Journal of Experimental Psychology:
Learning, Memory, and Cognition
2009, Vol. 35, No. 1, 113–121

0278-7393/09/$12.00 DOI: 10.1037/a0013647

The Temporal Dynamics of Strategy Execution in Cognitive Skill Learning

Daniel Bajic and Timothy C. Rickard
University of California, San Diego

The transition from algorithmic to memory-based performance is a core component of cognitive skill learning. There has been debate about the temporal dynamics of strategy execution, with some models assuming a race (i.e., independent, capacity unconstrained parallel processing) between algorithm and retrieval, and others assuming a choice mechanism. The authors investigated this issue using a new approach that allows the latency of each algorithm step to be measured, in turn providing new insight into (a) whether there is slowing of 1 or more algorithm steps on trials immediately preceding the 1st retrieval trial for an item, as might be expected if there is a competitive strategy execution process of some type other than a race, and (b) whether there is partial algorithm completion on retrieval trials, as would be expected if the 2 strategies are executed in parallel. Results are uniquely consistent with a strategy choice mechanism involving a competition between the retrieval strategy and the 1st step of the algorithm.

Keywords: skill, learning, cued recall, strategy execution, strategy choice

An important component of cognitive skill learning—indeed, arguably the signature learning event for many tasks—is the shift from initial use of a slow, multistep *algorithm* to a faster and subjectively less effortful memory look-up of the answer (*direct retrieval*). The classic example is arithmetic learning. In doing single-digit multiplication, children may initially perform a repeated addition algorithm, but with sufficient practice they will transition to direct retrieval (e.g., Siegler, 1988). Multiple laboratory studies have confirmed the ubiquity of this shift for arithmetic and arithmetic-like tasks (Delaney, Reder, Staszewski, & Ritter, 1998; Jenkins & Hoyer, 2000; Logan, 1988, 1992; Onyper, Hoyer, & Cerella, 2006; Palmeri, 1997; Reder & Ritter, 1992; Rickard, 1997, 1999, 2004; Rogers, Hertzog, & Fisk, 2000; Schunn, Reder, Nhouyvanisvong, Richards, & Stroffolino, 1997; Touron, Hoyer, & Cerella, 2001, 2004).

Similar shifts from algorithmic (defined broadly) to retrieval-based performance are believed to occur in a wide variety of nonarithmetic domains, including recall from episodic memory (Rickard & Bajic, 2006), the shift from mnemonically mediated to unmediated memory retrieval (Kole & Healy, 2007; Rickard & Bajic, 2003), lexical decision (Logan, 1988), word reading (e.g., Tao & Healy, 2002), and text comprehension (Rawson, 2004). Similar shifts may occur under item repetition conditions for visuospatial tasks such as mental rotation (Kail, 1986). A reasonable argument can be made, in fact, that any efficiently executed cued recall is a consequence of this shift.

Recent efforts to characterize the mechanism underlying this shift make diametrically opposing claims about the dynamics of strategy execution on each trial. One class of models, exemplified by the instance theory of automaticity (Logan, 1988) and its theoretically allied successor, the exemplar-based random walk (EBRW) model (Nosofsky & Palmeri, 1997; Palmeri, 1997), assumes a straightforward race between the two strategies: Both strategies are attempted on each trial, and the finishing time for each strategy is unaffected by competition with the other strategy. The direct retrieval process that is initiated at the start of the trial is assumed to continue throughout algorithm execution even for complex algorithms involving multiple steps.

In this class of models, instance representations are assumed to support direct retrieval. With each repetition of a given item, a new memory instance is encoded. These instances (each with its own distribution of finishing times) race with one another and with the algorithm during each trial. As more instances accrue with practice, there is an increased probability that retrieval of at least one of the instances will beat the algorithm. The EBRW model elaborates on the instance theory by assuming that the instance retrieval feeds into a random walk retrieval process that races with the algorithm.

A second class of models assumes that only one strategy can be executed at any given moment (Rickard, 1997, 2004; Schunn et al., 1997; Siegler, 1988; for a more general theoretical framework that is consistent with this assumption, see Byrne & Anderson, 2001). Siegler's (1988) distribution of associations model assumes that retrieval is always attempted first, with the algorithm serving as a back-up strategy. The Schunn et al. (1997) source activation confusion (SAC) model, which builds upon Reder's (1987, 1988) work on question answering, focuses on the feeling of knowing phenomenon as a mechanism of strategy choice and on factors that affect feeling of knowing. Rickard's (1997) component power laws (CMPL) model provides perhaps the most natural framework, for the current purposes, within which to draw predictive comparisons between choice and race models, and it is thus the focus of the following development. We consider the other choice models further in the Discussion section.

Daniel Bajic and Timothy C. Rickard, Department of Psychology, University of California, San Diego.

We would like to thank Elena Eiden, Jazmin Morales, Laura Muzzatti, Laura Osborne, and Wang Pan for their assistance with data collection.

Correspondence concerning this article should be addressed to Timothy C. Rickard, Department of Psychology, University of California, San Diego, 9500 Gilman Drive, La Jolla, CA 92093. E-mail: trickard@ucsd.edu

The CMPL model treats the algorithm (e.g., the repeated addition algorithm for multiplication) as a sequence of memory retrieval steps (henceforth referred to as algorithm steps) that tap the same memory retrieval system that is used to execute the direct retrieval strategy—a memory system that is in turn assumed to be the same as that used in explicit cued recall. The model assumes a memory retrieval bottleneck such that only one retrieval can be completed at a time (for supporting evidence in the case of cued recall, see Nino & Rickard, 2003). Thus, for any algorithm that involves one or more retrievals from long-term memory, the two strategies cannot be executed in parallel.

Strategy choice in the model is based on a brief parallel competition between two interpretations of the stimulus at the beginning of each trial (i.e., between the "problem-level" nodes in the Rickard, 1997, simulation model). 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. The stimulus interpretation that first reaches an activation threshold is selected and the memory retrieval(s) for the corresponding strategy are executed. The other strategy is aborted and undergoes no further processing on that trial. The retrieval strategy becomes gradually more competitive over trials (through a strengthening process rather than instance accrual) until it eventually wins the competition with the algorithm.

