Mem Cogn (2011) 39:1147–1161
DOI 10.3758/s13421-011-0114-z

# Toward a generalized theory of the shift to retrieval in cognitive skill learning

Daniel Bajic · Timothy C. Rickard

Published online: 26 May 2011

**Abstract** Prior research on cognitive skill learning has shown that algorithmic and direct memory retrieval strategies are not executed in parallel if the algorithm entails a series of long-term memory (LTM) retrieval steps (as in the case, for example, of mental arithmetic). This phenomenon has been hypothesized to reflect a bottleneck in LTM retrieval processes that forces a strategy choice during an early stage of processing. Here, we investigate simple perceptual–motor algorithms that involve no memory retrieval steps, a largely unexplored case in which parallel strategy execution models remain viable. Pronounced strategy interference was again observed, albeit interference that was different in important respects from that observed for LTM retrieval algorithms. It appears that neither parallel nor choice models, as developed to date, are sufficient as a generalized theory of this skill learning phenomenon. Issues central to the development of a more comprehensive theory are discussed.

**Keywords** Skill · Learning · Algorithms · Strategy execution · Strategy choice

A signature phenomenon in cognitive skill learning is the shift from reliance on a multistep algorithm to direct memory retrieval. In doing single-digit multiplication, for example, children may initially perform a repeated addition algorithm, but with sufficient practice will transition to direct retrieval (e.g., Siegler, 1988). Multiple laboratory studies have confirmed this shift over a variety of arithmetic and nonarithmetic tasks (Delaney, Reder, Staszewski & Ritter, 1998; Hertzog, Touron & Hines, 2007; Jenkins & Hoyer, 2000; Logan, 1988, 1992; Onyper, Hoyer & Cerella, 2006; Palmeri, 1997; Rawson, 2004; Reder & Ritter, 1992; Rickard, 1997, 1999, 2004; Rickard & Bajic, 2003, 2006; Rogers, Hertzog & Fisk, 2000; Schunn, Reder, Nhouyvanisvong, Richards & Stroffolino, 1997; Touron, Hoyer & Cerella, 2001). An understanding of the temporal dynamics of strategy execution in these tasks is integral to the broader goal of modeling the underlying learning processes and performance mechanisms, as well as the factors that may modulate the rate of the strategy shift (e.g., Bourne, Raymond & Healy, 2010; Onyper et al., 2006; Salvucci & Taatgen, 2008; Touron & Hertzog, 2009).

One group of models assumes parallel strategy execution (Logan, 1988; Nosofsky & Palmeri, 1997; Palmeri, 1997, 1999). In these models as developed to date, the algorithm and retrieval strategies compete in a race, which is a special case of parallel processing involving independent and capacity-unconstrained strategy execution (for a general treatment, see Townsend & Nozawa, 1997). A second group of models assumes a strategy choice at an early stage of processing, such that only one strategy is executed at a time (Rickard, 1997, 2004; Schunn et al., 1997; Siegler, 1988). The most recent evidence (Bajic & Rickard, 2009) unambiguously favors the strategy choice account for the case of algorithms that involve long-term memory (LTM) retrieval steps. Bajic and Rickard used a task that allowed indexing of not only the response time (RT) for each trial (defined as the latency between stimulus onset and response execution) but also the latency of completion of each step of the algorithm. On each trial of their experiment, subjects were presented with a two-digit stimulus. The algorithm involved silent forward counting from the presented number, pressing the space bar in synchrony with each count, until the computer informed subjects to stop. They then spoke aloud the number to which they had counted.

D. Bajic · T. C. Rickard (✉)
Department of Psychology, University of California, San Diego,
La Jolla, CA 92093–0109, USA
e-mail: trickard@ucsd.edu

The latency of each keypress was recorded, providing the approximate latency of each counting step, and a microphone voice-key recorded the latency from the onset of the numerical stimulus to the vocal response (i.e., the RT). Each stimulus was presented multiple times over training blocks. For each stimulus (e.g., 21), the same number of counts was always required (e.g., 11), and the same response was always to be spoken (e.g., 32), leading to a shift to direct retrieval for most items—that is, the ability to recall a given stimulus's vocal answer without the use of the associated algorithm. Any trial in which an answer was vocalized prior to algorithm step completion could unambiguously be identified as a trial in which the direct-retrieval strategy had been used successfully as the basis for responding.

This task allowed Bajic and Rickard (2009) to explore two previously unaddressed questions about the temporal dynamics of strategy execution: (1) On the last few algorithm trials preceding the first correct retrieval trial for an item, is there evidence of progressively slower latencies for one or more algorithm steps, as would be expected if the retrieval strategy becomes more competitive over trials and if there is a latency-consuming strategy competition? And (2) on retrieval trials, is it generally the case that some fraction of the algorithm steps are completed, as would be expected if the two strategies can be executed in a race and if the latency of each algorithm step is much shorter than that for direct retrieval?

With respect to both questions above, the data supported a strategy choice account in which there is a competition between the algorithm first step (e.g., the first tap–count event) and the direct-retrieval strategy. Specifically, on the last few algorithm trials preceding the first correct retrieval trial for an item, there was a marked slowing in the execution of the algorithm first step (but not of subsequent steps), reaching a peak value of over 800 ms on the trial immediately preceding the first correct retrieval. Bajic and Rickard (2009) termed this selective first-step slowing the algorithm *pause effect*, and we will refer to the increasing magnitude of this effect on trials approaching the first correct retrieval trial as the *pause-effect slope*. The pause effect indicates that the algorithm was blocked temporarily while subjects attempted to retrieve the answer. Apparently, answer retrieval either failed on those trials, or subjects were insufficiently confident to execute the retrieved answer, and thus switched to the algorithm as a backup strategy. The pause effect, then, is consistent with strategy choice but not with parallel strategy execution.

Further, on the first five retrieval trials for each item, when retrieval was slowest, one or more algorithm steps were completed on only about 10% of trials. We will refer to this subset of retrieval trials as *partial-algorithm retrieval* trials. Based on the assumptions of a strategy race model (Logan, 1988; Palmeri, 1997; for a general treatment, see Townsend & Nozawa, 1997), it would have been predicted that one or more algorithm steps should have been observed on about 86% of those trials (calculation of this prediction is described later). Finally, on partial-algorithm retrieval trials, there was a roughly 700-ms *partial-algorithm pause effect* (i.e., a 700-ms delay in execution of the algorithm first step), mirroring the pause effect for algorithm trials. In combination, these results ruled out both the race model and any straightforward implementation of a limited-capacity parallel strategy execution model.

Bajic and Rickard (2009) interpreted their results as being consistent with an elaborated version of Rickard's (1997) component power laws (CMPL) model. That model, which is an example of the more general class of strategy choice models, assumes a bottleneck in cued recall from LTM (see Nino & Rickard, 2003; Rickard & Bajic, 2004). The entire retrieval event, up to the point at which the answer becomes available for responding, must be completed before a second retrieval can be initiated. Thus, for any algorithm that involves principally a series of LTM memory retrievals (even a series of retrievals as simple as counting), the algorithm and retrieval strategies cannot be executed in parallel. The relatively rare partial-algorithm retrieval trials in the Bajic and Rickard experiment can be accommodated by a choice model such as CMPL if it is assumed that, on occasion, subjects (1) retrieve the answer, yielding the partial-algorithm pause effect; (2) subsequently initiate algorithm execution anyway, perhaps due to insufficient confidence in the retrieved answer; yet (3) prior to completion of the full algorithm, choose to execute the previously retrieved answer. See Bajic and Rickard for a discussion of why subjects might interleave strategies in this way on occasional trials.

