, he proposed a performance bottleneck. On each trial, the participant must select one strategy at the exclusion of the other. To avoid confusion with serial models, which would assume that both strategies are executed sequentially on each trial, the CMPL model is referred to as a strategy selection model.

Rickard (1997) modeled the algorithm as a sequence of memory retrieval steps that tap the same memory system that is used to execute the retrieval strategy. He further did not differentiate either of these types of retrieval from retrieval in traditional recall tasks, such as paired associate learning. He proposed that there is a competition between the algorithm first step (i.e., retrieving "E" from the cue "D" in the alphabet arithmetic example above) and

[1] The simplifying assumption of a constant algorithm mean may not be critical to the theory but has been made in all model fitting to date.

[2] Logan (1992) showed that if the assumption that all instances are drawn from the same Weibull distribution with the same parameter values is relaxed, mean RTs for instance retrieval will continue to follow a nearly exact power function of practice, provided that the general Weibull form of the distribution is maintained. Logan (1988) showed that instance retrieval latency will follow a Weibull distribution after a sufficient number of instances are encoded even if the parent latency distribution is not Weibull. In fact, Cousineau, Goodman, and Shiffrin (2002) showed that over a wide range of plausible parent RT distributions, the distribution for the winner of the race becomes a close approximate of the Weibull once there are about eight racers, or as few as four in some cases.

the direct retrieval strategy. Either the algorithm first step wins and the direct retrieval strategy is suppressed, or the retrieval strategy wins and the algorithm first step (and hence all subsequent steps) is suppressed. This strategy selection process takes place prior to completion of any memory retrievals for either strategy and is based on a competition between two possible interpretations of the stimulus (i.e., between the problem-level nodes in the simulation model; Rickard, 1997, p. 292). In one interpretation, the stimulus is treated as a cue for executing the algorithm first step; in the other, it is treated as a cue for executing the direct retrieval strategy. More generally, the CMPL model proposes a bottleneck in cued recall, such that only one independent response can be retrieved at a time (for more discussion of this aspect of the model, and for supporting evidence, see Nino & Rickard, 2003; Rickard & Bajic, in press).

Like the instance theory, the CMPL model proposes that the memory retrieval strategy speeds up as a power function of practice. This prediction is not generated by instance accrual, however, but by the strengthening and activation rules of the connectionist simulation model described in Rickard (1997). Because in CMPL the algorithm is, to a first approximation, a sequence of memory retrieval steps, the model also predicts algorithm speed-up with practice. Rickard proposed that algorithm speed-up can be approximated as a power function under the reasonable assumption that the power function parameters are similar for each retrieval step. The extent to which algorithm speed-up is observed for a given task, however, is determined by how much algorithm-relevant previous learning has taken place (as is the case for any mental operation that speeds up according to a power or other decelerating function).[3]

An example theoretical CMPL speed-up prediction for μ_{overall} for a hypothetical item is depicted in Figure 1 (Panel a). Here, it is assumed that there is no appreciable algorithm speed-up during the task (i.e., the previous learning for the algorithm is large, yielding nearly asymptotic performance for the algorithm at the outset of practice). The algorithm wins the competition for the first *n* trials (12 in this example) and the retrieval strategy wins for the remaining trials. The unique prediction of an independent step-function discontinuity at the transition point for each item occurs because the power functions governing the algorithm and retrieval strategies are unlikely have the same parameter values. Provided that algorithm latency is larger than retrieval latency at the transition point, such a discontinuity must be present in the expected value curve.[4]

The CMPL model predicts that the same type of strategy discontinuity will be observed at the item level for σ_{overall}, with separate power functions governing the standard deviation curve for each strategy. It also predicts that the rate parameters of the power functions for μ and σ within each strategy are identical. These predictions were not “wired into” the model, but rather fell out of other assumptions.

Following previous researchers, Rickard (1997, 1999) fit the CMPL model to group-level data, averaged over items within practice block and then over subjects within practice block. Because the strategy transition is extremely unlikely to occur on the same practice trial for every item and subject, the RT step-function discontinuity predicted at the item level will not be observed in averaged data. Rather, it will be smoothed over a range of practice trials. The result is that the average RT data should exhibit neither a step function drop nor power function speed-up but a gradual transition from the algorithm power function to the retrieval power function. For σ_{overall}, the deviation from the power function and from the item-level CMPL prediction in the averaged data is predicted to be even more pronounced, because on some practice blocks the algorithm strategy will be selected for some items,

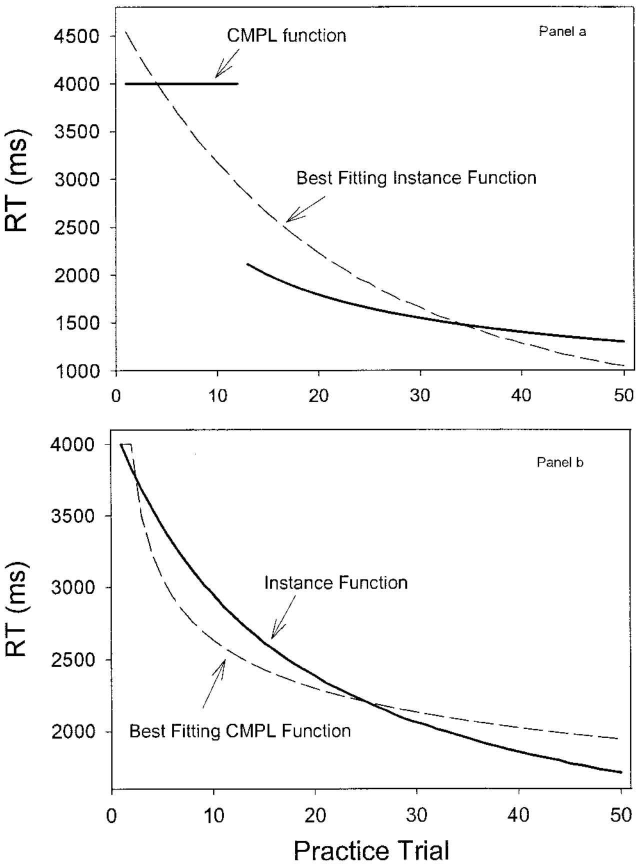

Figure 1. Panel a: An example of the component power laws (CMPL) function at the item level, with the best instance model fit overlaid. Panel b: An example of the instance function at the item level, with the best CMPL fit overlaid. RT = response time.

[3] Algorithm speed-up was observed in both experiments of Rickard (1997), but none was observed in Rickard’s (1999) reanalysis of Palmeri’s (1997) numerosity judgment task. These findings make sense from the CMPL perspective. Element counting, as in Palmeri’s task, is likely a highly overlearned skill among adults and may have little room for additional speed-up after brief warm-up, whereas the pound arithmetic and alphabet arithmetic algorithms used in Rickard’s (1997) article, for example, were relatively novel to subjects and required learning of an unfamiliar sequence of steps.

[4] The CMPL model predicts that the condition of greater latency for the algorithm than for retrieval can be achieved for any task that exhibits a shift to retrieval. Specifically, the model makes the empirically supported (Palmeri, 1997; Rickard, 1997, 1999) assumption that, for a given type of algorithm (e.g., alphabet arithmetic), retrieval latency does not increase systematically as more steps are added to the algorithm (Rickard, 1997). Thus, the model predicts that it must be possible to see step-function drops in RT at the shift point, provided that the algorithm has enough steps, making it sufficiently time consuming.

whereas the retrieval strategy will be selected for others.[5] Example CMPL μ_{overall} and σ_{overall} functions, applied to averaged data, are given in Rickard (1999; Figures 1 and 5). The results of Rickard's (1997, 1999) studies were consistent with all of the above predictions, eliminating the version of the instance theory that was fit to data in Logan (1988).

It is important to note that, although the main empirical variable under consideration here is speed-up with practice, the primary theoretical issue is about performance. This article does not speak directly to the models' different assumptions about learning (instance accrual vs. strengthening). It does speak directly to their different assumptions about strategy execution (performance).

Recent Challenges to the CMPL Account

Palmeri (1999) argued that parallel strategy execution models such as the instance theory and its successor, the exemplar-based random walk model (EBRW; Nosofsky & Palmeri, 1997; Palmeri, 1997), might be able to account for the results of Rickard (1997, 1999), given a new assumption about the values of the Weibull parameters for the retrieval strategy. Palmeri dropped the simplifying assumption of Logan (1988) that the mean retrieval latency for the parent-instance distribution is similar to that of the algorithm and, instead, suggested it might be significantly above that of the algorithm. In this version of the model, the retrieval strategy may begin to dominate performance only after the first several practice blocks (i.e., after several instances have accrued for each item). In this way, the instance theory can generate deviations from power function speed-up for both μ_{overall} and σ_{overall} that are qualitatively similar to the group-level predictions of the CMPL model, as least for the case of no algorithm speed-up. Note that this modified version of the instance theory still predicts that overall task speed-up follows a smooth function for both individual items and averaged data.

Palmeri (1999) did not fit this modified model to the data, however, so sufficiency is yet to be demonstrated. In addition, Palmeri's solution apparently cannot account for the effects of algorithm difficulty on the shape of the learning curve. Empirically, as algorithm difficulty (i.e., latency) increases, so do the deviations from the power function in the RT and standard deviation curve, just as the CMPL model predicts (Rickard, 1997, 1999). To account for this result, Palmeri's model must assume that the mean RT for the parent-instance distribution increases at a faster rate than does mean algorithm RT with each increment in algorithm difficulty. In the work to date, however, there is no evidence of any systematic increase in retrieval latency with increasing algorithm difficulty (Palmeri, 1997; Rickard, 1997, 1999; see also footnote 4).

Wenger (1999) also questioned Rickard's (1997) conclusions. Rickard used mean contrast (factorial interactions) logic to evaluate (among other things) whether strategy execution is parallel or serial in the alphabet arithmetic task. He concluded in favor of parallel strategy execution. However, that conclusion rested on assumption of selective influence in his factorial manipulations (i.e., the assumption that each factor that was manipulated affected only the mental process toward which it was directed). Wenger argued that there was empirical support for that assumption. Nevertheless, the model testing described below has an advantage in this regard, as it requires no assumptions regarding selective influence.

As noted earlier, Haider and Frensch (2002) suggested that the shift to retrieval may be an item-general phenomenon, in the sense that, once the subject begins retrieving for one item, that strategy is immediately available to all items. If correct, their model would falsify both the instance (EBRW) and CMPL theories, both of which assume that the shift to retrieval takes place more or less independently for each item (i.e., that it is a item-specific phenomenon). Haider and Frensch's argument was based on plots of mean RTs for individual subjects performing an alphabet arithmetic task. They claimed to find abrupt, step-function drops in these means with practice, a result which would be expected according to their item-general shift hypothesis. However, inspection of their Figure 3 yields little if any evidence of abrupt RT drops in the subject means. Moreover, their analytical technique for identifying the strategy shift (p. 398) in no way demonstrated an item-general shift.

Haider and Frensch (2002) also argued that the relatively fast RTs exhibited by subjects when they were transferred to new items (their Figure 3) confirmed the item-general strategy shift (if the shift is item general, it should transfer fully to new items). Yet, there was substantial slowing on the first transfer block relative to the last practice block, a finding which is at least as consistent with a reversion back to the algorithm, in turn indicating that the shift to retrieval during practice was an item-specific phenomenon (for other evidence for a reversion back to the algorithm at transfer, see Rickard, 1997; Touron et al., 2001). Haider and Frensch also showed that subjects who exhibited a transition to retrieval during practice exhibited faster learning during the transfer phase, and again they took this as evidence for an item-general strategy shift. However, that interpretation ignores the slowed RT on the first transfer block and is confounded by their post hoc categorization of subjects into shift and no-shift groups.[6] To further evaluate their proposal, I investigate the distribution of shift points over items separately for each subject. If the shift is item general, then for each subject it should occur on roughly the same practice block for all items.