To a close approximation, the CMPL model instantiates a race to determine which stimulus interpretation will be selected for further processing.[1] That is, choice processing adds no latency component to the total time to finish the trials, regardless of the strategy that is selected. An alternative and perhaps more viable version of the model would incorporate a time-consuming choice process, such that the competing but unsuccessful strategy slows the selection time of the winning strategy in proportion to its competitiveness (i.e., its "strength" relative to the winning strategy). These two hypothetical cases constitute what are referred to as *zero latency* and *positive latency* choice processes, respectively.

Two additional strategy execution dynamics that have not been considered formally in simulation models to date also merit consideration. The first of these is a modified version of the EBRW or instance models in which there is capacity-limited parallel strategy execution rather than a race (for related work see Navon & Miller, 2002; Tombu & Jolicoeur, 2003). As noted earlier for the race models, the retrieval process that is initiated at the start of the trial would be assumed to continue throughout algorithm execution. This model would predict that processing of one or both strategies will be slowed when the two strategies are in competition.

The second possibility, consistent with both parallel and choice models, is that subjects might execute the algorithm as a check on retrieval accuracy on one or more trials before gaining sufficient confidence to rely exclusively on retrieval for that item. In the case of choice models, strategy execution on such trials would be sequential—retrieval followed by the algorithm.

An Overview of the Empirical Evidence Bearing on the Models

As developed to date, race models assume that the retrieval strategy comes to dominate the algorithm gradually over multiple trials as the most quickly retrieved instance becomes probabilistically more likely to beat the algorithm. These models were originally designed to account for (among other things) smooth, power function response time (RT) speedup while making the novel predictions of power function reduction in the standard deviation (*SD*) and matched learning rate parameters for the RT and *SD* functions.

Rickard (1997, 1999, 2004) showed that, at least for some tasks, those predictions do not hold. Instead, patterns are consistent with what were novel predictions of the CMPL model: First, speedup in mean RT does not follow a power function; rather, there are separate power functions with different parameter values that govern speedup for the algorithm and retrieval trials.[2] Second, *SD* does not decrease as a power function; instead, separate power functions with different parameter values govern reductions in *SD* for the algorithm and retrieval trials. Third, due to the strategy mixture effect over items that is implied by the CMPL model, overall *SD* can, under some circumstances, reach its maximum value at roughly the point during training wherein about 50% of the trials involve memory retrieval; and fourth, RT speedup curves for individual items for each subject can exhibit an abrupt (step-function) RT reduction at the point of the strategy shift (Rickard, 2004; for related work see Haider & Frensch, 2002). The CMPL model predicts this step-function RT reduction provided that retrieval is executed more quickly than the algorithm at the point during training wherein the strategy shift occurs. (As a simplifying assumption, it has been assumed in CMPL model fits to date that once the shift to retrieval occurs for an item, the retrieval strategy is used for every subsequent trial. This assumption is not critical to the CMPL model, and it is not a requirement of choice models generally.) Step-function RT reduction is not expected in the averaged data because the strategy shift is not expected to occur on the same trial for every item.

The first three results outlined above might be explainable by the race models through modification of assumptions about how parameters differ over items (Palmeri, 1999). Rickard (2004), however, argued that the fourth result above is not consistent with any race model that assumes a gradual and probabilistic replacement of algorithm by retrieval over many trials, because gradual strategy replacement predicts a smooth, continuous speedup in expected RT value even at the item level.

Race models are nevertheless still viable, for two reasons. First, step-function RT drops have to date been demonstrated for only one task (alphabet arithmetic; Rickard, 2004). Second, even the step-function RT drops can be explained by a race model if the gradual strategy replacement assumption in those models is dropped. Suppose that for the first $n - 1$ trials for a given item, the algorithm is

[1] In the Rickard (1997) simulation, selection of the algorithm stimulus interpretation is slowed by an inhibitory connection from the retrieval path as the retrieval strategy becomes more competitive. Due to the properties of the activation functions in that model, however, completion time for the algorithm first step is unaffected by the competition from the retrieval strategy and vice versa, constituting a race dynamic.

[2] The CMPL model also predicts power law speedup for each strategy at the item level. Race models make the same prediction for retrieval speedup. Heathcote, Brown, and Mewhort (2000) showed that the exponential function fits slightly better than the power function to item-level data, calling these predictions into question. However, the CMPL prediction of item-level power function speedup is not central to either the CMPL architecture or to choice models generally, and this issue is not pertinent to the focus of this article.

executed, and that no memory instance (or only a very weak memory instance) is encoded. On trial n, the algorithm is again executed and a strong memory instance that can support fast direct retrieval on subsequent trials is finally encoded. From trial $n + 1$ onward, direct retrieval will win the race with the slower algorithm, potentially resulting in a step-function RT drop at the strategy shift point.

Indexing the Latency of Each Algorithm Step

In the current experiment, we attempted to gain more theoretical leverage by using a task that allows indexing of not only the RT for each trial (defined as the latency between stimulus onset and vocal response execution) but also the latency for completion of each step of the algorithm. On each trial of the experiment, subjects saw a two-digit number and were instructed to count forward from that number, pressing the space bar in synchrony with each count, until the computer informed them to stop and to speak the number to which they had counted. For each stimulus number (e.g., 21) the same number of counts was always required (say, 11), and the same response was always to be spoken (e.g., 32). Each stimulus was presented multiple times over training blocks. If subjects remembered the answer at any point during a trial, they could end the trial prior to completing the algorithm by speaking that answer. Each keypress recorded the approximate latency of each counting step, and a microphone voice key recorded the RT.

This task design allows us to explore two previously unaddressed questions about the temporal dynamics of strategy execution: First, on the last few algorithm trials preceding the first correct retrieval trial for an item, is there evidence of progressively slower execution times for one or more algorithm steps, as might be expected if the retrieval strategy becomes more competitive over trials and if there is a latency-consuming strategy competition (i.e., limited capacity parallel processing or positive latency choice)? And second, on retrieval trials (i.e., trials on which the answer is spoken prior to completion of the algorithm), is there evidence that some fraction of the algorithm steps are completed, as would be expected if the two strategies race?