The CMPL model leaves open the possibility that strategy execution may run in parallel for algorithms that do not involve LTM retrieval operations. An example laboratory task is noun-pair lookup (Hertzog et al., 2007; Touron et al., 2001). In a typical version of that task, a set of noun pairs is presented at the top of the screen. On each trial, a noun-pair cue is presented in the center of the screen, and the subject must indicate, by pressing one of two keys, whether the cue pair corresponds to one of the pairs at the top of the screen. Subjects must either remember the paired nouns (direct retrieval) or visually search the noun-pair table (the algorithm) to find it. In this task, execution of the search algorithm may not require LTM retrieval at all, or at least may require much less than other algorithms (e.g., arithmetic).

There are also numerous retrieval-shift phenomena outside of the laboratory involving perceptual–motor

(P-M) "algorithms" having minimal or no LTM-retrieval component. As examples, consider remembering a keyboard shortcut to perform a word-processing operation versus searching a dropdown menu, directly remembering the location of a control lever in an unfamiliar automobile versus searching, and remembering a person's name or phone number versus looking it up on a contact list.

There are at least two reasons to suspect parallel strategy execution for the case of P-M algorithms. First, there is evidence from the dual-task literature for domain specificity in LTM retrieval interference. Fernandes and Moscovitch (2000, 2003) observed that retrieval performance is more adversely affected by a secondary task that involves the same domain (e.g., verbal retrieval and a verbal secondary task) than by a secondary task from a different domain (e.g., verbal retrieval and a numerical secondary task). Presumably, then, the direct (LTM) retrieval strategy would be less influenced by a P-M algorithm than by an LTM algorithm. Second, in many cases (e.g., the examples noted above), P-M algorithms are subjectively less taxing on general cognitive resources than are LTM retrieval algorithms (e.g., arithmetic algorithms). It may be that the critical factor is not whether the algorithm requires LTM retrieval per se, but whether it leaves sufficient resources unallocated (i.e., attentional capacity; Kahneman, 1973) to execute retrieval in parallel. In the P-M conditions of the experiments described below, the algorithms—involving repeated key tapping or mouse movements—are among the simplest possible from both subjective and theoretical perspectives. Strategy execution for these algorithms was compared to that for matched LTM retrieval algorithms—differing only in that they required that counting be performed concurrently with the P-M algorithm steps, as in Bajic and Rickard's (2009) tap–count task.

## Experiment 1

This experiment is nearly identical to that described in Bajic and Rickard (2009). In both that and the present experiment, the algorithm involved the repeated tapping of a key on a computer keyboard. However, whereas the algorithm described by Bajic and Rickard also required concurrent counting, the algorithm in the present experiment simply required the subject to begin tapping when a visual stimulus was presented and to continue tapping until the computer indicated the appropriate vocal response for that stimulus. In both Bajic and Rickard's study and the present experiment, the subject could terminate the algorithm early—or skip it altogether—if he or she directly retrieved the vocal response associated with a given stimulus.

### Method

*Subjects* A total of 31 University of California at San Diego undergraduate students participated for course credit.

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

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 two-digit number stimulus was presented visually, and the subject had to speak the answer (another two-digit number) either by use of the tapping algorithm or through memory retrieval. 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 versus "20" and "34" in the training phase). The Appendix also lists the number of algorithm steps, which was the number of key taps required to complete the algorithm for each item. Although the stimulus–response pairings were the same across all subjects, the number of required algorithm steps (key-presses) was randomly assigned to items for each subject, with a constraint such that the number of algorithm steps for a particular item could not be equal to the value of the correct response minus the numerical stimulus. In the description below, a *block* is defined as one randomly ordered presentation for each of the 10 stimulus–response items.

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 plus signs) 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 begin rapidly tapping the space-bar key when the number appeared. When the number of taps equaled the number of required algorithm steps for an item, the stimulus disappeared, and the word STOP was presented on the screen, along with the answer for that trial presented just below. The subject would then speak the answer into the microphone.

After the subject spoke the response, the experimenter entered it using the computer keyboard and recorded whether the voice key tripped synchronously with responding. Immediately after an incorrect response (rare in the warm-up phase), 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 response number. Subjects were informed that either or both of two methods could be used to find the answer for each trial: (1) tapping the space bar until the word STOP and the answer for that trial appeared on the screen (the algorithm), and (2) remembering the answer associated with the presented stimulus. To promote parallel strategy execution if it was possible, the instructions stated (falsely) that, "Many subjects have reported 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.

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 blocks and continued to receive new blocks until 45 min from the start of the training phase, after which time the experiment concluded and the subject was debriefed.

## Results

Only the training phase data were analyzed. Prior to analysis, subjects were excluded if their response error rate was greater than 20% or if the voice key failed to trip appropriately on more than 10% of trials.

For Experiment 1, all subjects had sufficiently high accuracy for inclusion, but 4 subjects were rejected due to their voice-key error rate. In all of the analyses that follow, trials were excluded if voice-key errors occurred (approximately 3% of trials in Exp. 1), if there was a latency of less than 180 ms from stimulus onset to the first algorithm step, or if less than 300 ms occurred from stimulus onset to the vocal response (less than 1% of trials across all experiments).[1]

Mean accuracy was initially near perfect (over 99% on the first training block), fell to its lowest value (approximately 91%) on Block 10, and then rose again to approximately 98% by the 27th block, the final block that all 27 subjects completed. The mean of the subject-level mean correct RTs (latency from stimulus presentation to vocal response, inclusive of both algorithm and retrieval trials) is plotted as a function of training block in Fig. 1a, and the mean of the subject-level standard deviations (*SD*s) is plotted in Fig. 1c. For reference, the corresponding data from the matched version of this experiment, which involved counting and tapping (Bajic & Rickard, 2009), are shown in Fig. 1b and d. Best-fitting three-parameter power functions for the RT data are also included for reference. Both the deviation from the power function in the mean RT and the inverted U-shaped *SD* values are consistent with results of prior studies (e.g., Rickard, 1997). These effects have to date only been observed for tasks that exhibit a shift to retrieval, and they reflect a strategy mixture over items and trials during the strategy-shift portion of training. The substantially faster algorithm latencies for the tap-only task (Fig. 1a) versus the tap–count task (Fig. 1b) is expected, because the relatively time-consuming counting operation is absent for the tap-only group.

The proportions of correctly answered trials on which the direct-retrieval strategy must have been the basis for responding—defined as those trials on which the subject spoke the answer before completing all algorithm steps—are shown as a function of training block in Fig. 1e. The strategy shift had occurred for 50% of items by about the 10th practice block, and was nearly 100% complete by the 27th block. Similar results were observed by Bajic and Rickard (2009); see Fig. 1f of the present article).

*Algorithm step latencies on trials preceding the first correct retrieval* Here and in subsequent analyses of algorithm step latency on trials approaching the first correct retrieval, the following procedure was used. First, the training block variable for each item for each subject was reset, such that zero corresponded to the first correct retrieval block for that item, with blocks preceding the first correct retrieval taking negative values. Below, when referring to block numbers synchronized in this manner, we will use the abbreviated term *sync-blocks*. For each subject, the mean latency for each algorithm step was then computed for sync-block values of −5 through −1 (i.e., for the last five algorithm blocks preceding each item's first correct retrieval block). Prior to calculating these means, error and voice-key error trials were removed (i.e., those trials were treated as missing data). Items with four or fewer blocks prior to the first correct retrieval were excluded from this analysis. These sync-block means were then averaged over subjects

[1] See Luce (1991) for discussion of plausible minimum response times in simple- and choice-RT tasks.