Note that although the theoretical distinction between the instance (EBRW) model and the CMPL model that is addressed in this article is about performance, the theoretical distinction between the instance, EBRW, and CMPL models on one hand, and Haider and Frensch's (2002) item-general shift model on the other, is about learning. Haider and Frensch appear to have assumed that the memory retrieval strategy is learned nearly simultaneously for all items for each subject, whereas all of the other models assume that it occurs independently for each item for each subject and,

[5] For those blocks, the influence of both within- and between-strategy variance must be considered (see Rickard, 1997, 1999, for more detailed discussion of this point). The result is a "bubble" in the σ_{overall} function describing averaged data, such that σ_{overall} during the interval of practice blocks with mixed strategy trials will have particularly large values.

[6] Inspection of the practice results in their Figure 3 shows that shifter subjects (Panels A and C) were faster on the first practice block and generally sped up faster. Hence, it is not surprising that they recovered faster when presented with the new problems at transfer. These subjects may also have been better at "learning-to-learn," resulting in an accelerated shift to retrieval at transfer. That possibility, however, does not imply that that shift to retrieval was a purely item-general process. It may well be, for example, that shifters were more motivated to code for retrieval at transfer because of their success with it during practice. However, the lack of convincing subject-level, step-function drops in their Figure 3, combined with the evidence from other transfer studies that the shift process is item specific, suggests that that effect is secondary.

thus, that shift points among items for each subject can occur over a wide distribution of practice trials.

Item-Level Model Evaluation

Despite the fact that the instance, EBRW, and CMPL theories all describe psychological processes operating at the item level, to date all tests of them have relied on analyses of data averaged over items and subjects and sometimes over practice blocks within sessions. This approach has been productive, but the increased stability achieved through averaging does not come without cost (for discussions of the dangers of averaging, see Anderson & Tweney, 1997; Estes, 1956; Heathcote, Brown, & Mewhort, 2000; Kling, 1971; Myung, Kim, & Pitt, 2000). There is no guarantee that a model that is correct at the item level will still perform well when fit directly to averaged data. Conversely, there is no guarantee that a model that fits well to the averaged data is also correct at the item level. In the current case, the CMPL model's unique prediction of a step-function discontinuity at the item level, but not at the subject or group levels, makes it clear that item-level analyses are crucial to obtaining the most valid and powerful test of the models.

I took two complementary approaches to the item-level data analysis. First, I used visual inspection, combined with statistical analyses using linear regression, to place item speed-up curves into distinct categories. The main goal here was to determine the proportion of items for which RT step-function speed-up effects that are potentially consistent with a strategy selection account are visually evident versus smooth speed-up patterns that could be more consistent with the parallel strategy execution account.

Second, I fit quantitative expected value models that were based both on the instance and CMPL theories separately to each item. For both models, I assumed that the algorithm had normally distributed RTs that did not speed up with practice. The assumption of no algorithm speed-up (which is tested later) is reasonable in the current study, because subjects were pretrained on the algorithm generally, as well as on each step of the algorithm for the items to be presented during the main task. (See also Touron et al., 2001, for evidence that algorithm speed-up ceases after practice on this task.) The assumption of a normal distribution is not relevant for the CMPL expected value prediction, because the population mean, μ_{alg}, is the only required algorithm parameter if there is no algorithm speed-up. It is relevant to the instance model, however, because of the stochastic race between the two strategies. The normality assumption is unlikely to be strictly correct, as some degree of right skew is present in virtually any RT data set. I show later, however, that a right skew in the true algorithm distribution does not materially affect the fit of an instance model that assumes a normal algorithm RT distribution if the lower half of the true algorithm distribution can be reasonably described as a normal curve.

The CMPL model predicts three-parameter power function speed-up (Equation 2) for the retrieval strategy. It requires a free parameter, *shift*, which specifies whether the algorithm or retrieval strategy is used on a given trial. The model can be expressed mathematically as:

$$\mu_{\text{overall}} = \mu_{\text{alg}} \quad \text{if } n < \text{shift, (algorithm trials), and} \tag{3a}$$

$$\mu_{\text{overall}} = a + b(n-1)^{-c} \quad \text{if } n >= \text{shift, (retrieval trials),} \tag{3b}$$

where n is the trial number, shift represents the practice trial on which the strategy shift to retrieval first occurs, and a, b, and c are the parameters of the retrieval power function (Equation 2).[7] This equation is expressed graphically in Figure 1a.

In Equation 3, and in development of the CMPL model to date, the strategy shift is a one-time process; the algorithm is never again used for a given item once the initial shift to retrieval is made. This idealized and simplest case assumption is not required for strategy selection models more generally. The core requirement for that class of models is that only one strategy is used on a given trial. It is quite possible that subjects do, on occasion, revert back to use of the algorithm after the initial transition to retrieval, perhaps because memory is occasionally inaccessible for some reason. The occurrence of frequent algorithm reversions could only work against the idealized model represented by Equations 3a and 3b, however, compromising the quality of its fit.

In the overall instance theory fits described below (including both Case 1 and Case 2), the mean of the parent-instance distribution was not constrained relative to μ_{alg} or to any other parameter. Thus, Palmeri's (1999) modified instance model, which is not constrained to predict power function speed-up and which may be able to fit both group- and item-level speed-up curves, was tested.

In Case 1 of the fits, the mean of the parent-instance distribution was constrained to be equal to or less than the mean of the algorithm. In this case, the instance and EBRW models predict a single-algorithm RT on the first practice trial, followed by three-parameter power function retrieval speed-up from Trial 2 onward.[8] Hence, the Case 1 instance model can generate a step-function RT shift on the second practice trial but only on that trial. Although this case has already been eliminated as a sufficient account of the data for all items (see Palmeri, 1999; Rickard, 1997, 1999), it could still hold for some items and, therefore, must be evaluated in the item-level fits conducted here.

In this first case, the prediction of the instance model is identical to the prediction of the CMPL model when shift is set to a value of 2; the algorithm determines performance on the first trial, and a

[7] Optimal fits of Equation 3 were found independently for each item and subject by iterating through all integer values of *shift* between two and one trial past the last practice trial completed (i.e., if the last practice trial is denoted by N then the largest possible value of *shift* in the fit was $N + 1$). On each iteration, the sample algorithm mean was computed for all trials on which $n <$ *shift* (forming the estimate for μ_{alg}), and a three-parameter power function was fit by gradient descent least squares regression to all trials on which $n >=$ *shift*. The parameter estimates and the total residual sum of squares were then stored, and the next value of *shift* was evaluated in the same manner. In this fashion, the values of the five parameters, μ_{alg}, shift, a, b, and c, that yielded the smallest residual sum of squares were identified for each item. The $n - 1$ notation in Equation 3 represents the reasonable assumption, which was applied to both models, that the retrieval strategy is not available on the first trial (i.e., that there was zero retrieval strength or, in the case of instance theory, zero encoded instances on the first trial). Iteration on *shift* proceeded to the trial number immediately following the last actual performance trial (Trial $N + 1$). In that extreme case, the model prediction reduces to μ_{alg} and the shift to retrieval never occurs.

[8] Logan (1988) showed this to be true for the case when the mean of the parent-instance distribution equals the mean of the algorithm. When the mean of the parent-instance distribution is less than that of the algorithm, the algorithm will be even less influential from Trial 2 onward, and hence the three-parameter power function speed-up for retrieval should still hold.

three-parameter power function determines performance thereafter. This fit was performed for every item, and the resulting r^2 value was then stored for comparison to the result for Case 2.

In Case 2, the mean of the parent-instance distribution was set to be greater than that of the algorithm. Here, there is no known analytical solution for optimizing the fit. Thus, separately for each item, 100,000 simulations of each trial were performed for each point in a grid search over values of the five parameters (μ_{alg}, σ_{alg}, and the parameters *a, b,* and *c* of the Weibull instance distribution) until an optimal fit was obtained (see the Appendix for details). Once each item was fit for both cases, the r^2 values were compared and the case with the higher value was selected as the optimal fit for the item.

It is important to recognize that the Case 2 instance model (the one which may be viable as a sufficient account) cannot mimic the CMPL model and that the CMPL model cannot mimic the Case 2 instance model. To demonstrate this, the best Case 2 instance model fit to expected values generated from the CMPL model and the best CMPL model fit to expected values generated from the Case 2 instance model are shown in Figure 1. The best instance model fit to the CMPL function (Figure 1a) yields a smooth speed-up curve, with no discontinuities. In this example, the fit overpredicts, underpredicts, overpredicts, and then again underpredicts RTs over the course of practice. It suffers from the inability of a smooth, monotonically decreasing and decelerating function to approximate a step-function discontinuity. The instance fit provides an example of the degree of smoothness, and thus the accuracy, of the function as estimated by the Monte Carlo simulations with 100,000 simulated observations per trial (see the Appendix).

The best CMPL fit to instance model data (Figure 1b) suffers from its inability to account for smooth speed-up that does not follow a power function. It optimized its fit by shifting to retrieval on the second practice block, allowing it to predict smooth power function speed-up thereafter.[9] The optimal CMPL function also exhibited this pattern when it was fit to instance functions generated from different parameter values. Previous empirical work clearly indicates that strategy transitions are typically completed only after several (often 20 or more) trials. Hence, a finding that the optimal CMPL fit typically produces a shift to retrieval on Trial 2 would suggest that the model is false and is attempting to mimic the Case 2 instance function, or some other smooth speed-up function.

The model fitting described above should provide a strong comparative test. The assumptions are minimal, testable, and free of distorting effects that are due to item averaging that may have plagued previous work. Both five-parameter models predict three-parameter power function speed-up for the retrieval strategy in isolation (at least beyond the first few trials for the instance model). Thus, provided that the algorithm distribution assumptions can be empirically validated, there is every reason to believe that either model will fit the data extremely well if it is correct. There appear to be no other questionable auxiliary assumptions for either model, relative to assumptions made for them in past fits.

A broader family of parallel strategy execution models that are not founded on instance retrieval can be tested here by evaluating whether a step-function discontinuity is present at the item level. Specifically, it appears that no stochastic race model that predicts gradual, probabilistic replacement of the algorithm by retrieval over several trials (i.e., that predicts that several trials intervene between the first retrieval-based response and the last algorithm-based response), along with smooth, monotonically decreasing RTs for retrieval, can account for item-level, step-function RT speed-up in the general case. Simulation results discussed later demonstrate this fact explicitly for the instance model tested here.

Similarly, strategy selection accounts can be tested as a general class. All such models, regardless of their specific assumptions about the algorithm strategy, the retrieval strategy, or the strategy selection mechanism, would predict a step-function RT shift at the strategy transition point for the majority of items. It would be extremely improbable in such a model for the separate speed-up functions of the two strategies to always result in matching RTs at the point of the strategy shift (much less a smooth function), especially given a data set with a large number of items (see also footnote 4).

Tests for Validity and Reactivity of Strategy Probes

It is common in this area of work to probe subjects' strategies immediately after some fraction of the trials (Green, Cerella, & Hoyer, 2000; Reder & Ritter, 1992; Rickard, 1997; Schunn et al., 1997). Typically, subjects indicate, by pressing a key, whether they used the algorithm, retrieved the answer directly, or did something else. To the extent that these probes provide an accurate index of processing, they can be quite useful both empirically and theoretically. However, to date there have been few attempts to directly assess their validity. It remains possible that strategy reports are merely correlated with RTs, which of course decrease with practice. Another possibility is that subjects attempt to satisfy perceived task demands by gradually changing their strategy reports over the course of practice to indicate more use of retrieval, regardless of their RTs or of the true underlying strategy.