Method

Subjects

Forty-one University of California at San Diego undergraduate students participated for course credit.

Materials, Apparatus, and Procedures

Subjects were tested individually on IBM-compatible personal computers, with each subject seated approximately 50 cm from the computer screen and approximately 3 cm from a microphone. The computer keyboard was positioned directly behind the microphone, such that the subject could comfortably place one hand over the space bar; the experimenter was seated to the right of the subject, with access to the keyboard's number pad. The experiment was created with E-Prime software (Psychology Software Tools, Pittsburgh, PA) and the accompanying voice-key apparatus (Model 200A).

The experiment consisted of a warm-up phase and a training phase. Prior to each phase, instructions were presented on the screen and were also read aloud by the experimenter. Within each trial of each phase, a stimulus would be presented visually, an algorithmic solution (if used) would consist of silent counting accompanied by concurrent keypresses, and the final response provided by the subject would be generated vocally. The Appendix lists all visual stimulus and vocal response items used in the training phase. The warm-up phase utilized the same values, each raised by 10 (e.g., 30 and 44 in the warm-up phase vs. 20 and 34 in the training phase, etc.). In the description below, a *block* is defined as one randomly ordered presentation for each of the 10 possible stimulus-response items, with each item therefore having a mean repetition lag of 10 trials across blocks.

The warm-up phase consisted of a single block. At the start of each trial, the screen went blank for 500 ms, a fixation field (consisting of three plusses) was presented at the center of the screen for 500 ms, the screen again went blank for 500 ms, and then a two-digit number—the trial stimulus—was presented at the center of the screen. Subjects were instructed to count silently upward from the presented number, pressing the space bar once with each count, until the word *STOP* was presented on the screen. At that point, the subject was to speak his or her answer—the number that he or she had counted up to—into the microphone. The stimulus remained on the screen during the keypresses and was replaced by the word *STOP* after the number of keypresses for that trial was equal to the value of the correct response minus the numerical stimulus, thus ranging from a minimum of 9 keypresses to a maximum of 15.

After the subject provided a vocal response, the experimenter entered the subject's response and recorded whether the voice key tripped properly. If the subject was in error (as might occur if a rapidly counting subject overshot the final answer), the correct response was presented for 5 s. Otherwise, the word *Correct!* was presented for 800 ms. Immediately following feedback, the next trial began.

The training phase of the study was identical to the warm-up phase, with the following exceptions. Multiple blocks were presented, and subjects were informed that the same set of starting numbers (stimuli) would be presented repeatedly throughout the phase, with each starting number always having the same final number. Subjects were informed that they therefore had two methods that could be used to find the correct answer for each trial: (a) counting upward from the starting number, tapping the space bar once with each count until the word *STOP* appeared on the screen, and (b) remembering the answer associated with the starting number for that trial and speaking the answer into the microphone without doing all of the keypresses. To promote parallel strategy execution, if such parallelism is possible, the instructions stated (falsely) that "Many subjects report good results when they attempt to use both strategies at the same time." Subjects were told that they could speak the answer into the microphone at any time during each trial. They were instructed that they should try to finish this part of the experiment as quickly as possible while still being accurate.

In this phase, each trial stimulus was removed from the screen either when the subject spoke an answer or when the subject had entered a sufficient number of keypresses to bring the word *STOP* onto the screen—whichever came first. Subjects were permitted a brief pause between each block and continued to receive new blocks until 45 min from the start of this phase, after which the experiment concluded and the subject was debriefed.

Results

Prior to analysis, data from 3 subjects were discarded due to unusually low accuracy (<80%) and data from 1 subject were discarded due to unusually frequent voice-key errors. All analyses reported below are for the training phase data of the remaining 37 subjects.

Voice-key errors, which occurred in 6% of trials, were removed prior to analysis. Mean accuracy on the first training block was 90.5%, increasing to 95.6% by the 23rd block, the furthest block that all 37 subjects completed.

The mean of the subject-level mean correct RTs (latency from stimulus presentation to vocal response) is plotted as a function of training block in Figure 1A, and the mean of the subject-level *SD*s is plotted in Figure 1B. Best fitting three-parameter power functions are also included in the figures for reference. The observed pattern of deviation from power function improvement in both cases is consistent with that observed in prior studies of tasks that exhibit the shift from algorithm to retrieval (Rickard, 1997, 1999). Of particular note, the pattern of increasing *SD*s over the first few blocks has been observed previously for both numerosity judgment (Rickard, 1999) and alphabet arithmetic (Wagenmakers & Brown, 2007), the two tasks most studied to date.

The proportion of trials in which subjects retrieved the answer (defined as those trials in which the subject spoke the answer before completing all algorithm steps) is shown as a function of training block in Figure 2. The strategy shift was about 80% complete by the 23rd block, roughly in accord with findings of studies in which strategy probes were used (e.g., Rickard, 1997). Both Figure 2 and the peak of the *SD* curve in Figure 1B indicate that the shift to retrieval had occurred for 50% of the items by about Block 9.

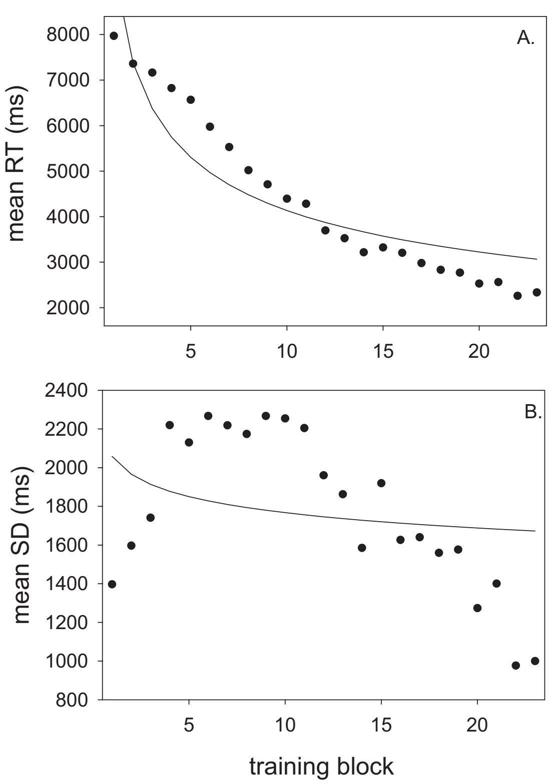

Figure 1. Mean response time (A) and mean standard deviation (B) as functions of training block for the first 23 blocks (to which all subjects contributed) along with the best fitting three-parameter power functions. RT = response time.