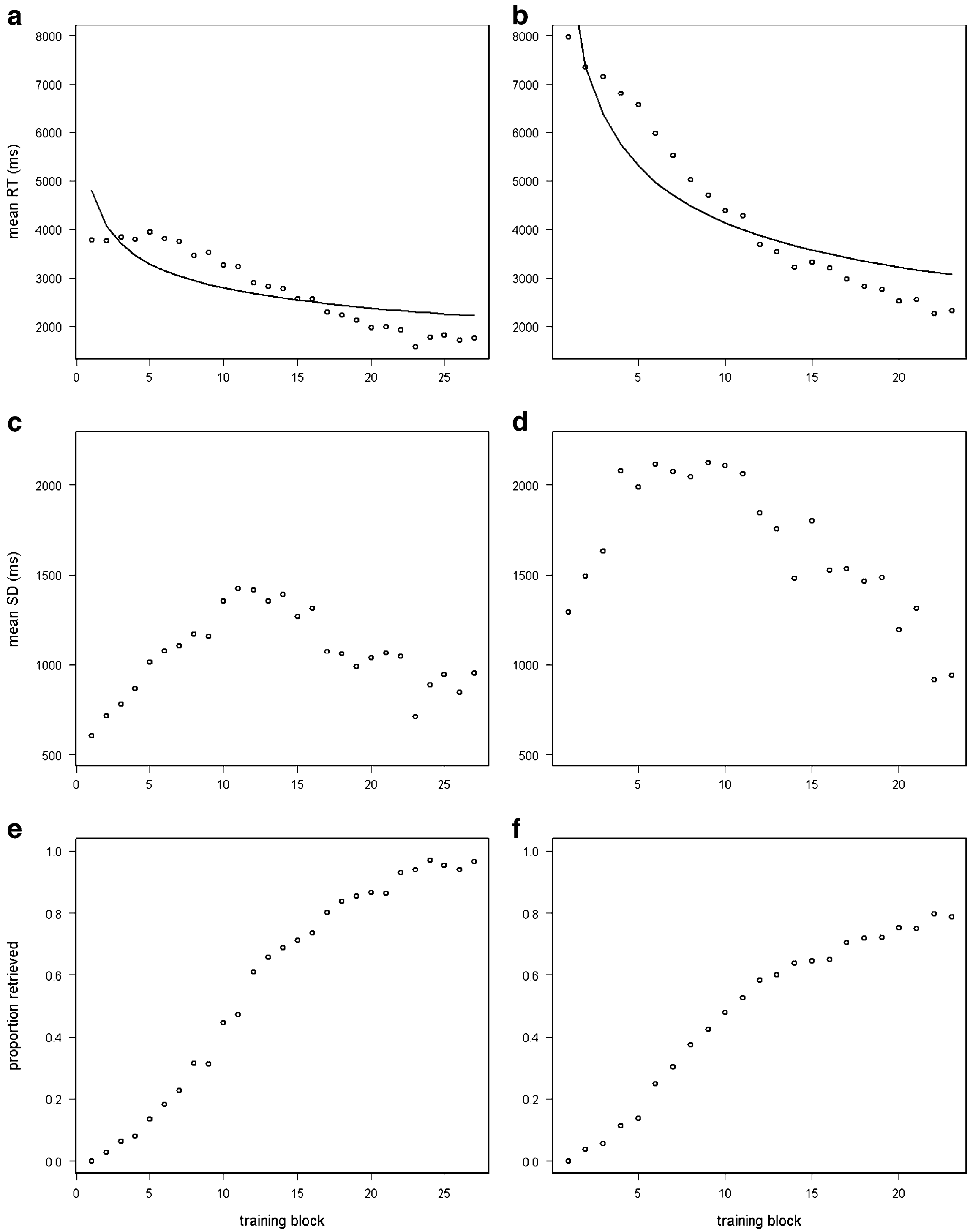

**Fig. 1** Mean response times (RTs) with the best-fitting three-parameter power functions (panels **a** and **b**), mean standard deviations (**c** and **d**), and proportions of trials on which the direct-retrieval strategy was selected (**e** and **f**), as functions of training block for Experiment 1 here (**a**, **c**, and **e**, respectively) and the experiment in Bajic and Rickard (2009; **b**, **d**, and **f**, respectively)

and plotted in Fig. 2a, which shows results for Algorithm Steps 1, 2, 3, and 4, along with the mean of Steps 5–9. (Most items required more than nine algorithm steps; data from those steps exhibited patterns like those for Steps 2–9.)

The algorithm first step was substantially slower than subsequent steps, presumably reflecting the need to orient to the presented stimulus and to initiate the tapping algorithm. Of greater interest, the algorithm first step exhibited a 340-ms increase in latency from Sync-