It is also possible that probes cause reactivity. For example, probes might cause subjects to adjust their speed–accuracy criterion, or they may induce an accelerated shift to retrieval. Green et al. (2000) explored this possibility by collecting strategy probes for only half of their subjects, and they found no evidence for it. Reactivity measures are intrinsically relative, however. In their design, nonprobed subjects received no filler tasks to help equate the rate of problem solving in the two groups. Although this approach is certainly reasonable, it opens the possibility of confound. For example, subjects who were probed may have experienced an accelerated shift to retrieval (perhaps the probing prompted them into use of the retrieval strategy). On the other hand, because the trial rate was presumably slower for the probed group, those subjects may also have experienced more forgetting between repetitions of an item, perhaps canceling out the former effect. Furthermore, Green et al.'s probed subjects experienced repeated task shifts, from the probing to the main task and back, whereas their nonprobed subjects did not (nonprobed subjects received no filler task in place of the strategy probe task). The suggestion here is not that their technique was flawed but that any comparison of a probed condition to a nonprobed reference condition opens the possibility that extraneous factors might be in-

[9] The retrieval RT prediction was constrained to be at or below that mean of the algorithm in this fit. This constraint resulted in the power function fit hitting parameter boundary conditions, and that fact accounts for the retrieval prediction on Trial 2 being equivalent to the algorithm mean.

volved. Hence, it may be useful to explore group comparisons under different conditions in a search for converging evidence.

Reactivity and validity are independent in principle. Probes can be valid indicators of the strategy used but still induce reactivity, or they can be flawed indicators but induce no reactivity. I thus investigated the probing technique using two separate measures. First, half of the subjects received "algorithm, direct, other" probing on half of the trials. The remaining subjects received no strategy probes and instead performed an unrelated filler task on half of the trials to roughly equate time on task and task-switching demands. Nonprobed subjects were not informed that there might be a strategy shift to retrieval. Reactivity is indicated if RTs or error rates differ for probed and nonprobed subjects.

Second, item-level analyses of data from probed subjects may provide a test for probe validity. If there are two distinct clusters of data for at least some items (one with slow RTs early in practice, and another with fast RTs late in practice), then regardless of one's theoretical perspective, it would be difficult to escape the conclusion that a strategy transition to retrieval has taken place. Hence, probe validity can be evaluated by observing the extent to which reports of the algorithm strategy are exclusively observed in the slower cluster. If there is not a close match, then the probes are not valid, and conclusions drawn from them in previous work are in question. If there is a good match, then the possibility that subjects report retrieval gradually more often with increasing practice, just to satisfy perceived task demands, can be eliminated. A good match would also show that probes can provide a generally accurate index of the underlying strategy, regardless of the possibility that strategy reports may sometimes be influenced by a correlated factor-like RT.

Method

Subjects

Twenty-four University of California, San Diego, undergraduate students participated for course credit.

Materials, Apparatus, and Procedure

Test stimuli consisted of 18 capital letters (A through R), and, for the alphabet arithmetic problems, combinations of each of these letters with either the digit 7 or the digit 9. Letters and alphabet arithmetic problems were presented in white on a black background at the center of a 14-in. monitor in a standard 3 mm × 5 mm font. Subjects were tested individually using IBM-compatible personal computers. The experiment was programmed using MEL software (Version 2.01; Schneider, 1995) and the accompanying voice key apparatus.

In the first phase of the experiment, the 10 letters (A through J) that would later be used as stimuli in the alphabet arithmetic test problems were presented one at a time. Subjects were asked to recite from the presented letter to the end of the alphabet as quickly as possible while still speaking clearly. The purpose of this task was to make the process of initiating and executing the alphabet recitation as fluid as possible during the main task. (Reciting from A is obviously trivial, but initiating recitation from other letters sometimes requires some thought prior to practice.) In this way, algorithm speed-up during the main task can be attenuated, simplifying the modeling requirements. Subjects received four blocks on this task, where each block contained one trial for each letter, with order of letter presentation randomly determined.

Each trial began with a fixation asterisk presented in the center of the screen for 500 ms. Following a 500-ms blank screen interval, the stimulus appeared in the location of the preceding asterisk. The experimenter, who was sitting beside the subject throughout the task, pressed a key when the subject reached the end of the alphabet. After a 500-ms delay, the next trial was initiated just as described above. At the end of each block, subjects were told that they could take a brief rest if they wanted to, and to press the space bar to proceed to the next block. Subjects rarely delayed between blocks.

In the second phase, a set of eight alphabet arithmetic problems was presented to subjects one at a time (e.g., "K + 7 =") and they were to speak the answer (R in this case) into a voice-key microphone as quickly as possible without making errors. The starting letters for these problems ranged from K to R. The digit addend was 7 for half of the items and 9 for the remaining items. Each problem had a unique letter response. The purpose of this phase was to familiarize the subjects with the alphabet arithmetic task in general terms, again with the goal of minimizing algorithm speed-up that might otherwise occur during the main task. There were again four blocks in this phase, each containing one randomly ordered trial for each of the eight items.

The basic trial timing was identical to that of the preceding phase. The microphone was placed about 2 in. in front of the subject's mouth. Subjects crossed their hands and elbows on the edge of the table while performing the task to keep their distance from the monitor and microphone roughly constant for all trials. After the subject made a letter response, the experimenter entered it into the computer keyboard. If the response was correct, no feedback was provided. If it was incorrect, the message "incorrect" was displayed two lines below the center of the screen, along with the correct answer, centered two lines below the "incorrect" display. Subjects were informed about this feedback arrangement prior to the task. If the voice key failed to trip as the subject vocalized his or her response, or if it tripped too soon (e.g., if the subject vocalized an "uh," before answering), the experimenter entered the subject's response, but then pressed the "-" key (as opposed to the "+" key, which was entered in the same manner for all trials on which there was not a voice key problem). These entries provided a record of trials on which voice key failures occurred.

In the main alphabet arithmetic task, subjects performed 10 new alphabet arithmetic problems just as in the previous phase. The presented letters were A through J. For half of these items the addend was 7 and for the other half it was 9. Each problem again had a unique answer letter. Subjects received up to 50 blocks of practice on this task, with the experiment ending before that point if time expired.

Half of the subjects received strategy probes after half of the trials in each block, with probed trials randomly determined, subject to the constraint that each item was probed once over each two-block sequence. On probed trials, subjects first made their response to the alphabet arithmetic problem and were then presented with a message instructing them to select the one of three buttons on a button box. The left-side button was marked "A" for algorithm, the next button was marked "R" for retrieval, and next button was marked "O" for other. Subjects were instructed to select A if they used the algorithm that they were taught, to select R if they retrieved the answer from memory (much as they might retrieve the answer to "2 × 4" from memory), and to select O if they used neither of these strategies or were unsure.

Nonprobed subjects were not informed about the possibility of a strategy shift to retrieval. For these subjects, a filler task was presented on half of the trials instead of strategy probes, following the same presentation rules as stated above for the probe trials. In the filler task, subjects were presented with a simple question, such as "square above triangle?" and the corresponding shapes were presented, one above the other, below the question. Subjects made a true or false judgment by pressing one of two identified buttons on the button box. Preliminary analyses showed that the average time to complete this task was similar to that required for probed subjects to complete the strategy probe task. The filler task was solely intended to roughly equate the task-switching requirement and the total time to complete the experiment for probed and nonprobed subjects.

The alphabet arithmetic task outlined above differs from previously used versions in that it requires subjects to produce a letter as a response. In previous studies, the problems were presented as statements with a candidate answer (e.g., "A + 7 = H"), and subjects were to make a true or false evaluation. Presentation of the problem–answer combinations for verification raises the possibility of three different types of strategy shifts with practice. First, there could be a shift from execution of the algorithm to retrieval of the correct letter response, followed by a comparison to the candidate answer to make the true or false decision. Second, there could be a sequence of two shifts for each item during practice: (a) a shift from execution of the algorithm to retrieval of the correct letter response as above, followed by (b) a secondary shift to direct retrieval of the true or false response, bypassing the need to retrieve the correct letter response. Third, there could be a transition directly from using the algorithm to direct retrieval of the true or false response, bypassing retrieval of the correct letter response altogether. These different strategy shift possibilities could complicate interpretation of results. For this reason, I chose to use the simpler letter production task. There appears to be only one possible type of strategy transition for that version of the task, the transition from use of the algorithm strategy to use of the retrieval strategy.

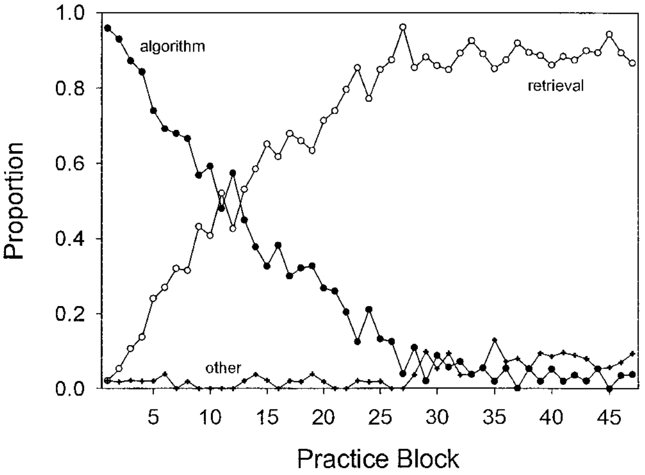

Figure 2. The proportion of trials on which each strategy was selected as a function of practice block.

Results and Discussion

Unusual data from 1 subject were not included in any analyses below.[10] The warm-up task data were also not analyzed. Accuracy on the main task, averaged over all items within subject, and then over all subjects within block, ranged from 90.2% on Block 1 to 98.2% on Block 47 (the last block that all subjects completed). There was minimal effect of addend size on accuracy (the means overall were 95.1% for addend 7 items and 93.8% for addend 9 items). Accuracy on the first and last test blocks was nearly identical for probed and nonprobed subjects. However, accuracy increased at a faster pace for probed than for nonprobed subjects.

For probed subjects, the average accuracy was 94.6% when they reported using the algorithm. Algorithm accuracy did not change with practice (beyond the first practice block). There was no visual trend, and the slope in a linear prediction of mean accuracy by practice block was nonsignificant, $t(45) = 0.38$.[11] When probed subjects reported using retrieval, their mean accuracy was 97%. In this case, the slope was significant, $t(45) = 2.42$, $p > .02$, indicating about a 3% increase in accuracy for that strategy over practice blocks.

All RT analyses were performed for correct trials only. Voice-key errors occurred on 0.73% of correct trials. These trials were also eliminated prior to the analysis. For all analyses of averaged data, RTs were averaged first over items within each practice block for each subject and then over subjects within block.

On the first practice block, the mean RT for addend 9 problems was about 1,000 ms slower than that for addend 7 problems. By the 47th practice block, that RT difference had reduced to about 300 ms. Analyses of the strategy probe data (see below) suggested that this residual RT difference was due primarily to an incomplete transition to retrieval for some items. The finding that the transition to retrieval was not complete is not surprising, given previous work suggesting that about 60 practice blocks are needed for that to occur (e.g., Rickard, 1997).

Strategy reports for probed subjects are summarized in Figure 2, collapsed over addend level (for which there was no visually evident main effect or interaction). Each point gives the proportion of correct trials on each practice block, averaged over items and subjects, for which the specified strategy was used. As expected, in nearly all cases, subjects reported using the algorithm on the 1st block. By the 47th block, retrieval was reported on the majority of trials. The proportion of "other" responses was very low initially and reached a peak of 8% on about Block 35.[12]

To test for strategy probe reactivity in the RTs, the average data (collapsed over addend level; there was again no interaction) were plotted separately for probed and nonprobed subjects, as shown in Figure 3, along with the best fitting three-parameter power functions for reference. Reactivity is clearly indicated. Although probed and nonprobed subjects had similar RTs at the beginning of practice, there is a large divergence in the RT curves that peaks at around 800 ms on Block 25. This result remained obvious when RTs for the two groups were shifted so that they matched on the first test block. This pattern, along with the faster increase in accuracy with practice for probed subjects, suggests that the shift to retrieval occurred earlier for probed subjects. Nevertheless, the deviations from power function speed-up seen in previous averaged data are clearly evident for both groups. It appears that strategy transitions took place regardless of whether subjects'

[10] This subject exhibited at U-shaped RT pattern for nearly every item. RTs were in the range expected for the algorithm initially, then fell into a range that is consistent with initial retrieval trials (around 2,000 ms), and then increased again toward the end of practice to levels seen during the first few blocks. Because no theory predicts such results, I assumed that this subject became uncooperative or unusually fatigued, and removed these data from further consideration. No other subjects exhibited this effect. It should be noted that unusual subject-level results of this sort may well have occurred before. In nearly all previous work, data have been analyzed only after averaging over subjects. Atypical results for occasional subjects could well be masked by that averaging.