There were 57 items over 9 subjects (15% of items) that exhibited no shift to retrieval. Mean latencies for these items decreased from about 8,000 ms on the first block to about 5,000 ms on the 23rd block. This speedup was well fit by a power function—a result that is consistent with the CMPL model, according to which separate power functions govern speedup for each strategy. Further analysis showed stepwise speedup of about 200 ms for Algorithm Steps 2 and onward over the course of the first 13 training blocks, with no further speedup thereafter. In contrast, over the course of training there was several hundred milliseconds of slowing in latency to execute the algorithm first step. These results presage the results for shift items discussed below and are consistent with a positive latency choice competition between retrieval and the algorithm first step. For these no-shift items, it appears that the retrieval strategy did not become sufficiently competitive to win against the algorithm strategy before the end of training.

We also evaluated item-level RT plots, following the visual and statistical categorization scheme used by Rickard (2004). The results are shown in Table 1, along with results from the alphabet arithmetic task (Rickard, 2004) for comparison. About 6% of items exhibited no speedup, defined as a p value greater than .20 for the slope in a linear regression. Another 2.7% exhibited a step-function RT improvement (i.e., a visually prominent, abrupt, and sustained drop in RT) between the first and second block, indicating an immediate shift to retrieval (see 1st block items in Table 1), 41.9% exhibited step-function RT drops after Training Block 2 (Type 1 cluster items), 33.2% exhibited step-function RT drops after Block 2 with occasional slow outlier RTs late in training (Type 2 cluster items), and 16.2% exhibited smooth speedup (i.e., no pronounced step-function RT decrease).

Among the 22 items exhibiting no speedup, 15 never exhibited a shift to retrieval (i.e., for those items the algorithm was run to completion on every trial). The remaining 7 items showed various unusual patterns that masked speedup in the linear regression, most often a reversion back to use of the algorithm toward the end of training, yielding a U-shaped learning curve. Among the 60 items exhibiting smooth speedup, 40 never exhibited a shift to retrieval. The remaining 20 smooth speedup items (5.4% overall) are candidate cases of parallel strategy execution in which retrieval gradually becomes more competitive with the algorithm over multiple trials. It should be noted, however, following Rickard (2004), that obvious and usually dramatic deviations from smooth speedup were necessary before an item was classified as a Type 1 or Type 2 cluster item. For most smooth speedup items, there were hints of discontinuities similar to those for items classified as Type 1 or Type 2 cluster items.

Although the Type 1 and Type 2 cluster items rule out parallel strategy execution in which there is a gradual shift from algorithm to retrieval over trials for most items, they do not rule out the special case of parallel processing involving the abrupt increase in retrieval competitiveness that was described earlier. The finer grained analyses that are afforded by the current task design and that are discussed below provide a strong test of that version of the parallel model versus the choice model.

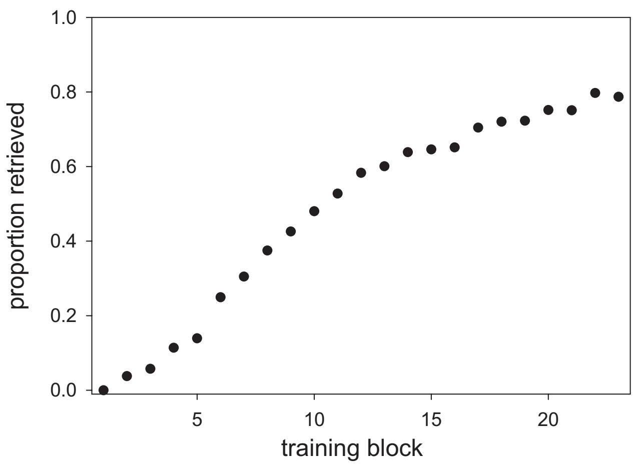

Figure 2. The proportion of trials on which the direct retrieval strategy was selected, as a function of training block.

Algorithm Step Latencies on Trials Preceding the First Correct Retrieval

Prior to conducting this analysis, we reset the training block variable for each item for each subject such that zero corresponded to the first correct retrieval block for that item, with blocks preceding the first correct retrieval taking negative values. For each subject, the mean latency (over items) for each algorithm step (correct trials only) was then computed for block values of -5 through -1 (i.e., for the last five algorithm blocks preceding each item's first correct retrieval block). These block means were then averaged over subjects and plotted in Figure 3A. Shown are results for the Algorithm Steps 1, 2, 3, and 4, along with the mean of Steps 5–9, among which there were no differences. Most items required more than nine algorithm steps, but data from those steps showed patterns like those for Steps 5–9 and so are not plotted.