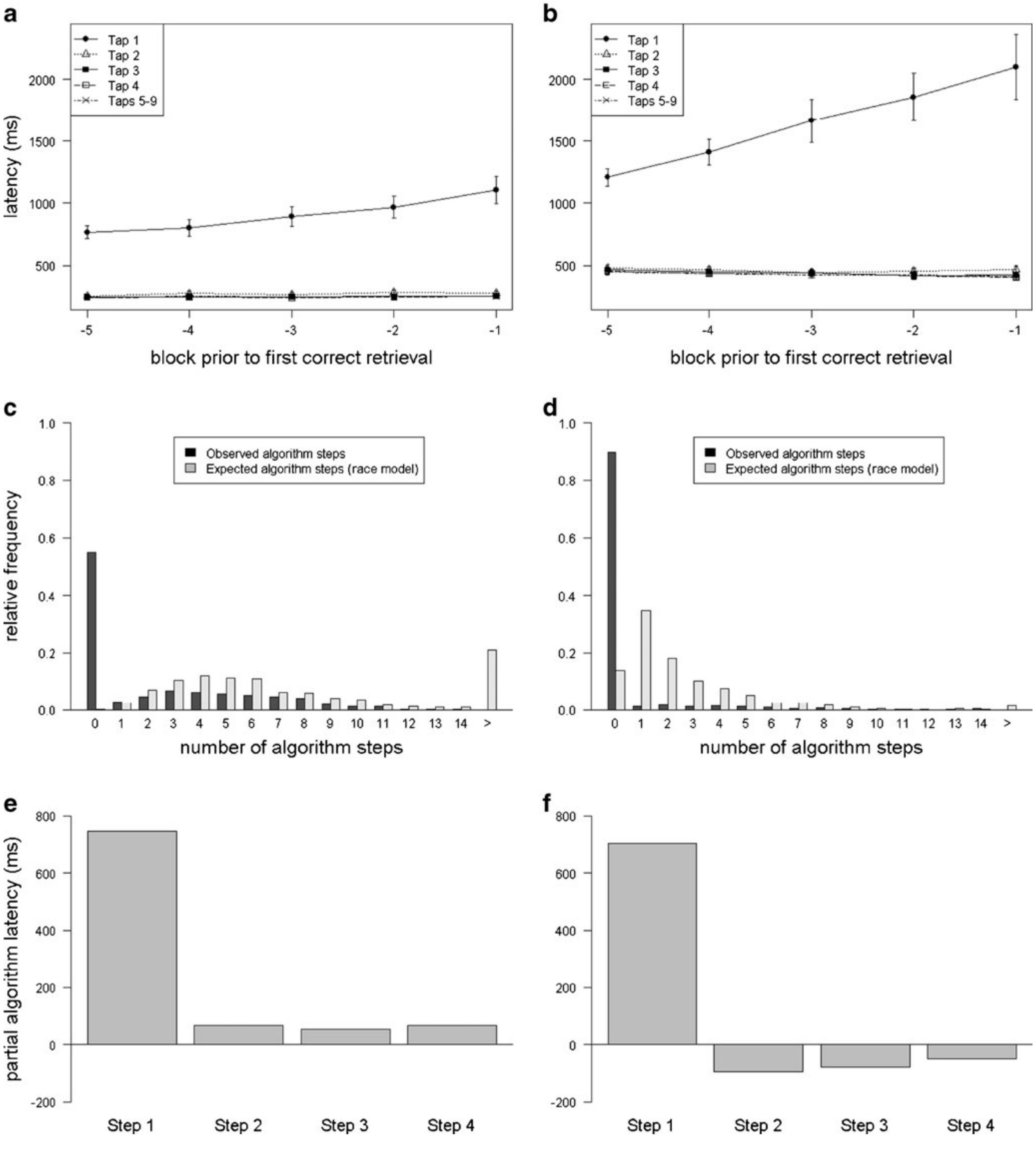

**Fig. 2** Mean algorithm step latencies on the five blocks prior to the first correct retrieval block for Algorithm Steps 1, 2, 3, and 4 and the mean of Steps 5–9 (panels **a** and **b**); relative frequency bar charts of expected (according to a race model) and observed completion frequencies for each algorithm step on the first five correct retrieval trials (**c** and **d**); and partial-algorithm latency difference scores (Sync-Blocks 0–4 vs. Block 1) for Steps 1–4 (**e** and **f**), from both Experiment 1 here (**a**, **c**, and **e**, respectively) and the experiment in Bajic and Rickard (2009); **b**, **d**, and **f**, respectively). For panels **a** and **b**, error bars represent the between-subjects standard errors, computed independently for each sync-block

Blocks −5 through −1 (the pause-effect slope), confirmed by a within-subjects ANOVA, $F(4,\ 104) = 9.63$, $p < .0001$. There was no significant effect of sync-block for Steps 2, 3, 4, or 5–9 ($p > .05$ in all cases).

To test for differences in the pause-effect slope in this experiment, as compared to that observed by Bajic and Rickard (2009), a mixed ANOVA with the factors Experiment and Sync-Block (−5 through −1) was conducted on the subject-level mean latencies for Step 1. There were significant effects of sync-block, $F(4, 232) = 17.25$, $p < .0001$, and group, $F(1, 58) = 17.84$, $p < .0001$. There was also a significant interaction, $F(4, 232) = 3.18$, $p < .02$; the pause slope in the present experiment, while substantial, was significantly smaller than that observed by Bajic and Rickard (cf. Fig. 2a and b).

Errors in Sync-Blocks −5 through −1 were primarily incorrect retrieval attempts (approximately 7.9% of trials in those blocks) and, as in the Bajic and Rickard (2009) study (in which the error rate was approximately 7.1%), followed a pattern of increasing frequencies over Sync-Blocks −5 through −1 (i.e., on blocks approaching the first correct retrieval, at Sync-Block 0).

*Algorithm step execution on retrieval trials* In Fig. 2c, the bar graph shows the frequency with which zero, one, two, three, and so on, algorithm steps were completed on trials during Sync-Blocks 0–4, along with the expected frequencies according to a race model. Sync-Blocks 0–4 were selected for this analysis because those trials tended to have the slowest retrieval latencies, and hence would be expected to exhibit the most algorithm step completion according to a parallel model. In the observed data, retrieval trials corresponded to a step count of zero, and partial-algorithm retrieval trials corresponded to a step count greater than zero but less than the number of steps required to complete the algorithm. Full-algorithm reversions, on which all steps of the algorithm were completed and on which use of the algorithm strategy to generate the response can be presumed, occurred on 27% of trials in Sync-Blocks 1–4 (17% of trials in the Bajic & Rickard, 2009, experiment; data from these trials are not included in Fig. 2c and d).

The expected number of algorithm steps according to a race model was derived as in Bajic and Rickard (2009). First, latencies for each algorithm step during the first block of the training phase were averaged over items for each subject. Prior analyses indicated no significant effects of item 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 in Sync-Blocks 0–4, the expected number of completed algorithm steps according to the race model was estimated by determining the number of algorithm steps that the subject would have been expected to complete on that trial, based on the mean step latencies from Block 1. For example, if a particular subject had a Block 1 mean latency of 1,200 ms for the completion of four algorithm steps, and 1,400 ms for the completion of five steps, then a race model would predict that this subject would complete four algorithm steps on a direct-retrieval trial with an RT of 1,300 ms.

As shown in Fig. 2c, there were fewer algorithm steps completed than would be predicted by the race model. The statistical significance of this effect was tested by first computing, for each trial and each subject, the difference between the number of algorithm steps expected by the race model (always more than zero) and the number of observed steps on each trial (frequently zero), taking the mean of these difference scores over trials for each subject, and then conducting a Wilcoxon signed rank test on those mean difference scores, T+ = 378, $p < .0001$. The failure of the race model to fit the data is driven largely by (1) a much larger than predicted percentage of trials with zero completed algorithm steps, and (2) particularly slow retrievals on about 21% of trials, such that a race model would predict that all algorithm steps should have been completed (represented by the ">" column of Fig. 2c and d). The latter case consisted of partial-algorithm retrieval trials with particularly slow Step 1 latencies.

Despite the strong statistical rejection of the race model, there were many more partial-algorithm retrieval trials in this experiment than in Bajic and Rickard (2009); cf. Fig. 2c and d of the present article) and slightly more completed steps on average (5.2 vs. 4.8). The statistical significance of the difference in the distribution shapes was confirmed by a $\chi^2$ test for independence, $\chi^2(12) = 353.1$, $p < .0001$.[2] Note also that for the tap-only group, but not for the tap–count group, there was a tendency for the observed distribution of completed algorithm steps (beyond the case of zero steps) to mimic the race distribution, albeit with a left-shifted mode.

The race model predicts that the speed of algorithm step execution on partial-algorithm retrieval trials will not be influenced by the race with retrieval. We evaluated that prediction by comparing algorithm step latencies on the partial-algorithm retrieval trials to the step latencies on the first training block. For each subject, the means of the latencies for Algorithm Steps 1, 2, 3, and 4 for the first training block were subtracted from the mean of the matched step latencies on partial-algorithm retrieval trials (due to data attrition, partial-algorithm latencies on subsequent steps were not analyzed). These difference scores are depicted in Fig. 2e, with positive scores indicating slowing relative to Block 1. Contrary to the race prediction, there was a pronounced and highly significant 745-ms pause effect for Step 1, $t(24) = 3.64$, $p < .002$. This result roughly

[2] Here and elsewhere, step bins were combined as needed to meet $\chi^2$ assumptions of expected frequency of no fewer than five.

matches the more than 700-ms partial-algorithm pause effect in the Bajic and Rickard (2009) experiment (Fig. 2f). The latency difference scores for Algorithm Steps 2–4 were 67.88, 54.53, and 65.53 ms, respectively. Despite being more than one order of magnitude smaller than the slowing observed for Step 1, these results were still significant ($p < .05$ in all cases). This result is in contrast to the data from Bajic and Rickard, in which the difference scores for Steps 2–4 did not significantly differ from zero.

These RT effects for Steps 2–4 in the present experiment may simply reflect small increases in algorithm step completion latencies following practice that occurred independently of any retrieval interference effects. For example, it may be that subjects performed the simple repeated keypress task as fast as possible at the outset of the experiment, with no room for a speed-up with practice. A build-up of some fatigue over the course of practice may then yield a slower keypress rate, resulting in the small positive difference scores for Step 2 and onward. In contrast, in the tap–count condition there may be room for skill learning with practice, and a consequent step latency speed-up, that offsets any slowing due to fatigue build-up.

### Discussion

The striking degree of strategy interference in this simple tapping task was not predicted by any of the models that have been developed to explain the temporal dynamics of strategy execution in cognitive skill leaning (e.g., Logan, 1988, 2002; Palmeri, 1997; Rickard, 1997, 2004). Nevertheless, these effects, and in particular the pause effect, may be intuitively plausible. In our experience, it is not uncommon to pause briefly to attempt to remember, say, where one's keys are before searching, or to try to remember a computer key combination before initiating a search of dropdown menus.