[11] For this and all subsequent tests, alpha was set to .05.

[12] This general pattern replicates that observed by Rickard (1997). The fact that the "other" strategy was selected rarely during the first 25 blocks suggests that it does not reflect some type of modified algorithmic process but is somehow associated with the retrieval strategy. I evaluated this possibility by comparing other and retrieval RTs for the interval between Blocks 29 and 47, over which both strategies were relatively common. The RTs did not differ meaningfully. With the caveat that the other responses cannot be conclusively diagnosed, those trials were eliminated from later analyses in which RTs were evaluated separately for specific probe categories.

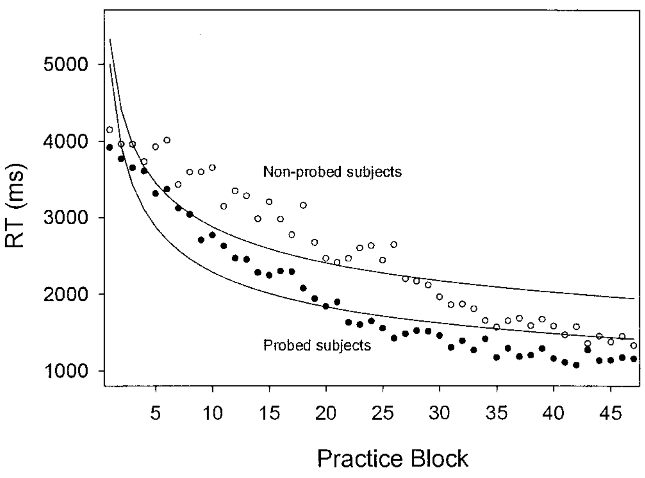

Figure 3. Group-level mean response times (RTs) plotted separately for probed and nonprobed subjects as a function of practice block. Best fitting three-parameter power functions are overlaid.

strategies were probed and regardless of whether they were made explicitly aware of the direct retrieval strategy.

The CMPL model does not predict but is consistent with this reactivity effect. The model incorporates subgoals to perform a particular strategy, and it is assumed that activation of these subgoals is partially under conscious control. Subjects who were probed may have been biased toward attempting retrieval. This effect could be implemented in the model by amplifying the retrieval subgoal activation, in turn increasing the competitiveness of that strategy and facilitating the shift to retrieval.

These results contrast with those of Green et al. (2000), who found no reactivity from probing in a similar task. It is unclear why the results from these two experiments differ so dramatically, but the most likely reason is the differing treatment of the nonprobed group. Green et al. did not include a filler task for their nonprobed group, whereas I did. The potential limitations of Green et al.'s approach were discussed earlier. A potential limitation of my approach is that the filler task may have somehow slowed main task performance or the transition to retrieval specifically. It is possible, for example, that the effects of a task switch from the filler task to the main task for nonprobed subjects in this experiment were more detrimental to main task performance than switching from the probing task to the main task for probed subjects.

It appears that the reactivity issue may be more difficult to resolve than might have been anticipated, because in any comparison of probed and nonprobed subjects, there will be some undesired secondary dimension along which the two groups are treated differently. Nevertheless, given the potential value of the technique, more research is warranted.

Visual and Statistical Categorization of the Item-Level Data

In the usual approach of analyzing averaged data, each point of an RT plot represents an average of tens or even hundreds of observations. Consequently, the effects of occasional extreme outliers are suppressed and may go unnoticed. At the item level, such outliers can be obvious and influential and must be dealt with explicitly. Preliminary analyses using item-by-item visual inspection revealed two types of outliers that were rare and easy to deal with. First, there were six outlier trials that occurred early in practice, apparently prior to the strategy transition, which had RTs that were more than three standard deviations below those on surrounding blocks and that were much smaller than all other RTs for the item. These trials constitute spurious outliers by any standard and neither model can accommodate them. They were removed from further analyses. Second, there were 43 trials with RTs that were far greater than any other RTs for the item, including RTs early in practice, when the slower algorithm was the likely strategy. These observations were also removed from further analysis. The only exception was unusually slow RTs that occurred on the first practice block. In a generally decelerating speed-up scenario, unusually slow RTs on the first block are expected and need not reflect spurious effects such as distraction. The total number of outliers in these two categories (49) constituted only 0.45% of the data.

As a first step in descriptively categorizing the item-level data, a practice block was used to predict RT for each item using least squares linear regression. If the p value for a negative slope was greater than .2, an item was categorized as a *no speed-up* item. Nineteen of the 211 items (9%) exhibited no speed-up by this criterion. Tests of these items using a three-parameter power function yielded nearly identical results, with no meaningful improvement in r^2. Eight of these items were from strategy-probed subjects. For those items, subjects reported using the retrieval strategy on a total of only 10.5% of probed trials. By contrast, for the remaining probed items, which did exhibit speed-up by the same regression criterion, retrieval was the reported strategy on 68.4% of probed trials. The RTs for the no speed-up items were consistent with the probe results, being quite slow throughout practice.[13] These results complement the data from the no-transition subject in Experiment 1 of Rickard (1997). The results also provide the first source of evidence that the goal of no algorithm speed-up in this design was achieved. Note that neither model requires that a strategy transition to retrieval takes place for all items. Because the models cannot be discriminated for these items, they were removed from all subsequent analyses.

Second, there were five items for which there was a single slow RT on the first trial, followed by a marked step-function RT drop on the second trial. For these items, subjects apparently made a shift to purely retrieval-based performance on the second trial. From the perspective of the instance model, these are obvious Case 1 items. As noted earlier, the CMPL and instance models also do not make different predictions for this case. As such, these items were also removed from further analysis.

There was a third set of 60 items (29.1% of the 206 items still under analysis) for which RTs decreased on a roughly smooth speed-up function (they are referred to as *smooth speed-up* items). Example items are plotted in Figure 4. Note that these are plots of individual RTs, with no averaging. Thus, the

[13] RTs for no speed-up items were averaged into five consecutive sets of 10 tests blocks (to reduce noise). The means for each of these 5 "super-blocks," averaged over all 19 no speed-up items, were 3,154, 2,931, 3,009, 3,018, and 3,014 ms, respectively. In contrast, for the other 211 items, the corresponding means were 3,600, 2,617, 1,908, 1,428, and 1,178, respectively. RTs for no speed-up items are therefore consistent with use of the algorithm throughout practice.

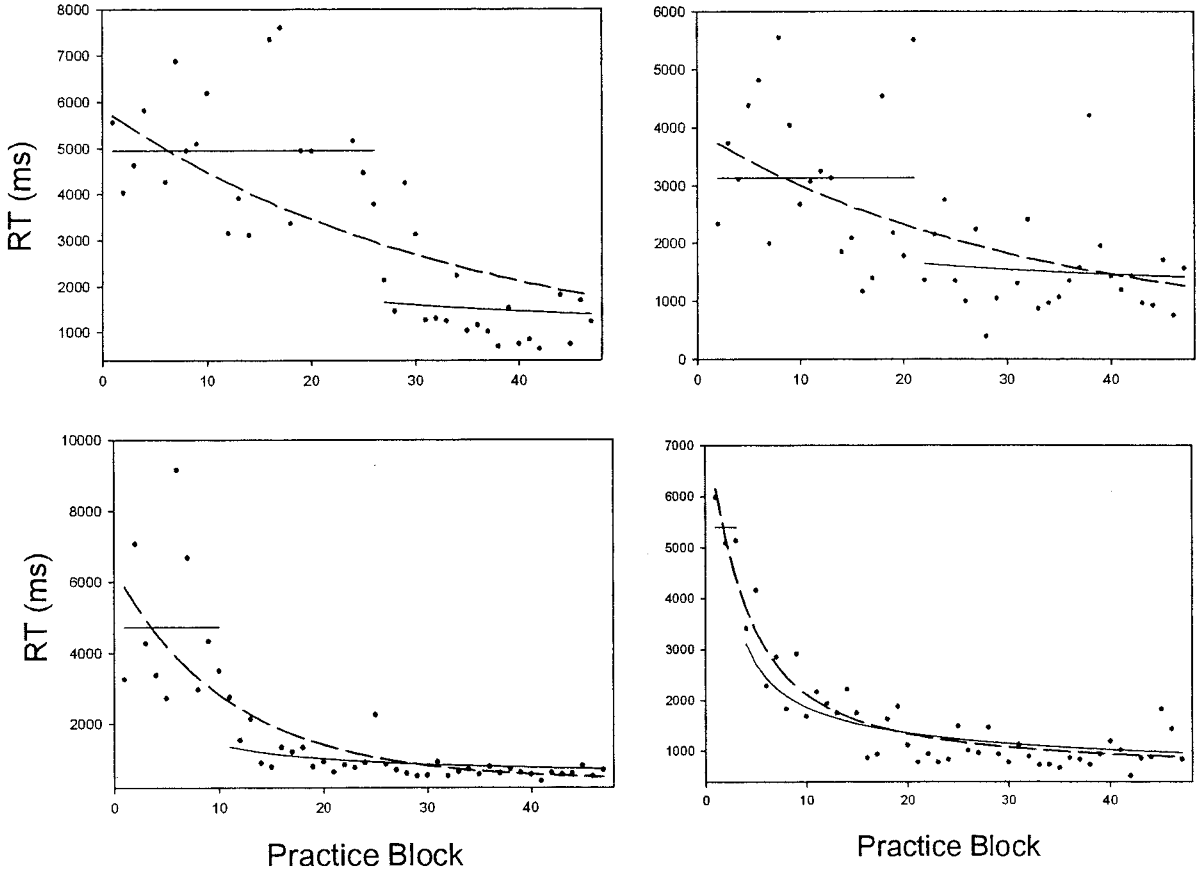

Figure 4. Example data from four items in the smooth speed-up category. The solid line represents the component power laws (CMPL) model fit, and the dashed line represents the instance model fit. RT = response time.

substantial variability in the plots is expected. Some of these items exhibited tendencies toward clustering of the sort expected by the CMPL model (see the left-side panels), but at this stage I judged that these effects might be observable on occasion even if the underlying population speed-up function is smooth. For these items, stronger theoretical conclusions might be achievable through model fitting (the model fits shown in Figures 4–6 are discussed later).

The remaining 146 items (71.1% of the data under analysis) exhibited visual trends that were more consistent with the CMPL model. I classified 102 of these items as Type 1 cluster items. Four examples are shown in Figure 5. These items all exhibited roughly step-function speed-up patterns. However, there were occasional slow RTs late in practice, as well as occasional RTs near the point of the apparent strategy shift that were difficult to classify confidently into either algorithm or retrieval strategies.

The 44 remaining items were classified as Type 2 cluster items. All of these items exhibited visually striking and pure step-function RT decreases (see Figure 6), with no unusually slow RTs after the initial shift to retrieval. For all of these items, there was a substantial RT band between the two data clusters in which there were no observations. Overall, these graphical results favor the CMPL account.