The algorithm first step is substantially slower than subsequent steps, presumably reflecting subjects' need to orient to the presented stimulus and to initiate the counting algorithm. Also, for the algorithm first step, there was a pronounced (and not previously noted in the literature) 839 ms increase in latency from Block -5 through Block -1, confirmed by a within-subjects analysis of variance (ANOVA), $F(4, 130) = 10.51$, $p < .0001$. This algorithm

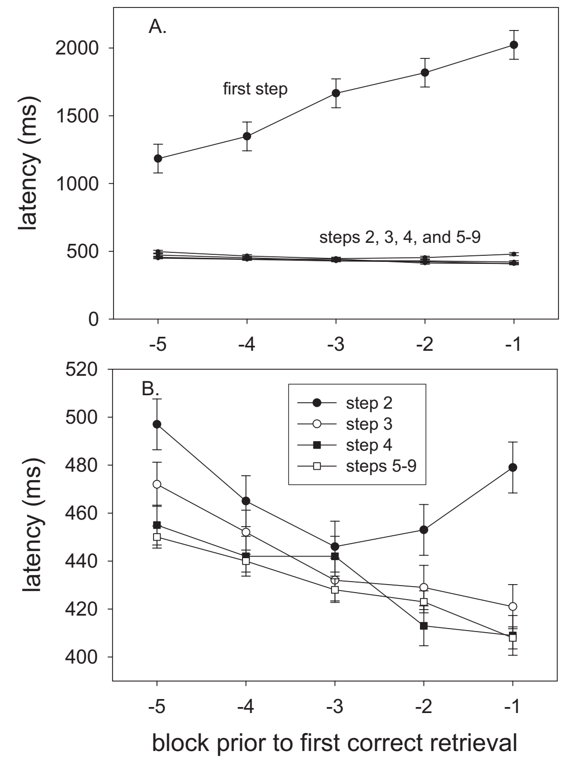

Figure 3. A: Mean algorithm step latencies on the five blocks prior to the first correct retrieval block for Algorithm Steps 1, 2, 3, and 4 and the mean of 5–9. B: Zoom of mean algorithm step latencies on the five blocks prior to the first correct retrieval block for Algorithm Steps 2, 3, and 4 and the mean of 5–9. Error bars represent standard errors of the mean.

first step slowing (which we term the *pause effect*) was not observed for Algorithm Steps 2–9, which are shown on a zoomed scale in Figure 3B. Instead, ANOVAs (identical to that described above) that were performed separately for each step indicated significant speedup over training blocks for Steps 3, 4, and the mean of Steps 5–9 ($p < .003$ in all cases). The U-shaped pattern for Step 2 did not reach significance ($p > .05$). In post hoc analyses, with the removal of the slower 2% of the data as outliers, the right section of the U-shaped curve for Step 2 was eliminated, yielding speedup analogous to that in Steps 3–9. The same outlier removal did not affect the shape of the functions in Figure 3 for any of the other algorithm steps. The speedup for Steps 2–9 appears to reflect algorithm learning over the course of the training blocks and is consistent with the algorithm speedup that was observed for the no-shift items. It is also possible that the speedup for Steps 2–9 on blocks approaching the first correct retrieval reflects attempts by subjects to make up for the time lost due to the Step 1 pause effect.

To explore whether the pattern described above extended beyond 5 blocks, we performed a supplementary analysis in which we included the first 15 blocks preceding the first correct retrieval block. Because many subjects completed the shift to retrieval in fewer than 15 blocks for many or all items, the number of items qualifying for this analysis was reduced by 75% relative to the 5-block case. Results are shown in Figures 4A and 4B. The general pattern is similar to that in Figure 3, with the slowing again being significant for Step 1, $F(14, 207) = 6.31$, $p < .0001$; and the speedup being

Table 1
Visual and Statistical Categorization of the Item-Level Data

Learning curve category	Frequency	Percentage[a]	Percentage in Rickard (2004)	Items exhibiting shift to retrieval[b]	
				n	%
No speedup	22	5.95	9	7	32
1st block	10	2.70	2.4	10	100
Type 1	155	41.89	48.3	154	99
Type 2	123	33.24	20.8	122	99
Smooth	60	16.22	29.1	20	33

[a] Out of all items. [b] Out of all items in each category.

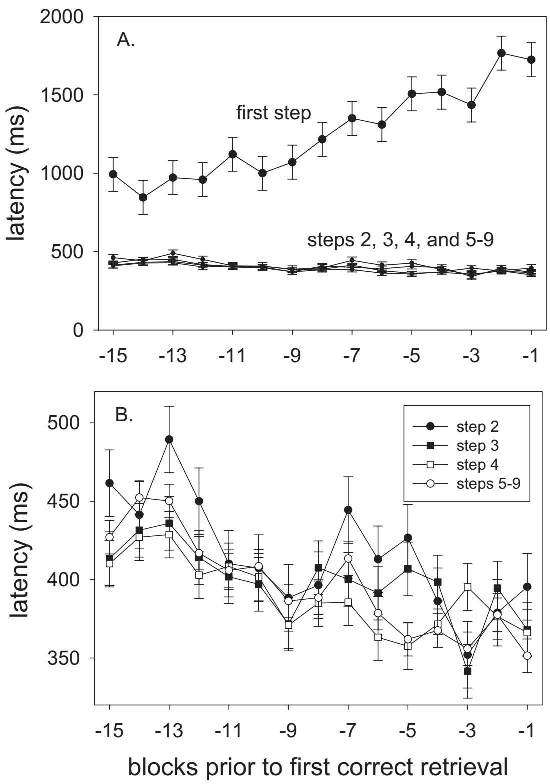

Figure 4. A: Mean algorithm step latencies on the 15 blocks prior to the first correct retrieval block for Algorithm Steps 1, 2, 3, and 4 and the mean of 5–9. B: Zoom of mean algorithm step latencies on the 15 blocks prior to the first correct retrieval block for Algorithm Steps 2, 3, and 4 and the mean of 5–9. Error bars represent standard errors of the mean.

significant for Steps 2, 4, and the mean of Steps 5–9 ($p < .05$ in all cases). In these averaged data, the algorithm first step slowing began about nine blocks prior to the first correct retrieval trial.

RTs (latency between stimulus onset and the vocal response) for items in the analyses described above (not shown in the figures) exhibited slowing from Block −5 to Block −1 that was analogous to, though of smaller magnitude than, that observed for Step 1 of the algorithm. That slowing was not statistically significant, however, because the Step 1 slowing was partially compensated for by the speedup that occurred over practice blocks for subsequent steps. Given this result, researchers should exercise caution when interpreting nonsignificant changes in overall algorithm RT over the course of training (e.g., Rickard, 2004). Note that algorithm first step slowing may not have been detected to date for other tasks because the first step latency has not been measured independently before, and because block numbers have not previously been synchronized to the first correct retrieval before averaging the data.