The strategy interference effects in this experiment are generally much smaller, however, than those observed by Bajic and Rickard (2009) for the tap–count algorithm, the only exception being the pause effect on partial-algorithm retrieval trials (cf. Fig. 2e and f). Further consideration of the theoretical implications of these results will be deferred until the General Discussion.

## Experiment 2

In this experiment, we investigated whether the results from Experiment 1 would generalize to a different P-M algorithm. Instead of key tapping, the motor task in this experiment required subjects to use the computer mouse to click alternately on target regions on the left and right sides of the computer screen. Subjects were randomly assigned either to a version of the algorithm that required only mouse clicks (*click-only*) or to an algorithm that required both clicks and counts (*click–count*) but that was otherwise nearly identical to the click-only algorithm. The lack of random assignment in the foregoing comparison of Experiment 1 to the Bajic and Rickard (2009) experiment, though not problematic in any obvious way, could have influenced the between-groups effects.

### Method

*Subjects* A total of 36 University of California at San Diego undergraduate students participated for course credit.

*Design and procedure* As with Experiment 1, this experiment consisted of a one-block learning phase and a 45-min training phase, with similar instructions prior to each phase. The Appendix lists the numerical visual stimuli, number of algorithm steps, and correct vocal responses for all 10 stimulus–response items. Prior to each trial, a small text box containing the words “Click Here to Begin” would appear just below the center of the screen, and clicking this box would immediately initiate the trial. (This ensured that the mouse pointer was always at roughly the same position at the start of each trial.) During each trial, the screen was divided into two equal halves by a thin vertical line down the center. At the beginning of each trial, a 1.5 × 1.5 cm square containing the numerical stimulus for that trial would appear at the center of the screen, and a green rectangle (6.5 × 11 cm) would appear on either the left or right half of the screen (randomly determined), centered vertically on the screen, with one edge positioned 2.5 cm away from the vertical line that divided the screen. This rectangle will henceforth be referred to as the *target*. If the target was clicked with the mouse pointer, it would immediately move to its equivalent position on the opposite half of the screen. Each mouse click of the target constituted one algorithm step.

For subjects in the click-only group, the algorithm simply involved repeated, alternating clicking of the left- and right-side target rectangles. After the required number of algorithm steps for a given stimulus had been completed, the mouse pointer and the trial's numerical stimulus would vanish from the screen. Simultaneously, the target rectangle would change from green to black, with white text inside it presenting both the word STOP, and the appropriate numerical answer for that trial's vocal response.

For subjects in the click–count group, the algorithm further required silent counting, starting with the value of the numerical stimulus, each time the target was clicked. That is, if the numerical stimulus for a given trial were 21, the subject would silently count “22” the first time that he or she clicked the target square, “23” the second time, and so forth. When the full set of required algorithm steps had been completed, the same events described above for the click-only group would occur, except that the white text at the end of the trial would only present the word STOP, and not the answer itself. At that point, subjects were to speak aloud their final count.

## Results and discussion

Following the rules described for Experiment 1, data from 4 subjects (2 in each group) were excluded due to high rates of voice-key errors, and data from 2 subjects in the click–count group and 1 subject in the click-only group were excluded due to low overall accuracy. Among the remaining subjects, voice-key errors occurred on approximately 3.85% of trials in the click–count group and approximately 4.24% of trials in the click-only group.

For subjects in the click-only group, mean accuracy was initially perfect (100%) on the first training block, fell to its lowest value (90.23%) on Block 6, then rose again to 98.67% by the 26th block, the final block that all subjects completed. For subjects in the click–count group, mean accuracy was initially 89.31% on the first training block, fell to its lowest value (85.58%) on Block 10, then rose to 93.41% by the 22nd block, the final block that all subjects completed. Figure 3 depicts the means of the subject-level mean RTs, their *SD*s, and the proportions of trials on which the retrieval strategy was used, for the click-only group (panels a, c, and e, respectively) and the click–count group (panels b, d, and f).

*Algorithm step latencies on trials preceding the first correct retrieval* Mean algorithm step latencies from Sync-Blocks −5 through −1 are shown in Fig. 4a (click-only) and b (click–count). The mixed-factors ANOVA on the algorithm first-step data revealed significant effects of sync-block, $F(4, 108) = 16.5$, $p < .0001$, group, $F(1, 27) = 30.3$, $p < .0001$, and their interaction, $F(4, 108) = 2.47$, $p < .05$. The interaction effect confirms the shallower pause-effect slope in the click-only group. The simple effects of sync-block were also confirmed in separate ANOVAs performed for each group: $F(4, 52) = 9.32$, $p < .0001$, for the click–count group; and $F(4, 56) = 8.91$, $p < .0001$, for the click-only group. As in Experiment 1, there were no significant effects involving sync-block when this analysis was performed on subsequent algorithm steps.

Errors in Sync-Blocks −5 through −1 were primarily incorrect retrieval attempts (approximately 9.5% and 6.9% of trials for the click–count and click-only groups, respectively) and followed a pattern of increasing frequency on blocks approaching Sync-Block 0.

*Algorithm step completion on retrieval trials* The distribution of steps completed on partial-algorithm retrieval trials in Sync-Blocks 1–4 is shown in Fig. 4c and d for the click-only and click–count groups, respectively. Full-algorithm reversions occurred on 25% of these trials in the click-only group and on 26% of these trials in the click–count group. The Wilcoxon tests allowed rejection of the race model for both groups: click-only, T+ = 102, $p < .0007$; click–count, T+ = 105, $p < .0002$. A $\chi^2$ test of independence, comparing the observed distribution of completed algorithm steps on retrieval trials for the two groups, was highly significant, $\chi^2(9) = 205.26$, $p < .0001$. Among partial-algorithm retrieval trials, the mean number of completed steps was slightly greater in the click-only group (4.4 vs. 4.1). As in Experiment 1, the shape of the distribution of completed P-M algorithm steps (beyond zero steps) is similar to that predicted by the race model, albeit with a left-shifted mode. The left-shifted mode was expected, given the pause effects on partial-algorithm retrieval trials in Experiment 1 and in this experiment (described below). When considering only partial-algorithm retrieval trials, the similarity of the distribution shape to that predicted by the race model suggests that, once a P-M algorithm is initiated, the temporal dynamics of strategy execution may approximate a race.

For the click–count group, the partial-algorithm step latency difference scores (calculated as in Exp. 1) for Algorithm Steps 1–4 were 607.6, –8.43, 128.31, and 62.16 ms, respectively (Fig. 4f). There was significant slowing for Step 1, $t(9) = 2.35$, $p < .05$, but tests for subsequent steps did not reach significance. For the click-only group (see Fig. 4e), the partial-algorithm difference scores for Algorithm Steps 1–4 were 459.49, 66.42, 43.05, and 13.06 ms, respectively. There was again significant slowing for Step 1, $t(14) = 3.15$, $p < .008$, but not for subsequent steps (although the trend was toward positive scores, as in Exp. 1).

It is worth noting that the pause effect on partial-algorithm retrieval trials was larger for the tap-only group of Experiment 1 than for the click-only group of this experiment. Although we have no strong theoretical account of that effect, it may reflect differences in the algorithms that may affect the subjects’ choices of when to initiate the algorithm. Perhaps, for example, the initial mouse movement of the click-only algorithm was less demanding on attention than was initiating a keypress, allowing it to be initiated earlier during the course of the retrieval attempt.

In summary, the results of this experiment replicate those for the comparison of the tap-only (Exp. 1) and tap–count (Bajic & Rickard, 2009) algorithms; in the click-only group, there was a shallower (but still robust) algorithm pause-effect slope and a larger number of partial-algorithm retrieval trials. The pause effect on partial-algorithm retrieval trials was observed for both groups.