An Item-Level Probe Validity Test

To explore the validity of the strategy probes, probe results were compared with RT patterns for Type 2 cluster items. The vast majority of the time, probe reports (algorithm or retrieval) matched expectations on the basis of the data clustering. There were no trials on which subjects reported using the algorithm when the RT was in the retrieval cluster, and there were only seven trials on which retrieval was reported when the RT was in the algorithm cluster, yielding a total misclassification rate of only 1.2%. It follows from this result that, in the vast majority of cases, the strategy shift point as identified by the CMPL fits to the RTs matched the strategy shift point as indicated by the strategy probes. Just as there was a discrete shift in RTs for most items, there was a corresponding discrete shift in the reported strategy.

These results suggest that, as a rule, strategy probes probably have good validity, although with the possibility of marked reactivity. This finding may have methodological implications for other cognitive research. It appears that a rich, generally valid, and relatively cheap source of data is being ignored across a variety of task domains. Ericsson and Simon (1993) have made this point quite forcefully for verbal reports generally. Although they discouraged use of strategy probes with preset categories, they had little direct evidence from which to make that recommendation

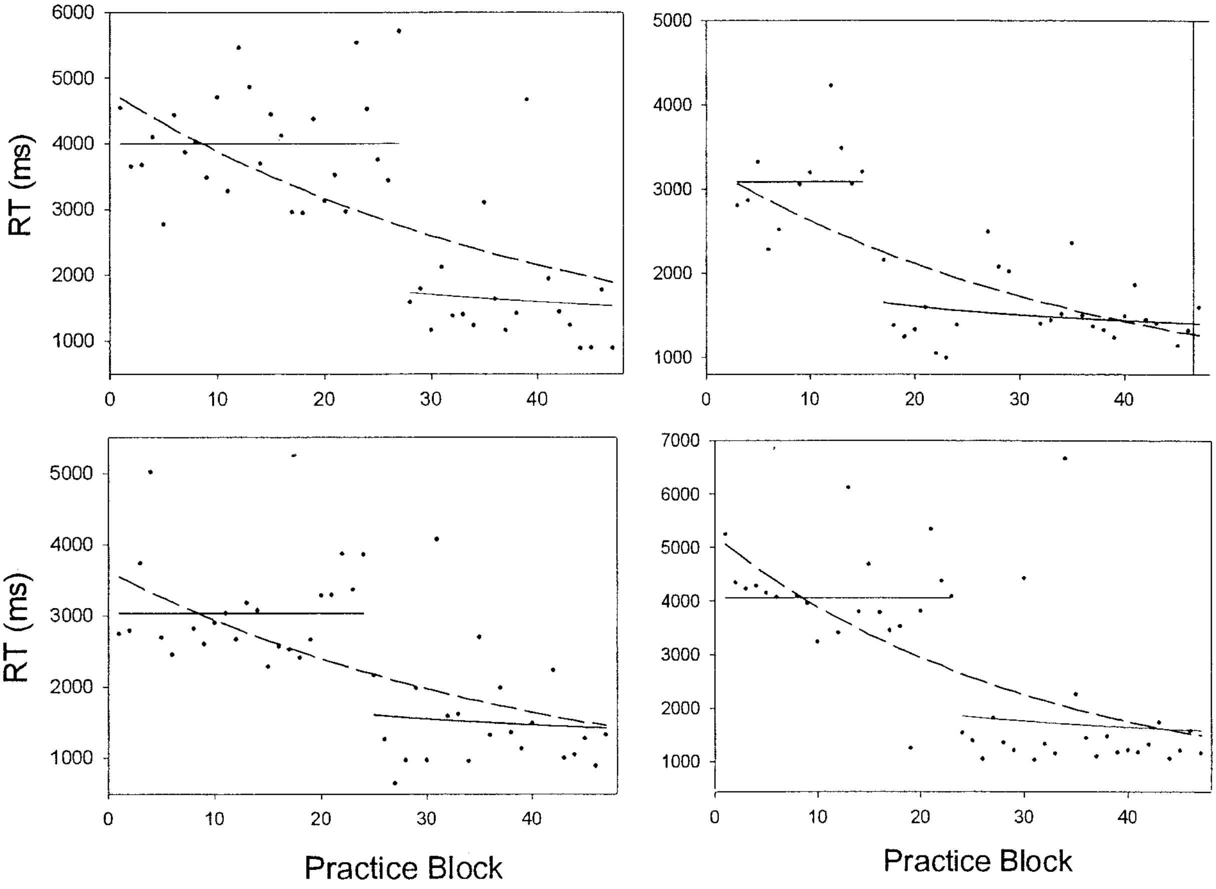

Figure 5. Example data from four items in the Type 1 cluster category. The solid line represents the component power laws (CMPL) model fit, and the dashed line represents the instance model fit. RT = response time.

with respect to the specific probing scheme used here. The current results suggest that probes can be sufficiently valid to make inferences even at the item level. A practical consideration is that, from the standpoint of data analysis, strategy probes are far easier to deal with than are recorded verbal reports.

Item-Level Model Fits

Although the item-level plots generally favor a strategy selection model such as CMPL, for the smooth speed-up items it was not possible to confidently differentiate between the models visually. In addition, unless the models are fit formally, the degree to which either can provide a sufficient account of the data cannot be determined. For these reasons, formal item-level models fits were conducted.

Preliminaries. As noted earlier, an assumption common to both quantitative models tested here is that the algorithm strategy does not speed up with practice. The finding that the no speed-up items exhibited slow RTs and had strategy reports indicating use of the algorithm throughout practice already supports that assumption. Now that visual item analyses have been performed and the validity of the strategy probes demonstrated, additional approaches to testing for algorithm speed-up for the remaining items are available. First, for each item, probe trials on which subjects reported using the algorithm were fit with a linear least squares regression line (for an item to be included in this analysis, the algorithm probe category must have been selected by the subject at least three times on correct trials). A t test on these 88 item-level t ratios for the slope was not significant, $t(87) = 0.85$, suggesting no speed-up. Second, linear regression analyses were also performed on all RT data that fell within the algorithm cluster (including both probed and nonprobed trials and subjects) for all Type 2 cluster items. Averaged over these 44 items, the mean t ratio for the slope estimate was again not significant, $t(43) = 1.13$, $p = .26$. The assumption that there is no algorithm speed-up with practice appears to be justified. Inspection of Figures 4 through 6 further supports this conclusion.

The assumption that algorithm RTs are normally distributed is not relevant to the CMPL fits but is potentially relevant to the instance model fits. I evaluated normality by first standardizing the Type 2 cluster algorithm data separately for each item to have a mean of zero and a standard deviation of 1.0. These standardized data were then pooled into a single distribution. If the algorithm RTs for each item are normally distributed, then this pooled, z-score transformed data should be standard normal. A frequency polygon indicated right skew in this distribution, violating normality. However, the lower half of the distribution followed a normal curve well, on the basis of a partial cumulative distribution fit. This suggests that treating the algorithm data as being normally distrib-

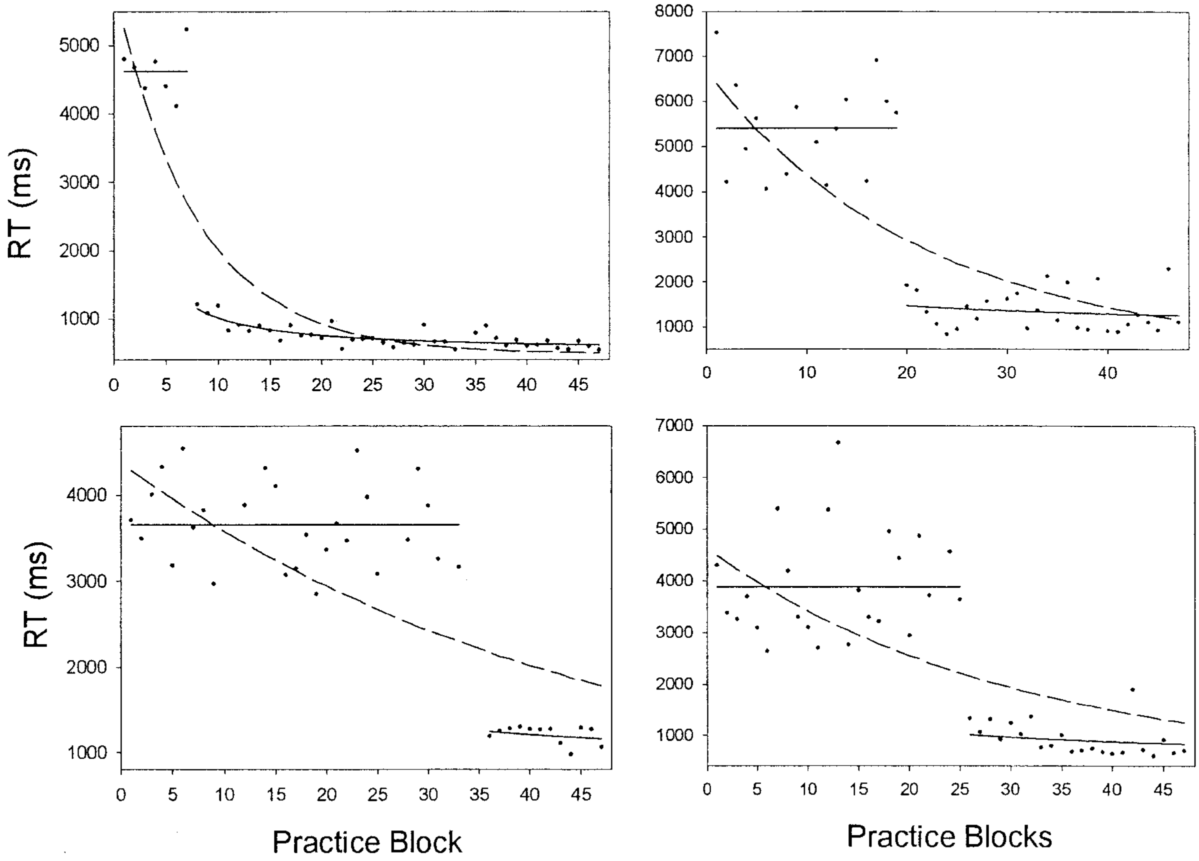

Figure 6. Example data from four items in the Type 2 cluster category. The solid line represents the component power laws (CMPL) model fit, and the dashed line represents the instance model fit. RT = response time.

uted was sufficient for current purposes. In the instance model, very slow algorithm completion times are unlikely to determine performance on most trials because they will be stochastically filtered out during the race with the memory instances.[14]

The parameters of the retrieval power function in the CMPL model and of the Weibull distribution in the instance model (which in turn generates the power function speed up; see Equation 2) had the following bounds: $c > 0$, $b > 0$, and $350 < a < 600$. The constraints on the asymptote parameter, a, were motivated by two factors. First, to my knowledge, mean RTs below about 350 ms have never been observed in cued-recall tasks. Rickard and Logan (2003) found that even simple vocal naming times for letters were no faster on average than about 350 ms, even after 100 trials of practice for each letter. Furthermore, there was no speed-up in response latency over those practice blocks (beyond the second one), suggesting, not surprisingly, that letter-naming latency in adults is essentially asymptotic. In the current experiment, there were only two observations below 350 ms, both of which were removed as obvious outliers by criteria noted earlier. The upper bound of 600 ms was motivated by previous findings that mean retrieval RTs can fall below that value on average with sufficient practice. In the current experiment, a total of 348 observations had RTs less than 600 ms, spread over 18 of the 23 subjects. The fact that this level of performance was achieved after only 50 or fewer repetitions for each item should leave little doubt that asymptotic expected RTs would reach or fall below this value for all items given enough practice.