Accuracy on blocks approaching the first correct retrieval block is shown for the last 5 blocks in Figure 5A and for the last 15 blocks in Figure 5B. In both cases linear regression indicated significant negative slopes ($p < .05$ for the 5-block case and $p < .0001$ for the 15-block case). Further analyses showed that algorithm accuracy was constant in both cases at around .92. The increasing error rates were therefore entirely driven by increases in the rate of incorrect retrieval attempts.

It is worth noting that for 40% of the shift items, the shift occurred after the 13th training block, beyond which no algorithm speedup was observed for the no-shift items. Thus, we can reasonably infer that about 40% of the strategy shifts were occurring under conditions of constant, roughly asymptotic algorithm execution. Shift patterns for these 40% of items were qualitatively the same as for the other shift items. It appears, then, that the algorithm speedup with training for Steps 2 and onward was not a factor in the observed shift dynamics. Note also that none of these results depended on the number of algorithm steps, which varied from 9 to 15 over items.

Partial Algorithm Step Completion on Retrieval Trials

A bar graph of the frequency with which 0, 1, 2, 3, and so forth algorithm steps were completed during Blocks 0 to 4 (as defined in the preceding section, where zero corresponds to the first correct retrieval block for each item) is shown in Figure 6, along with the expected frequencies according to a race model. (Error trials and trials on which subjects reverted back to use of the complete algorithm were excluded.) These trials were used because they tended to have the slowest retrieval latencies, and hence they would be expected to exhibit the most algorithm step completion according to a parallel model. Results did not depend critically on this choice.

We derived the expected number of algorithm steps according to a race model in the following way. First, latencies for each algorithm step during the first block of the training phase were averaged over items for each subject (correct trials only). Prior anal-

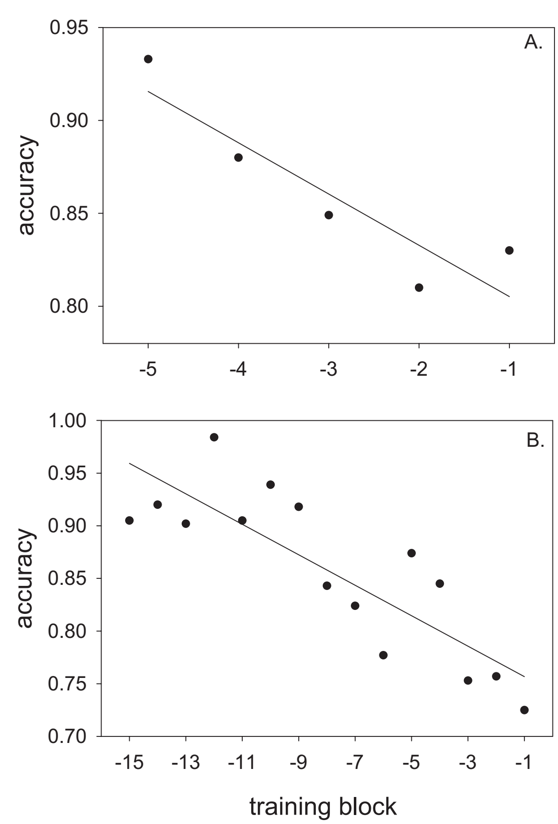

Figure 5. Accuracy on the last 5 (A) and last 15 (B) blocks prior to the first correct retrieval block.

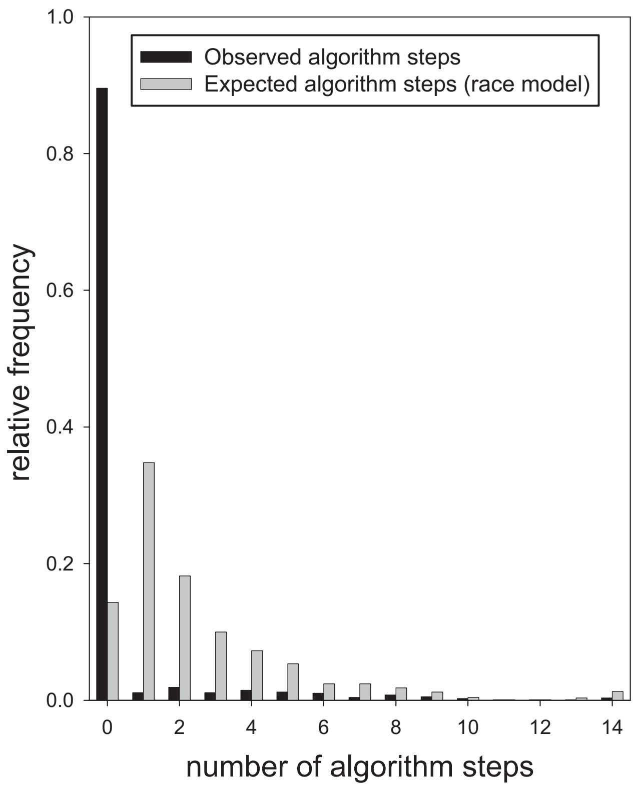

Figure 6. Relative frequency bar chart of expected (according to a race model) and observed completion frequencies for each algorithm step on the first five correct retrieval trials.

yses indicated no significant effect of items on these latencies, motivating the averaging. The first training block was used because no algorithm step slowing due to retrieval competition would be present. Next, for each retrieval trial under consideration, we estimated the expected number of completed algorithm steps under a race assumption by determining the number of first training block algorithm steps that the subjects would have been expected to complete on that trial—that is, by adding step latencies and recording the largest step number with a cumulative latency less than the retrieval latency for that trial. Note that because the algorithm exhibited speedup with practice, use of the latencies on the first training block in this calculation will tend to yield fewer predicted algorithm steps than would actually be expected by a race at the point of the strategy shift. Given the outcome described below, that bias does not complicate interpretation.