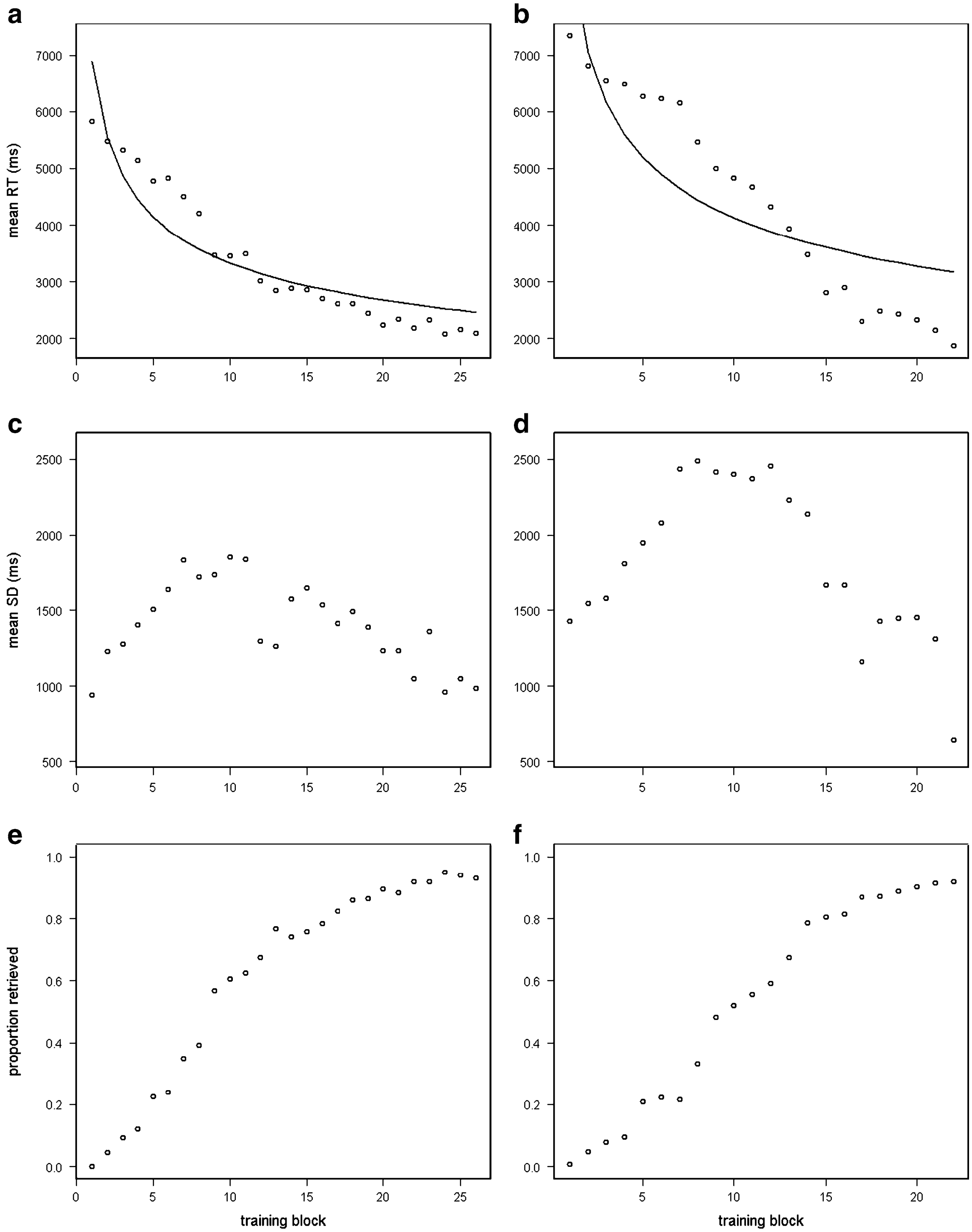

**Fig. 3** Mean response times (RTs) with the best-fitting three-parameter power functions (panels **a** and **b**), mean standard deviations (**c** and **d**), and the proportions of trials on which the direct-retrieval strategy was selected (**e** and **f**), as functions of training block for the click-only group (**a**, **c**, and **e**, respectively) and the click–count group (**b**, **d**, and **f**, respectively) of Experiment 2

In contrast to Experiment 1, the number of required clicks in the click-only condition of this experiment always matched the number of counts between the stimulus number and the correct response number. Although this feature of the design has the advantage of equating the click–count and click-only algorithms to the maximum extent possible (i.e., the numbers of clicks for each stimulus–response pair were always identical in the two conditions), it raises the possibility that subjects in the click-only condition could, in principle, have performed