Fits. Four examples of best fitting instance and CMPL functions to item data are shown in Figures 4, 5, and 6. In all cases, RTs

[14] To test this claim, an RT expected-value speed-up curve was generated for a six-parameter version of the instance model (based on 100,000 simulated observations per trial), treating the algorithm distribution as a convolution of a normal and an exponential distribution. The parameter values were $\mu_{\text{alg}} = 4{,}000$, $\sigma_{\text{alg}} = 800$, $\tau_{\text{alg}} = 1{,}000$, $a = 500$, $b = 100{,}000$, and $c = 2.0$, where τ_{alg} is the rate parameter for the exponential component of the algorithm distribution and is responsible for generating the right skew in that distribution. The other five parameter values are typical of results from the instance-function fits to the data. The value of τ_{alg} was chosen to roughly match the degree of right skew seen in the algorithm data. Next, the five-parameter instance model (i.e., with no τ_{alg} component) was used to fit the RT expected-value data that were generated from the six-parameter model. All parameter values in this fit remained the same as for the six-parameter curve, with the exception that μ_{alg} was increased to 5,000 ms, which is the mean of the algorithm distribution in the data generated from the six-parameter instance model (i.e., the mean of the exponential–normal convolution, given by $\mu_{\text{alg}} + \tau_{\text{alg}}$). No attempt was made to further optimize any parameter values. The r^2 between these two expected-value speed-up curves was better than .999. Thus, the five-parameter version of the instance model is sufficient for obtaining the best possible fits to a very close approximation, even given the right skew in the true distribution for the algorithm data.

for all correct trials for the item are depicted. In the majority of cases, the visual fit of the CMPL model was better than that of the instance model, often strikingly better. The average r^2 for the CMPL model was .688, versus .552 for the instance model, and the CMPL model yielded a higher r^2 for 201 of the 206 items for which the models could potentially be differentiated (i.e., excluding the 24 no speed-up and second trial strategy transition items). A binomial test on this result, under the null hypothesis that the two models provide equivalently good fits, was highly significant, $p < .00001$. The outlier filtering described earlier was not a factor in this outcome. If the 36 items involved in that filtering are removed, then at least 165 of the 170 remaining items (for which there was no filtering of any sort) were still better fit by CMPL. By the same reasoning, the CMPL fit advantage was also not idiosyncratic to either the addend size or to whether a subject was probed.

For the smooth speed-up items, the r^2 advantage for the CMPL model was less pronounced (.572 vs. .511 for the instance model) but was still nearly ubiquitous (it fit better for 58 of the 60 items; see Figure 4 for examples). Inspection of the model fits explains this finding. For some items, there were subtle step-function–like speed-up patterns (see the two left side panels in the figure). A related pattern for these items was an initial series of RTs with no speed-up followed by a steady RT reduction over the remaining blocks. The CMPL model has no difficulty fitting such data. The instance model does encounter difficulty, however. It must attempt to fit the faster RT data toward the end of practice but doing so requires that its smooth function has decreasing RTs throughout practice. Hence, that model tends to underpredict most of the nondecreasing RTs for the later algorithm trials.

For the Type 1 cluster items, a more pronounced advantage for CMPL became apparent, with a mean r^2 of .679 versus .529 for the instance model (see Figure 5 for examples). The CMPL model fit better to 99 of the 102 items. For Type 2 cluster items (Figure 6), the fit advantage for CMPL was striking for all items, with a mean r^2 value of .865, versus .660 for the instance model. Not surprisingly, the greater the visual evidence was of clustering and of a marked step-function RT drop, the greater the fit advantage was for the CMPL model.

For Type 2 cluster items, there were several cases in which the optimal instance fit almost completely overshot the lower left RT cluster (see Figure 6). This fit pattern is understandable when considering that it becomes more pronounced as the number of nondecreasing RTs in the algorithm cluster increases. The greater the number is of nondecreasing algorithm RTs prior to the step-function shift, the more the optimal instance fit must be weighted toward those trials and the greater the overprediction is of the subsequent fast RT trials.

Is the Shift Item General or Item Specific?

Haider and Frensch's (2002) proposal that the shift to retrieval is an item-general phenomenon was tested by computing the range and standard deviation of shift points over items, separately for each subject. The shift point in these analyses was identified by the value of the shift parameter in the CMPL model fits. The mean range of the shift points over subjects was 22.3 blocks, and the mean standard deviation was 8.1. Thus, the shift to retrieval is not all-or-none for each subject, as predicted by an item-general shift model. Rather, it appears to occur independently for each item, as predicted by both the instance (EBRW) and CMPL theories.

A Group-Level Sufficiency Test for the Item-Level Model Fits

One valuable but not previously recognized consequence of fitting models at the item level is that the average of those fits can be overlaid on the averaged data as a test for sufficiency. In this way, the increased stability of data that is due to averaging can be obtained, free of the potential for functional bias that is due to averaging the data prior to fitting the model. If a model provides a correct, unbiased fit at the item level, the match of the averaged fits to the averaged data should be excellent. On the other hand, if a model misfits in subtle ways at the item level, that fact may be easier to see in the averaged plot. The results, with the data, as well as the predictions of both models, averaged over items and subjects in the same manner as in the earlier averaged plots, are shown in Figure 7.

Panel A of Figure 7 shows these fits for all 206 items. The CMPL fit is good ($r^2 = .9945$), reflecting its good fits at the item level. Figure 7b shows the fits for Type 2 cluster items, which by definition had no outliers late in practice that might have influenced and distorted the fits. Again, the CMPL fit is quite good ($r^2 = .9940$; slightly lower in this case because fewer datum were

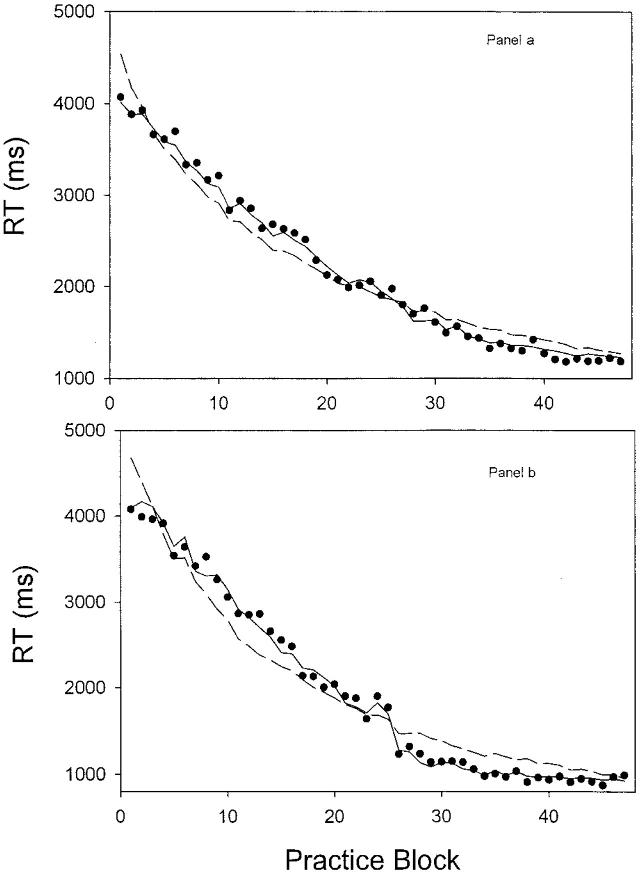

Figure 7. Panel a: Overlays of averaged item-level fits of both the component power laws (CMPL) and instance models on averaged data. Panel b: Overlays of averaged item-level fits of both the CMPL and instance models on averaged data (Type 2 cluster items only). In both cases, the solid line represents the CMPL fit, and the dashed line represents the instance theory fit. RT = response time.

involved in the fit). The instance theory fits in both panels of Figure 7 show significant distortion, overpredicting, underpredicting, and then again overpredicting the data. This distortion is obvious despite r^2 values of .9654 for the fit to all data and .9495 for the fit to Type 2 cluster items. The source of this effect is easily seen in the item-level fits for the Types 1 and 2 cluster items. Panel B of Figure 7 confirms that this effect is not attributable to leveraging from slow RTs that occurred late in practice for some of the smooth speed-up and Type 1 cluster items.

The procedure outlined above multiplies the number of free parameters by a factor of 206 (the number of items) compared with simply fitting each model a single time to data that have already been averaged over items and subjects, as in past studies. Nevertheless, the procedure turns out to be far more sensitive to problems with the model fits. This fact can be appreciated by comparing the fits discussed above with fits of both models to the averaged data, in the traditional manner. Here, each model was fit only once, to data that had already been averaged over items and subjects. For the CMPL fit, I used a five-parameter version of the model that is similar to that specified in Rickard (1999) as an approximation for fitting average data. The equation is:

$$\mu_{\text{overall}} = \mu_{\text{alg}}(p) + (a + b(n-1)^{-c})(p-1), \qquad (4)$$

where $p = \mathrm{e}^{-r^*(n-1)}$, and represents an educated guess for the proportion of trials on which the algorithm was used as a function of test block.

The resulting fits are shown in Figure 8. For the CMPL model, the two methods of fitting the average data yielded nearly equivalent results, with the r^2 (.9913) being only .0032 points below that of the preceding analysis. For the instance model, however, the single fit of the model to the average data was far better than the earlier fit of the averaged item-level fits to the average data (r^2 = .99). Indeed, the fit advantage for the CMPL model vanished in this case. It appears that, when the instance model is fit once to averaged data, its free parameters are able to adjust their values to match the data. In contrast, when the models are first fit to individual item data, this optimization process takes place for each item, and the parameters are then locked. The subsequent fits to the averaged data are then parameter free and are not privy to any direct information regarding the shape of the averaged speed-up curve. As such, any systematic bias in the item-level fits are reflected at the averaged level. This result is, of course, not idiosyncratic to the instance model but could occur for any model that does not fit well to a significant proportion of the item-level data.

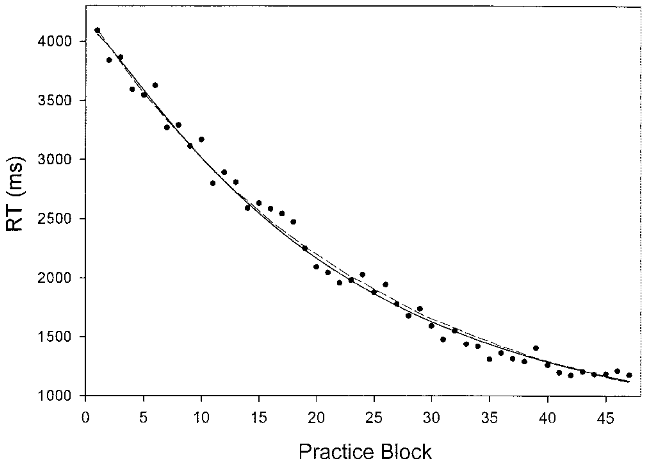

Figure 8. Fits of both models to averaged data. The solid line represents the component power laws (CMPL) fit, and the dashed line represents the instance theory fit. RT = response time.

The results above suggest that item-level fits, combined with a group-level sufficiency test, should become a standard approach to research on learning curves and skill acquisition. Indeed, it may be that testing of any parametric model can benefit from this approach. A model can be fit first to each of the smallest subsets of data to which it applies (i.e., to either item- or subject-level data), and then both types of averaged fits discussed above can be generated. It may be that many models that pass the test for the traditional single fit of a model to averaged data would fail when the average of the item- or subject-level fits are overlaid on the averaged data. Note that if a model is exactly correct at the item level, with no bias in its fit, it should not be possible for a single fit of that model to the averaged data to be better than the fit of the averaged item-level fits to the averaged data. The method thus provides a good benchmark test for sufficiency. It is a more valid and stringent test than fitting of the model once to averaged data. Furthermore, if a model passes this test and also exhibits extremely good fits as measured by r^2, then it is unlikely that any future alternative model could fit the data meaningfully better at either the item or group levels.