Figure 6 shows that partial algorithm step completion was far less frequent than expected by a race, $\chi^2(14) = 5{,}360$, $p < .0001$. Indeed, on 89.6% of those retrieval trials no algorithm steps at all were completed, a result that is in agreement with a strategy choice involving a competition between the retrieval strategy and the first step of the algorithm. For the remaining 121 trials, one or more algorithm steps were completed prior to answer retrieval (we refer to these as *partial algorithm* trials). These trials are candidates for parallel strategy execution.

A race model predicts that the speed of algorithm step execution on the partial algorithm trials will not be influenced by the race with retrieval. That prediction can be tested through analysis of algorithm step latencies on the partial algorithm trials shown in Figure 6 relative to the step latencies on the first training block, on which retrieval was not possible. For each subject, the means of the first, second, third, and fourth algorithm step latencies for the first training block were subtracted from the mean of these step latencies on partial algorithm trials. These difference scores were then subjected to matched t tests. (There were substantially fewer trials available for analysis of the later steps due to attrition. The trends, however, matched those of Steps 2–4.) Contrary to the race prediction, for Step 1 there was a highly significant 703.3 ms slowing on partial algorithm trials, $t(15) = 5.45$, $p < .0001$. For Steps 2–4, however, there were nonsignificantly faster completion times for the partial algorithm trials (difference scores of -93.8, -79.8, and -50.6, respectively; $p > .05$ in all cases).

Discussion

Implications for Skill Theories

None of the theories as developed to date can fully account for performance on the current task. The race theories cannot account for the results, even for the relatively infrequent partial algorithm trials, because of the substantial algorithm first step slowing that occurred even on those trials. Parallel models as a more general class, including limited capacity models, fare little better. Although a limited capacity model can explain the algorithm first step slowing on trials approaching the first retrieval trial (and on partial algorithm trials) when that result is considered in isolation, it cannot in any straightforward way explain why that slowing was not also observed on subsequent algorithm steps for those trials.

The results instead indicate a strategy choice process that involves a competition between the algorithm first step and the memory retrieval strategy. At that level of analysis the CMPL model fares well. However, the CMPL simulation model as developed to date (Rickard, 1997) assumes a zero-latency choice process, an assumption that is not, on its face at least, consistent with the data. Given the magnitude of the observed slowing (839 ms for Block -1 relative to Block -5 in Figure 3A), the simplest way to modify the CMPL model to account for the results would be to assume that the direct retrieval strategy wins the initial competition on some of those trials, and is executed, but that subjects are in some cases not sufficiently confident in the retrieved answer. They may then hold the retrieved answer in working memory and run the algorithm as a check. If the answers generated by the two strategies match, they may then tag that item as supporting correct retrieval and then rely solely on retrieval on subsequent trials if they recall the tag. This hypothesis may also explain the algorithm first step slowing that was observed on the partial algorithm trials. During algorithm execution on those trials, subjects may have decided to speak the previously retrieved answer prior to finishing the algorithm. This might occur, for example, because the act of counting narrows the range of candidate answers, potentially leading subjects to have more confidence in their initially retrieved answer. For both of these types of trials, this account characterizes the algorithm first step slowing as being the result of a *postchoice* strategy. The zero-latency choice process is thus, in principle, consistent with this account (i.e., the initial choice to retrieve might not be slowed by the competition with the algorithm). It should also be noted that the initial attempt at retrieval might simply fail to yield an answer. Within a framework such as the CMPL model, the retrieval strategy might win the competition, but the association to the answer may not be strong enough to bring answer activation above a response threshold. In this case, subjects would shift to the algorithm as a back-up strategy, in a manner analogous to that hypothesized in Siegler's (1988) distribution of associations model.

Alternatively, or in addition to the postchoice, sequential strategy execution hypothesized above, it is possible that the first step slowing reflects a *prechoice* (and presumably preconscious) competition that increases the time for the algorithm first step to be selected. This possibility corresponds, by our earlier definition, to a positive latency choice process, and it does not currently have an implementation in the CMPL model. Given the observed first step slowing of more than 800 ms (see Figure 3A), we speculate that the postchoice, sequential strategy account is most likely correct, at least as the major component of the slowing.

The Schunn et al. (1997) choice model has a number of features in common with CMPL. Both models assume a strategy choice that occurs prior to initiating either retrieval or the algorithm. Reder and colleagues (e.g., Schunn et al., 1997) have focused on feeling of knowing, which is modeled by the activation of stimulus representations within a semantic network, as a mechanism of strategy choice. The CMPL model embodies a similar network implementation of choice while also staking claims about the specific nature of the bottleneck that requires a strategy choice and about the manner in which the algorithm competes with retrieval. The two models appear to be compatible, and indeed they may offer prospects for synthesis. In the Rickard (1997) simulation model, activation of the problem-level node that corresponds to the retrieval strategy could serve as the basis for subjective feeling of knowing. The SAC model incorporates a mechanism for interference among items with overlapping operands that could be integrated with CMPL.

Generalization to Other Tasks

Clearly, strategy execution is not parallel in the current task, but to what range of tasks does that conclusion extend? Although more research is needed to address this question, a reasonably strong prediction can be made by considering the simplicity and low subjective cognitive load of our counting–tapping algorithm. We submit that similar results would be obtained for any of the broad class of algorithms that require a series of long-term memory retrieval steps. For example, given the current results it seems unlikely that a repeated addition algorithm for single-digit multiplication—which is much more subjectively taxing for children than is simple counting for adults—would run in parallel with retrieval.

Our task is atypical among those explored in the literature to date in that it required a simple keypress response in coordination with each algorithm step. It is not unique, however, in its requirement that a motor event take place in coordination with each algorithm step. The dot-counting task (e.g., Palmeri, 1997), which requires an orienting eye movement with each count, shares that property. For a number of reasons, it is unlikely in our view that simple algorithm-related motor events are the primary reason why retrieval and algorithm strategies were not executed in parallel in the current task. First, the counting component of the algorithm is a form of serial memory retrieval with (presumably) subvocal manifestation of each count, and it is on its face more likely to interfere with verbally based direct retrieval of the answer than is simple repetitive keypressing. Buttressing this claim is evidence that simple tapping alone has a negligible effect on other cognitive processing (for discussion see Pashler, 1994). Second, as noted above, the overall attentional demands of the algorithm used here are small compared to most other algorithms that have been explored in the literature to date or that occur in natural settings (e.g., arithmetic algorithms). Third, the item-level learning curve categorization for the current experiment is highly similar to that observed by Rickard (2004), who used an algorithm that involved no motor component (see Table 1).