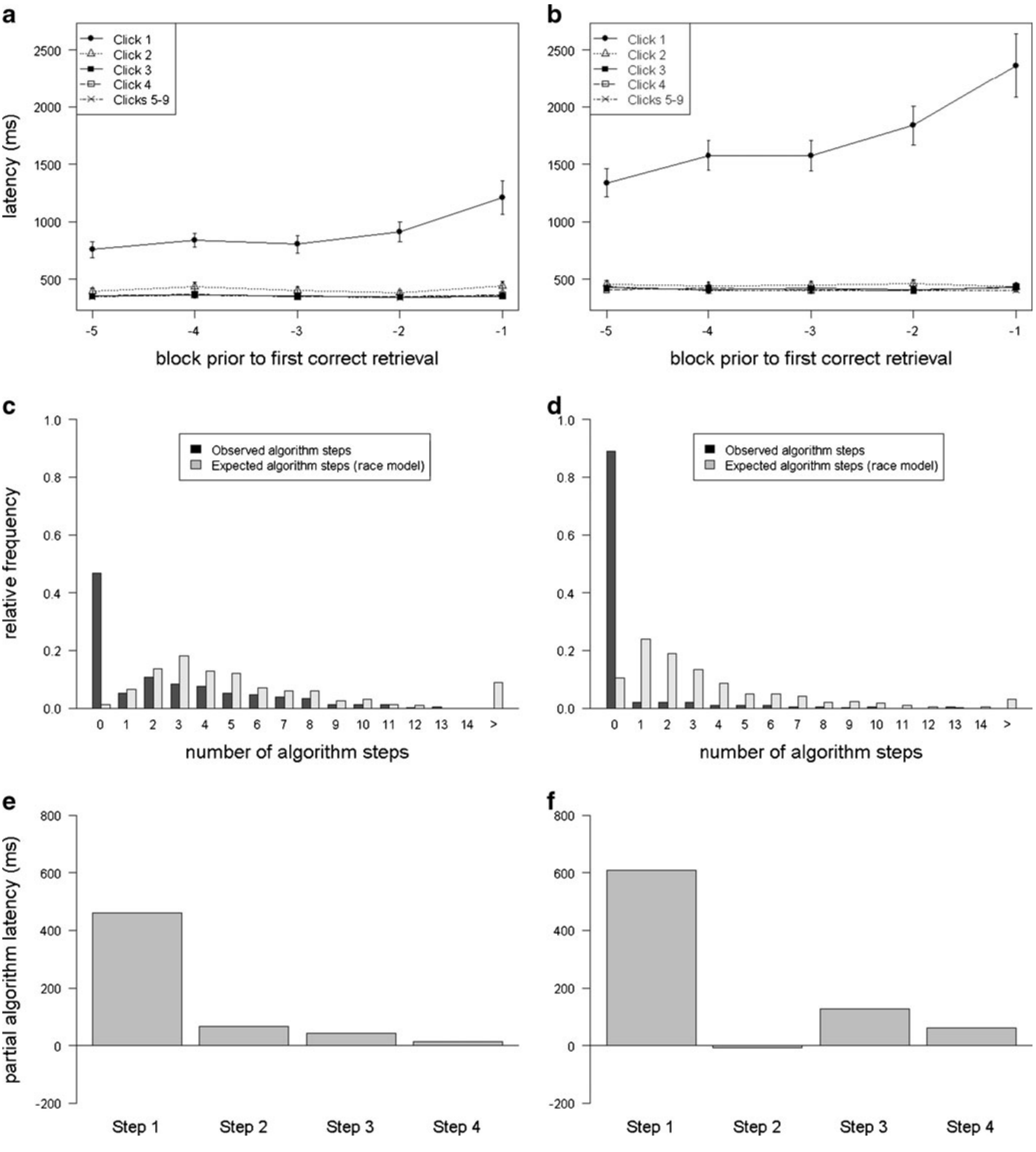

**Fig. 4** Mean algorithm step latencies on the five blocks prior to the first correct retrieval block for Algorithm Steps 1, 2, 3, and 4 and the mean of Steps 5–9 (panels **a** and **b**); relative frequency bar charts of expected (according to a race model) and observed completion frequencies for each algorithm step on the first five correct retrieval trials (**c** and **d**); and partial-algorithm latency difference scores (Sync-Blocks 0–4 vs. Block 1) for Steps 1–4 (**e** and **f**), for both the click-only group (**a**, **c**, and **e**, respectively) and the click–count group (**b**, **d**, and **f**, respectively) of Experiment 2. For panels **a** and **b**, error bars represent the between-subjects standard errors, computed independently for each sync-block

silent (subvocal) counting as part of algorithm execution. Two factors, however, speak against that possibility. First, use of a silent counting strategy in the click-only group would have made the algorithm essentially identical to that of the click–count group, and as such, the large group performance differences should not have occurred. Second, if even a subset of subjects in the click-only group did some silent counting, then we would expect the pause-effect slope on algorithm trials in Experiment 2 to be substantially larger than in Experiment 1, wherein silent counting would have no conceivable adaptive value (indeed, it would presumably have interfered with performance in that experiment). No such effect was evident. Finally, given the overall pattern of results, the possibility that a few subjects employed counting in the click-only group of Experiment 2 does not appear to compromise the overall conclusions to be emphasized in the General Discussion.

Before we proceed, it is important to point out that the subject instructions in these experiments, which stated that many subjects had reported good results when they attempted to use both strategies at the same time, might have constituted a demand characteristic, leading to a larger pause-effect slope and more partial-algorithm retrieval trials than would otherwise have been observed. Under less leading instructions, the results for the P-M algorithms might have looked more similar to those for LTM algorithms. Even if that were the case, however, our results provide an "existence proof" that, after a pause, subjects can initiate and execute simple P-M steps while running the retrieval strategy to completion in parallel. Thus, it is unlikely in our view that any instructional demand characteristic would seriously bias the theoretical conclusions, noted below, regarding the algorithm types and processing stages at which parallel strategy execution is and is not possible.

## General discussion

The results for the click–count group of Experiment 2, as well as for the tap–count task of Bajic and Rickard (2009), are consistent with strategy choice accounts such as CMPL (Rickard, 1997). When the algorithm entails a series of LTM retrieval steps—even steps as simple as counting—strategy execution appears to be a one-at-a-time phenomenon.

The results for the P-M (i.e., the tap-only and click-only) algorithms, however, were not predicted by any of the currently applicable models. The race account, which has been adopted in fits to data of both the instance theory (Logan, 1988) and the exemplar-based random walk model (Palmeri, 1997, 1999), cannot explain the substantial strategy interference that was observed. The race account was viable for P-M algorithms prior to this study, but it can now be eliminated from consideration. Any limited-capacity parallel strategy execution model that assumes a constant capacity demand from both strategies throughout all steps of their execution can also be rejected. Simple strategy-choice models are also challenged by these results. The CMPL model, which was proposed for the case of LTM algorithms, leaves open the possibility that strategy interference for P-M algorithms may occur through some mechanism other than LTM retrieval interference, but it provides no process account of that interference.

### Toward a more generalized theory of strategy execution

A new or modified model of the temporal dynamics of strategy execution, and in particular the shift to retrieval with practice, is needed to accommodate the full set of empirical results reported here and in the literature. Although formal development of such a model is beyond the scope of this article, some of its basic properties can now be identified.

First, it is clear that the retrieval strategy, once it becomes competitive through training, interferes markedly with initiation of both LTM and P-M algorithms. For the P-M case, but not the LTM case, this interference appears to dissipate during the course of a successful retrieval attempt, as indicated by the frequent partial-algorithm retrieval trials for the P-M algorithms only. Once the P-M algorithm is initiated on partial-algorithm retrieval trials, however, algorithm step execution is slowed only minimally, if at all, by the ongoing execution of the retrieval strategy (i.e., there is no pause effect on partial-algorithm retrieval trials for Step 2 and onward). The retrieval strategy, then, profoundly interferes with initiation of, but not the subsequent execution of, P-M algorithms. These patterns suggest a strategy initiation bottleneck, such that execution of a retrieval strategy delays initiation of the P-M algorithm strategy but does not affect execution of that strategy (i.e., execution of the P-M steps) once initiated. We leave open the question of whether the hypothesized strategy initiation bottleneck is an immutable property of the cognitive architecture, or rather is an adaptation to capacity limits that render parallel strategy initiation inefficient.

Given our proposal that the two strategies cannot be initiated simultaneously, subjects must choose on each trial to initiate either the retrieval strategy first or the algorithm strategy first. Phenomena such as the pause-effect slope show that, on at least a large proportion of trials, subjects choose to initiate and attempt the retrieval strategy first, once it becomes competitive through training. The data do not speak conclusively to the possibility that, on some other trials, subjects may first initiate the P-M algorithm (followed by the retrieval strategy), although some evidence for that possibility is considered below.

A remaining question is why the interference that blocks P-M algorithm initiation dissipates over the course of executing the retrieval strategy. One possibility is that an LTM retrieval involves a graded capacity demand, such that, at some point during the retrieval attempt, sufficient attentional capacity becomes available to initiate the P-M algorithm without compromising ongoing execution of the retrieval strategy. An alternative account is that LTM retrieval can, for the present purposes, be conceived of as having two sequential stages. By this account, the first stage acts as a bottleneck, completely blocking initiation of the algorithm. During the second retrieval stage, however, there is no bottleneck for P-M algorithms, and they can be initiated and executed without further interference. This two-stage retrieval account would be most satisfying if the two stages could be differentiated in psychologically simple terms. One possibility is that the first stage involves the entire retrieval event up to the point of answer generation, and the second stage involves conversion of the generated answer into an overt articulatory response.

Given the large interference effect that the retrieval strategy has on P-M algorithm initiation, it would seem more efficient for subjects to first initiate the P-M algorithm and then to initiate retrieval. Presumably, the latency for initiation of the P-M algorithm would be relatively small (perhaps on the order of 100 ms), with minimal variability. The data in fact suggest this possibility. Post-hoc analyses of the subject-level data showed that, whereas many subjects exhibited large pause-effect slopes on algorithm trials and large pause effects on partial-algorithm retrieval trials (some nearly as steep as those observed for the matched LTM algorithms), a few subjects exhibited minimal pause effects. Subjects with minimal pause effects might have adopted a strategy of initiating the P-M algorithm first. This pattern of extreme subject-level variability was not present in the LTM algorithm groups. These individual difference patterns, if confirmed, could be important to future model development and could provide insight into factors that determine whether a subject does or does not adopt optimal strategy scheduling.