The Source of Speed-Up

Because the CMPL model provides a good fit to the data, it is appropriate to use it as a framework for exploring how well speed-up can be characterized simply as a pure step-function strategy shift, which appears to have been the dominant learning effect. With this goal in mind, a three-parameter version of the model was fit to each item. In this model, the three-parameter power function for the retrieval strategy was replaced by single parameter, μ_{ret}. Hence, the model parameters were μ_{alg}, shift, and μ_{ret}. The mean r^2 for fits of this simplified model was .664, only .026 points below that of the five-parameter CMPL model. The average predicted speed-up in the three-parameter model fits ($\mu_{\text{alg}} - \mu_{\text{ret}}$) was 2,631 ms, compared with the average predicted speed-up in the five-parameter model fits of 2,794 ms (computed as μ_{alg} − the predicted retrieval RT on the 47th practice trial). The three-parameter model thus accounted for a striking 96% of the speed-up predicted by the full five-parameter model (the mean predicted speed-up for the retrieval strategy in the five-parameter model was 349 ms). Clearly, speed-up was dominated by the step-function RT shift.

The fits of the three-parameter CMPL model were better than those of the five-parameter instance model for 175 of the 206 items, an extremely significant result using a simple binomial test. This finding should eliminate any remaining concern that the five-parameter CMPL model may have fit better because of greater flexibility.

Despite the surprisingly good fits of the three-parameter version of the CMPL model, the full five-parameter version still fit better to 181 of the 206 items (it fit equally well to the remaining items, as it must because the three-parameter model is nested within the five-parameter model). Given the strong a priori expectation of within-strategy speed-up for memory retrieval, it would be inappropriate to dismiss this finding as an artifact of overfitting. If one accepts the existence of retrieval speed-up, then a nonzero asymptote parameter is in principle necessary (RTs would not, afterall, drop to zero at asymptote, although in most cases the asymptote can be ignored with negligible decrement in quality of fit; that was the case here for the CMPL fits). It therefore appears that the five-parameter version of the model is analytically necessary to provide a sufficient account. If algorithm speed-up occurs in a task, as it no doubt does in some cases (e.g., Rickard, 1997), then a seven-parameter version of the model may, in principle, be needed (treating algorithm speed-up as a three-parameter power function). Again, however, a five-parameter version of this extended model that ignores both the algorithm and retrieval asymptote parameters would probably fit such data about as well in most cases.

General Discussion

Among the models in the literature, the CMPL model now provides the best available account of practice effects in tasks exhibiting a transition to retrieval-based performance, on each of three grounds: (a) It is apparently the only model that is able to handle the empirical effects of the algorithm difficulty manipulations in previous work, (b) it is the only model that predicts the item-level step-function speed-up patterns seen here, and (c) it is the only model that has successfully passed the group-level sufficiency test introduced in this article.

Implications for Global Empirical Learning Laws

Psychologists have long held an interest in the general empirical function that governs performance improvements with practice (Fitts & Posner, 1967; Guilliksen, 1934; Mazur & Hastie, 1978; Newell & Rosenbloom, 1981; Restle & Greeno, 1970; Thurstone, 1919; Welford, 1968). Newell and Rosenbloom performed a meta-analysis of average learning curves and argued that a three-parameter power function described them best across a variety of tasks. Their proposed power law of practice has been influential. A number of theorists have viewed it as important to, or even as a benchmark test of, a viable learning theory (Anderson, 1982, 1993; Anderson & Schooler, 1991; Cohen, Dunbar, & McClelland, 1990; Logan, 1988, 1992; Newell & Rosenbloom, 1981; Palmeri, 1997).

More recently, researchers have pointed to limitations of the power law. Following Rickard's (1997) and Delaney et al.'s (1998) demonstrations that the power law fails for averaged data when there is a strategy shift, Heathcote et al. (2000) showed that it does not optimally describe speed-up in the more general case at the item level. In a meta-analysis of data from multiple experiments, Heathcote et al. showed that the three-parameter exponential function generally fit somewhat better than did the power function. They also advanced a new four-parameter smooth function, a hybrid of the power and exponential functions, that they termed the APEX function. It fit slightly but consistently better than did any of the other functions that they tested. Heathcote et al. were the first to systematically explore the smooth learning curves at the item level, a fact that explains their divergent findings.

The step-function speed-up results in this study appear to falsify the exponential, power, APEX, or indeed any other smooth speed-up function as a universal empirical learning law.[15] Qualitative process shifts similar to those observed here appear to be quite common in skill learning, and the group-level deviations from the power law that are predicted by CMPL have now been demonstrated in multiple tasks. Hence, given appropriate empirical scrutiny, similar item-level step-function RT shifts may be observable for a substantial proportion of learning tasks.

It would be possible to elaborate on smooth speed-up functions by adding a step-function shift parameter. In fact, the basic candidate function is given in Equation 3. In my opinion, however, such an endeavor may have limited value. The time may have come to abandon the effort to find a single, universal function for describing practice effects and to instead focus on testing and expanding existing process models that yield theoretically motivated and empirically supported speed-up predictions.

However, this conclusion is directed only to the level of the entire learning curve for a task. The search for the empirical learning law may still be of value at the level of specific underlying processes or strategies (for more discussion of the process-specific approach to learning curves, see Rickard, 1997, and Delaney et al., 1998). It is reasonable to expect that a component cognitive process, such as memory retrieval, may follow the same speed-up function regardless of what other properties the overall task may have (e.g., regardless of whether a qualitative strategy shift occurs with practice). The end result of such work might be identification of a single smooth function, like the exponential or power function, that governs process-pure speed-up for all component cognitive processes. The question of exactly what constitutes a pure-process mental event in the more general case would need to be addressed, but that question could be instructive in itself and should not dissuade researchers from investigating learning curves from this perspective.

Prospects for the Strategy Race Assumption of the Instance Theory

Fits of the instance theory to date have assumed that an instance is encoded on every trial and that all instances are retrieved on every trial. Logan (1988), however, suggested that this simplifying assumption might not be realistic. As examples of elaborations to the model, one could assume that instances are encoded on some random proportion of trials or that instances are encoded more or less frequently as practice proceeds. Both elaborations would require that one or more new free parameters be added. Random probabilistic instance encoding might not impact the power-function of expected-value prediction for retrieval very much and certainly would not violate the smooth-function property expected-value speed-up. Gradual increasing or decreasing of the rate of

[15] Mathematically speaking, a smooth function that can exhibit a steep, logistic-type curve at the strategy shift point is not ruled out by the current results. However, for Type 2 cluster items, such a function would be impossible to differentiate from a step-function RT drop, because the entire logistic shift could occur over an interval of less than one practice trial. It is unclear, moreover, how such a function could be justified on psychological grounds. It would seem to require that learning is not a discrete, quantized event that happens at most once per trial per item but has a truly continuous property between trials, yielding a concave curve, an inflection point, and a convex curve. Furthermore, such a function would likely require two or more parameters to describe the strategy shift, whereas a true step-function model requires only one parameter to describe the shift.

instance accrual as a function of practice might yield a retrieval speed-up function other than a power function, but it would again yield a smooth expected-value curve for retrieval and thus for the speed-up curve overall. These or related elaborations would thus not be able to account for the step-function strategy transitions observed for most items in this study.

The idea of parallel strategy execution might be preserved, however, if one allows for the possibility that there is no (or only very weak) instance accrual up to Trial *n*. On that trial, the first instance is encoded. If that instance allows for very fast retrieval by itself, there will be an abrupt RT drop on trial $n + 1$, from which point retrieval could solely determine performance. In this way, step-function reductions in RT, even beyond the special case occurrence on the second practice trial, could be generated by a strategy race model.

This alternative cannot be eliminated by the current data. It may have limited appeal, however. This version of the theory abandons what in my view is one of the more elegant aspects of the original instance model, the gradual replacement of the algorithm by retrieval over practice trials. It preserves the strategy race in principle, while effectively eliminating any empirical consequence of it for the learning curve. It also increases the complexity of the model. To fit the data, a new free parameter must be added to specify the trial on which the first potent instance is encoded. Furthermore, extension of the instance model to this special case would make the broader model much more flexible and thus more difficult to test. In contrast, the CMPL model is constrained to predict item-level step-function RT drops. If such drops cannot be detected either visually or statistically, even when the algorithm is reasonably time consuming, then the CMPL model, and perhaps the entire class of strategy selection models, can be rejected for the task.

References

Anderson, J. R. (1982). Acquisition of cognitive skills. *Psychological Review, 89,* 369–406.

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

Anderson, J. R., & Schooler, L. J. (1991). Reflections on the environment of memory. *Psychological Science, 2,* 396–408.

Anderson, R. B., & Tweney, R. D. (1997). Artifactual power curves in forgetting. *Memory & Cognition, 25,* 724–730.

Ashcraft, M. H. (1981). Mental addition: A test of three verification models. *Memory & Cognition, 9,* 185–196.

Carrier, L. M., & Pashler, H. (1995). Attentional limited 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, 97,* 332–361.

Colonius, H. (1995). The instance theory of automaticity: Why the Weibull? *Psychological Review, 102,* 744–750.

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

Cousineau, D., Goodman, V. W., & Shiffrin, R. M. (2002). Extending statistics of extremes to distributions varying in position and scale and the implications for race models. *Journal of Mathematical Psychology, 46,* 431–454.

Crutcher, R. J., & Ericsson, K. A. (2000). The role of mediators in memory retrieval as a function of practice: Controlled mediation to direct access. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 26,* 1297–1317.

Delaney, P. F., Reder, L. M., Staszewski, J. J., & Ritter, F. E. (1998). The strategy-specific nature of improvement: The power law applies by strategy within task. *Psychological Science, 9,* 1–7.

Ericsson, K. A., & Simon, H. A. (1993). *Protocol analysis: Verbal reports as data.* Cambridge, MA: MIT Press.

Estes, W. K. (1956). The problem of inference from curves based on group data. *Psychological Bulletin, 53,* 134–140.

Fernandes, M. A., & Moscovitch, M. (2002). Factors mediating the effect of divided attention during retrieval of words. *Memory & Cognition, 30,* 731–744.

Fitts, P. M., & Posner, M. I. (1967). *Human performance.* Belmont, CA: Brooks/Cole.

Green, D. R., Cerella, J., & Hoyer, W. J. (2000). *Do strategy probes affect the probability of item retrieval?* Poster presented at the 41st annual meeting of the Psychonomics Society, New Orleans, LA.

Guilliksen, H. (1934). A relational equation of the learning curve based on Thorndike's law of effect. *Journal of General Psychology, 11,* 395–434.

Haider, H., & Frensch, P. A. (2002). Why aggregated learning follows the power function of practice when individual learning does not: Comments on Rickard (1997, 1999), Delaney et al. (1998), and Palmeri (1999). *Journal of Experimental Psychology: Learning, Memory, and Cognition, 28,* 392–406.

Heathcote, A., Brown, S., & Mewhort, D. J. K. (2000). The power law repealed: The case for an exponential law of practice. *Psychonomic Bulletin and Review, 7,* 185–207.

Jenkins, L., & Hoyer, W. J. (2000). Memory-based automaticity and aging: Acquisition, re-acquisition, and retention. *Psychology and Aging, 15,* 551–565.

Kling, J. W. (1971). Learning: An introductory survey. In J. C. Kling & L. A. Riggs (Eds.), *Woodworth and Scholsberg's experimental psychology* (pp. 551–613). New York: Holt, Rinehart & Winston.

Liang, T., & Healy, A. F. (2002). The unitization effect in reading Chinese and English text. *Scientific Studies of Reading, 6,* 167–197.

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

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. (1995). The Weibull distribution, the power law, and the instance theory of automaticity. *Psychological Review, 102,* 751–756.

Logan, G. D., & Delheimer, J. A. (2001). Parallel memory retrieval in dual task situations: II. Episodic memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 27,* 1072–1090.

Logan, G. D., & Schulkind, M. D. (2000). Parallel memory retrieval in dual task situations: I. Semantic memory. *Journal of Experimental Psychology: Human Perception and Performance, 26,* 1072–1090.

Mazur, J. E., & Hastie, R. (1978). Learning as accumulation: A reexamination of the learning curve. *Psychological Bulletin, 85,* 1256–1284.