We cannot strictly rule out the possibility that subjects used desynchronized algorithm tapping and counting; for example, tapping between counts. However, because desynchronizing is likely more time consuming than synchronizing, and because there was no task performance benefit to desynchronizing, it is reasonable to assume that subjects synchronized. If subjects did desynchronize, with counting preceding tapping, then it is possible that they executed retrieval in parallel through completion of the first count but aborted retrieval on the first tap. This possibility implies, however, that the tapping is the primary factor preventing parallel strategy execution, a possibility that appears unlikely in light of prior data on finger tapping that we noted above. Note also that neither the race models of skill nor any other current model of attention and performance that we know of would predict that simple, repetitive keypresses themselves would be sufficient to preclude parallel direct retrieval from long-term memory. The race models of skill as developed to date treat all algorithms homogeneously, and they assume that retrieval can take place in parallel throughout the execution of any algorithm.

It is an open question whether the same results will be obtained for the class of algorithms that do not require a sequence of long-term memory retrieval steps. Examples include practice on visual search and mental rotation with repeated presentation of the same items. More generally, any algorithm that involves only the execution of rules held in working memory is a member of this class. Additional work to explore strategy execution dynamics in such task domains is needed.

References

Byrne, M. D., & Anderson, J. R. (2001). Serial modules in parallel: The psychological refractory period and perfect time-sharing. *Psychological Review, 108*(4), 847–869.

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.

Haider, H., & Frensch, P. A. (2002). Why aggregated learning follows the power law 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). Instance-based automaticity and aging: Acquisition, reacquisition, and long-term retention. *Psychology and Aging, 15,* 551–565.

Kail, R. (1986). The impact of extended practice on rate of mental rotation. *Journal of Experimental Child Psychology, 42*(3), 378–391.

Kole, J. A., & Healy, A. F. (2007, November). *Examining the retrieval process in mediated learning using priming effects.* Poster presented at the 48th annual meeting of the Psychonomic Society, Long Beach, CA.

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*(5), 883–914.
Navon, D., & Miller, J. (2002). Queuing or sharing? A critical evaluation of the single-bottleneck notion. *Cognitive Psychology, 44,* 193–251.
Nino, R. S., & Rickard, T. C. (2003). Practice effects on two memory retrievals from a single cue. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 29*(3), 373–388.
Nosofsky, R. M., & Palmeri, T. J. (1997). An exemplar-based random walk model of speeded classification. *Psychological Review, 104,* 266–300.
Onyper, S. V., Hoyer, W. J., & Cerella, J. (2006). Determinants of retrieval solutions during cognitive skill training: Source confusions. *Memory & Cognition, 34,* 538–549.
Palmeri, T. J. (1997). Exemplar similarity and the development of automaticity. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 23*(2), 324–354.
Palmeri, T. J. (1999). Theories of automaticity and the power law of practice. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 25*(2), 543–551.
Pashler, H. (1994). Dual-task interference in simple tasks: Data and theory. *Psychological Bulletin, 116*(2), 220–244.
Rawson, K. A. (2004). Exploring automaticity in text processing: Syntactic ambiguity as a test case. *Cognitive Psychology, 49,* 333–369.
Reder, L. M. (1987). Strategy selection in question answering. *Cognitive Psychology, 19,* 90–138.
Reder, L. M. (1988). Strategic control of retrieval strategies. In G. H. Bower (Ed.), *The psychology of learning and motivation* (Vol. 22, pp. 227–259). San Diego, CA: Academic Press.
Reder, L. M., & Ritter, F. E. (1992). What determines initial feeling of knowing? Familiarity with question terms, not with the answer. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 18*(3), 435–451.
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*(2), 532–542.
Rickard, T. C. (2004). Strategy execution in cognitive skill learning: An item-level test of candidate models. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 30*(1), 65–82.
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*(6), 1381–1386.
Rickard, T. C., & Bajic, D. (2006). Cued recall from image and sentence memory: A shift from episodic to identical elements representation. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 32*(4), 734–748.
Rogers, W. A., Hertzog, C., & Fisk, A. D. (2000). An individual differences analysis of ability and strategy influences: Age-related differences in associative learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 26*(2), 359–394.
Schunn, C. D., Reder, L. M., Nhouyvanisvong, A., Richards, D. R., & Stroffolino, P. J. (1997). To calculate or not to calculate: A source activation confusion model of problem familiarity's role in strategy selection. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 23*(1), 3–29.
Siegler, R. S. (1988). Strategy choice procedures and the development of multiplication skills. *Journal of Experimental Psychology: General, 117,* 258–275.
Tao, L., & Healy, A. F. (2002). The unitization effect in reading Chinese and English text. *Scientific Studies of Reading, 6,* 167–197.
Tombu, M., & Jolicoeur, P. (2003). A central capacity sharing model of dual-task performance. *Journal of Experimental Psychology: Human Perception and Performance, 29,* 3–18.
Touron, D. R., Hoyer, W. J., & Cerella, J. (2001). Cognitive skill acquisition and transfer in younger and older adults. *Psychology & Aging, 16,* 555–563.
Touron, D. R., Hoyer, W. J., & Cerella, J. (2004). Cognitive skill learning: Age-related differences in strategy shifts and speed of component operations. *Psychology & Aging, 19,* 565–580.
Wagenmakers, E., & Brown, S. (2007). On the linear relation between the mean and the standard deviation of a response time distribution. *Psychological Review, 114,* 830–841.

Appendix

Phase 2 Stimulus–Response Pairings

Stimuli	Responses
20	34
21	36
22	32
23	35
24	33
25	37
26	41
27	40
28	39
29	38

Received March 27, 2008
Revision received July 24, 2008
Accepted July 25, 2008 ■