Another core feature of any comprehensive model of the shift to retrieval will be an account of the apparent one-at-a-time strategy execution (i.e., a pure strategy choice) that occurs for LTM retrieval algorithms. The most obvious possibility is that, whereas a P-M algorithm and the direct-retrieval strategy (e.g., an LTM retrieval) can proceed in parallel once initiated, two LTM retrievals (i.e., the direct-retrieval strategy and each step of an LTM algorithm) cannot run in parallel during any stage of their processing. Thus, there may be two sources of strategy interference in the general case: (1) strategy initiation interference, which occurs for all algorithm types, and (2) a more extensive interference effect that precludes two simultaneous LTM retrievals (see Nino & Rickard, 2003, for supporting evidence for the case of two retrievals from a single cue). Consider this account in terms of the two-stage retrieval model outlined above. Whereas it might be possible to initiate and execute a simple, repetitive motor action during the second stage of retrieval, it might not be possible to initiate another retrieval, even one as simple as counting, during either the first or second stage.

The strategy initiation bottleneck hypothesis can explain the finding that partial-algorithm retrieval trials are far less frequent for LTM retrieval algorithms than for P-M algorithms. Given that the direct-retrieval strategy cannot be executed in parallel with an LTM retrieval algorithm, partial-algorithm retrieval trials would entail the inefficient sequence of steps noted in the introduction: (1) successful completion of direct retrieval (providing an account of the partial-algorithm pause effect), (2) a subsequent shift to algorithm execution for a few steps, yet (3) an early termination of the algorithm in favor of speaking the previously retrieved response. Thus, in the case of LTM algorithms, partial-algorithm retrieval trials have a latency cost with no apparent benefit, and they would be expected to occur infrequently. For P-M algorithms, on the other hand, partial-algorithm retrieval trials appear to reflect an efficient, postinitiation strategy race that could in principle reduce mean RTs, relative to one-at-a-time strategy execution.

The substantially smaller algorithm pause effect for the P-M algorithms might also be accommodated by the initiation bottleneck hypothesis. Given that P-M algorithms, but not LTM algorithms, can be initiated prior to the retrieval attempt running to completion, the pause effect should then be reduced.

The strategy initiation bottleneck hypothesis complements prior work showing that a feeling of knowing influences strategy choice (Delaney et al., 1998; Reder & Ritter, 1992; Schunn et al., 1997). Given that an initiation bottleneck exists, some mechanism must exist that leads to selection of a strategy for first initiation. A feeling of knowing based on, perhaps, increasing familiarity with the stimuli over training blocks (Reder & Ritter, 1992; Schunn et al., 1997) could be one such mechanism.

Our results can also be related to the threaded cognition framework of Salvucci and Taatgen (2008). Within that theoretical framework, for any multitask situation, performance bottlenecks can arise at the level of any resource (e.g., perceptual, motor, or declarative memory resources) shared by the competing tasks. This includes possible bottlenecks in a central procedural resource, where production rules (e.g., “if a particular stimulus is present, perform a particular action”) fire within the context of various goal states.

From the perspective of that framework, the algorithm and direct-retrieval strategies within our studies could be viewed as competing threads. The brief initiation bottleneck observed for the P-M tasks (tap-only and click-only) could reflect a bottleneck at the level of the central procedural resource, whereas the more pronounced bottleneck observed for the LTM algorithm tasks (tap–count and click–count) might be caused by bottlenecks in both the central procedural resource and the declarative memory resource. Further refinement of that model for this case, however, would be needed before determining whether it can account for the full pattern of results.

## Conclusions

These experiments constitute the first systematic investigation of the shift to retrieval for the general class of P-M algorithms, as well as the first matched comparison of P-M and LTM retrieval algorithms. The data support the following empirical and theoretical inferences: (1) for the case of LTM retrieval algorithms, a pure strategy choice process, in line with that assumed by the CMPL model (Rickard, 1997), governs performance on at least the great majority of trials; (2) there is substantial strategy interference even for very simple P-M algorithms that involve little more than sequential tapping or clicking; (3) the interference effects for the P-M case, however, are generally of smaller magnitude; (4) none of the models that have been considered in the literature (i.e., race, limited-capacity parallel, and choice models) can, in their current forms, accommodate either the results for P-M algorithms or the performance differences between the two types of algorithms; (5) it appears that any comprehensive model of the shift to retrieval will have to incorporate both a ubiquitous strategy initiation bottleneck (or at least an extremely disruptive capacity limit on strategy initiation) and an account of the greater interference—and resulting pure strategy choice behavior—for the case of LTM retrieval algorithms.

**Author note** We thank Kristin West, Lisa Graves, Jennifer Um, Shivali Agarwal, Samir Patel, Jon Strunk, and Caitlin Oldenkamp for their assistance with data collection.

## Appendix

**Table 1** Phase 2 Stimulus–Response Pairings

| Stimuli | Responses | Algorithm Steps |
|---|---|---|
| 20 | 34 | 14 |
| 21 | 36 | 15 |
| 22 | 32 | 10 |
| 23 | 35 | 12 |
| 24 | 33 | 9 |
| 25 | 37 | 12 |
| 26 | 41 | 15 |
| 27 | 40 | 13 |
| 28 | 39 | 11 |
| 29 | 38 | 9 |

For Experiment 1, the algorithm steps listed above were randomly reassigned to the various stimulus-response pairings at the start of the experiment, following randomization constraints listed in the Method section of that experiment.

## References

Bajic, D., & Rickard, T. C. (2009). The temporal dynamics of strategy execution in cognitive skill learning. *Journal of Experimental Psychology. Learning, Memory, and Cognition, 35,* 113–121.

Bourne, L. E., Raymond, W. D., & Healy, A. F. (2010). Strategy selection and use during classification skill acquisition. *Journal of Experimental Psychology. Learning, Memory, and Cognition, 36,* 500–514.

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.

Fernandes, M. A., & Moscovitch, M. (2000). Divided attention and memory: Evidence of substantial interference effects at retrieval and encoding. *Journal of Experimental Psychology. General, 129,* 155–176.

Fernandes, M. A., & Moscovitch, M. (2003). Interference effects from divided attention during retrieval in younger and older adults. *Psychology and Aging, 18,* 219–230.

Hertzog, C., Touron, D. R., & Hines, J. C. (2007). Does a time-monitoring deficit influence older adults' delayed retrieval shift during skill acquisition? *Psychology and Aging, 22,* 607–624.

Jenkins, L., & Hoyer, W. J. (2000). Instance-based automaticity and aging: Acquisition, reacquisition, and long-term retention. *Psychology and Aging, 15,* 551–565.

Kahneman, D. (1973). *Attention and effort*. New York: Prentice Hall.

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. (2002). An instance theory of attention and memory. *Psychological Review, 109,* 376–400.

Luce, R. D. (1991). *Response times: Their role in inferring elementary mental organization*. New York: Oxford University Press.

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,* 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,* 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.

Rawson, K. A. (2004). Exploring automaticity in text processing: Syntactic ambiguity as a test case. *Cognitive Psychology, 49,* 333–369.

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,* 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,* 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,* 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,* 1381–1386.

Rickard, T. C., & Bajic, D. (2004). Memory retrieval given two independent cues: Cue selection or parallel access? *Cognitive Psychology, 48,* 243–294.

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,* 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,* 359–394.

Salvucci, D. D., & Taatgen, N. A. (2008). Threaded cognition: An integrated theory of concurrent multitasking. *Psychological Review, 115,* 101–130.

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,* 3–29.

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

Touron, D. R., & Hertzog, C. (2009). Age differences in strategic behavior during a computation-based skill acquisition task. *Psychology and Aging, 24,* 574–585.

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.

Townsend, J. T., & Nozawa, G. (1997). Serial exhaustive models can violate the race model inequality: Implications for architecture and capacity. *Psychological Review, 104,* 595–602.