Myung, I. J., Kim, C., & Pitt, M. A. (2000). Toward an explanation of the power law artifact: Insights from a response surface analysis. *Memory & Cognition, 28,* 832–840.

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

Nino, R., & Rickard, T. C. (2003). Practice effects on two retrievals from a single cue. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 29,* 373–388.

Nosofsky, R. M., & Palmeri, T. J. (1997). An exemplar-based random walk model of speeded classification. *Psychological Review, 104,* 266–300.

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

Palmeri, T. J. (1999). Theories of automaticity and the power law of practice. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 25,* 543–551.

Reder, L. M., & Ritter, F. E. (1992). What determines initial feeling of knowing? *Journal of Experimental Psychology: Learning, Memory, and Cognition, 18,* 435–451.

Restle, F., & Greeno, J. (1970). *Introduction to mathematical psychology.* Reading, MA: Addison-Wesley.

Richardson, J. T. E. (1988). The availability and effectiveness of reported mediators in associative learning: A historical review and an experimental investigation. *Psychonomic Bulletin & Review, 5,* 597–614.

Rickard, T. C. (1997). Bending the power law: A CMPL theory of strategy shifts and the automatization of cognitive skills. *Journal of Experimental Psychology: General, 126,* 288–311.

Rickard, T. C. (1999). A CMPL alternative account of practice effects in numerosity judgment tasks. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 25,* 532–542.

Rickard, T. C., & Bajic, D. (2003). Automatic mediation or absence of mediation? Commentary on Crutcher and Ericsson (2000). *Journal of Experimental Psychology: Learning, Memory, and Cognition, 29,* 1381–1386.

Rickard, T. C., & Bajic, D. (in press). Memory retrieval given two independent cues: Cue selection or parallel access? *Cognitive Psychology.*

Rickard, T. C., & Logan, G. D. (2003). *Memory retrieval practice: Strengthening or instance accrual?* Manuscript in preparation.

Rickard, T. C., & Pashler, H. (2003). *A bottleneck in memory retrieval from a single cue.* Manuscript submitted for publication.

Rogers, W. A., Hertzog, C., & Fisk, A. D. (2000). Age-related differences in associative learning: An individual differences analysis of ability and strategy influences. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 26,* 359–394.

Rohrer, D., Pashler, H., & Etchegaray, J. (1998). When two memories can and cannot be retrieved concurrently. *Memory & Cognition, 26,* 731–739.

Ross, H. B., & Anderson, J. R. (1981). A test of parallel versus serial processing applied to memory retrieval. *Journal of Mathematical Psychology, 24,* 182–233.

Schneider, W. (1995). Micro Experimental Laboratory (MEL; Version 2.0) [Computer software]. Pittsburgh, PA: Psychological Software Tools.

Schunn, C. D., Reder, N. L., Nhouynanisvong, A., Richards, D. R., & Stroffolino, P. J. (1997). To calculate or not to calculate. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 23,* 1–27.

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

Siegler, R. S., & Shrager, J. (1984). A model of strategy choice. In C. Sophian (Ed.), *Origins of cognitive skills* (pp. 229–293). Hillsdale, NJ: Erlbaum.

Thurstone, L. L. (1919). The learning curve equation. *Psychological Monographs, 26,* 51.

Touron, D. R., Hoyer, W. J., & Cerella, J. (2001). Cognitive skill acquisition and transfer in younger and older adults. *Psychology and Aging, 16,* 555–563.

Welford, T. A. (1968). *Fundamentals of skill.* London: Methuen. (Original work published 1938)

Wenger, M. J. (1999). On the whats and hows of retrieval in the acquisition of a simple skill. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 25,* 1137–1160.

Appendix

Instance Model Fit Procedure

The instance model was fit separately to each item for each of the two cases. The first case is described in the main text and has an analytical solution. The second case, in which the mean of the parent-instance distribution is greater than mean of the algorithm, is complicated by the fact that there is no known equation specifying the model. As such, the only way to generate the model's predictions for a given set of parameter values is to run Monte Carlo simulations in which the expected value of the RT on each trial is estimated by taking the mean of the many simulated observations (100,000 per trial in this case). The resulting predicted speed-up curve for each set of parameter values is smooth by visual inspection of a graph (and sufficiently smooth for accurate estimation of r^2) but is not smooth at a very fine-grained level. This fact makes optimization using standard gradient descent algorithms difficult (because the numerically estimated derivatives are also not smooth). As an alternative approach, I performed a grid search over the parameter space for each item. With 206 items being fit, a sufficiently fine-grained grid search simultaneously over all five parameters was computationally prohibitive. I thus attempted to simplify the process by (a) finding good starting values for the parameters and (b) using reasonable heuristics to guide the search.

I began by noting that, if the RTs are assumed for the moment to be deterministic for both the algorithm and retrieval strategies, the Case 2 instance model would predict a single-node spline speed-up function. This function consists of a horizontal line for *n* blocks (reflecting the constant algorithm mean), followed by a three-parameter power function in which the initial RT prediction ($a + b$) is constrained to be the same as the algorithm mean. This function has four parameters, μ_{alg}, *a, b,* and *c* of the retrieval power function. This curve exactly describes the expected value prediction for the full, stochastic race model (in Case 2) up to the first trial on which an instance wins the race and beyond the last trial on which the algorithm wins the race. Thus, this function may provide reasonable parameter starting values for fitting the full stochastic model.

As the first step in model fitting, I found the best fitting spline function for each item. The mean r^2 for these fits was .5199. This fitting process was very similar to that used to fit the CMPL model, with the exception that the two components in the spline function were constrained to take the same value at the strategy shift point. Next, and before proceeding with further model fitting to actual data, I investigated the parametric properties of the spline function when it is fit to an instance theory function of known parameter values. I generated an expected-value instance function (to 50 trials) through Monte Carlo simulation (100,000 simulated datum per trial) for each of 20 different sets of the instance function's five parameters, spanning the expected range of the optimized parameter values for the final instance fits to the item data (these estimated ranges proved to be accurate in subsequent comparisons with actual parameter values of the final instance fits). I then fit the spline function to each of these 20 instance curves and compared the parameter values for μ_{alg}, *a, b,* and *c* as generated from

(*Appendix continues*)

the spline fits to the actual parameters that were used to generate the 20 instance curves. Parameter estimates that were based on the spline fits were often, though not always, in the same range as those used to generate the instance curve. Of particular note, the deviations of the parameter estimates that were based on the spline fits from the true-instance parameters were always in the same direction for each parameter. Specifically, for all 20 simulated items, estimates for μ_{alg}, *b*, and *c* were always too low, and the estimate for *a* was always too high. This finding allowed the grid search to proceed only in one direction for each parameter and item in the next step of data fitting.

Following the spline function fit, a series of partial grid searches were performed over the five instance-function parameters to iterate toward the optimal solution for each item. The starting parameter estimates for the first iteration were based on the results of the spline fits. Each step is listed below:

1. Multiple grid points were searched over μ_{alg}, *b*, and *c*, holding σ_{alg} and *a* at constant values. The value of *a* was held at the value achieved in the spine function fit. The value of σ_{alg} was set to $0.2 \times \mu_{\text{alg}}$, a value that is consistent with standard deviations relative to means in previous research. Relative to the parameter estimates at the end of the spline fit phase, the seven grid points that were searched on μ_{alg} were 0 through 900 ms, in equal increments of 150 ms. The search on *c* took values of one, two, three, four, and five times the value of *c* obtained in the spline fit. The search on *b* involved 11 grid points: $b \times 10^0$, $b \times 10^{.5}$, $b \times 10^1$, $b \times 10^{1.5}$, $b \times 10^2$, $b \times 10^{2.5}$, $b \times 10^3$, $b \times 10^{3.5}$, $b \times 10^4$, $b \times 10^{4.5}$, and $b \times 10^5$, where *b* corresponds to the value obtained in the spline fit. Preliminary analyses had shown that the optimal value of *b* occurred over a very large range of values over items. Generally, optimal fits having large values of *b* also had large values for *c*, and vice versa. Searching of the entire grid space defined by these three parameters (encompassing 385 grid points) for all of the 206 items required approximately 140 continuous hours of CPU time on a Pentium IV 1.6 Ghz computer. This generated a new mean r^2 value of .5303.

2. A search was performed solely on the parameter σ_{alg}, allowing that parameter to take values that were the following multiples of μ_{alg} achieved in the previous search step (as determined separately for each item): 0.05, 0.10, 0.15, 0.20, 0.25 and 0.30. Thus, σ_{alg} for each item in this search took a minimum of $0.05 \times \mu_{\text{alg}}$ and a maximum of $0.30 \times \mu_{\text{alg}}$. The new mean r^2 value, .5309, was hardly improved relative to the result of the previous search.

3. A search was performed on the parameter *a* alone, allowing it to take all values from 350 and 600 ms in increments of 50 ms. The mean r^2 increased to .5377.

4. A search was performed again over μ_{alg}, *b*, and *c*. In this case, the parameters took values, relative to their best fitting values in the earlier fit, −5%, 0%, and 5%. The new r^2 was .5402.

5. Another search was performed on the parameter *a*, this time in increments of 25 ms from 350 ms to 600 ms. The mean r^2 increased to .5458.

6. Next, μ_{alg}, *b*, and *c* was searched again. The parameters took values, relative to their best fitting values in the earlier fit, −5%, −2.5%, 0%, 2.5%, and 5%. The new r^2 was .5495.

7. The parameter μ_{alg}, which appeared to be the main parameter needing optimization on the basis of degree of change and corresponding r^2 improvement in the earlier fits, was searched again. The search included all values from −10% to 10%, in increments of 1%, relative to the value of μ_{alg} that was obtained in the preceding fit. The new mean r^2 was .5501.

8. The parameter *a* was again optimized, this time in increments of 10 ms, from 350 ms to 600 ms. The mean r^2 improved to only .5507. No further search on this parameter was performed.

9. A final search was performed on σ_{alg}, allowing that parameter to take values that ranged from 5% to 30%, in increments of 1%, of the value of μ_{alg} obtained in the preceding search step. The new mean r^2 value improved to only .5516. No further optimization on this parameter was performed.

10. Step 6 was repeated twice. The new mean r^2 was .5522.

11. At this point, only seven items continued to show consistent r^2 improvement from fit to fit. These items were searched again on μ_{alg}, *b*, and *c*, from −10% to 10% of their previous values, in increments of 2.5%. Two iterations of this search were conducted. The final mean r^2 value, .5520, was not improved.

The initial spline fit captured most of the variance. Subsequent fit improvements were incremental and of diminishing magnitude. To further determine whether the fits were optimized, the differences scores for each item were computed between the r^2 of the final fit and the r^2 of the immediately previous fit for each item. The mean of these r^2 difference scores was −.00019, with a standard deviation of less than .003. A frequency polygon of the distribution indicated that there was no skew. The lack of right skew constitutes important evidence for optimum fits. There are two sources of variability in r^2 from fit to fit. One is simply a chance fluctuation in the fine-grained shape of the optimal instance curve from simulation to simulation, without any change in the parameter estimates. If optimal parameters have been found, this should be the only source of variability in the distribution of difference scores. This distribution should be symmetric and centered at zero. Note that the effect of this statistical fluctuation in fit could result in either a slight underestimation of the true r^2 or in a slight overestimation of the true r^2 for a given item. A second component could be the result of better fitting grid points being found for some items, leading to a systematic increase in r^2 for those items. Convolving of this component with the former would result in a right-skewed distribution overall. Hence, the lack of skew combined with the lack of increase in mean r^2 on the last iteration (i.e., an average difference score of near zero) provide solid evidence that a close approximation to the optimal fit to each item was found. For 181 of the 206 items, the r^2 achieved from this Case 2 instance model fit was higher than that achieved from the Case 1 fit.

Received December 24, 2002
Revision received June 17, 2003
Accepted July 7, 2003 ■