Serial Attention as Strategic Memory

Erik M. Altmann (altmann@gmu.edu)
Wayne D. Gray (gray@gmu.edu)
Human Factors & Applied Cognition
George Mason University
Fairfax, VA 22030

Abstract

Serial attention is the process of focussing mentally on one item at a time. This process has two phases: attention switching and attention maintenance. Attention switching involves rapidly building up the activation of a new item to dominate old items. Attention maintenance involves letting the current item decay while in use to prevent it from intruding on the next item later on. SASM, a model based on this analysis, suggests that this balance of high initial activation followed by gradual decay reflects a strategic adaptation to task demands on one hand and principles of memory on the other. The model makes novel and accurate predictions about response times and error rates, integrates past use and current context as memory activation sources, and integrates attention switching and attention maintenance into one unified account.

Introduction

To think about mental attention, we can adopt a metaphor from visual attention (e.g., Posner, 1980) and imagine a spotlight directed internally at memory and focussed on the current thought. A sequence of thoughts, or *serial attention*, would then involve moving the spotlight around — maintaining it at one position for a time, then switching it to the next, and so forth. Serial attention is basic to many cognitive activities, from searching a memory set for a probe, to achieving one goal and switching to another during problem solving.

Understanding the *costs* of paying attention and where they occur is crucial to understanding serial attention as a whole. For example, the cost of switching attention has often been interpreted as the time needed to pull a mental lever that switches attention from one task to another (e.g., Garavan, 1998; Gopher, Greenshpan, & Armony, 1996; Rogers & Monsell, 1995). Under additive-factors logic, this switch cost would be reflected in the first measurement taken after the lever is pulled. By extension, this switching action, once complete, should have no effect on the train of thought, which simply continues down the new track. Thus the mental-lever view suggests that attention switching is an active process but attention maintenance is essentially a passive system state and should produce stable performance.

Unfortunately for the mental-lever view, attention maintenance has its own cost, measured as a gradual increase in response time between attention switches (Altmann & Gray, 1999). Our initial explanation for this *maintenance cost* was in terms of interference in memory (Altmann & Gray, 1998). In brief, our proposal was that interference among trials made attention maintenance more difficult over time, and that this interference was "released" by the act of switching attention. However, the functional role of this interference was not clear. Did it satisfy some constraint other than fitting the data? Also, though our explanation was grounded in a cognitive theory (ACT-R; see Anderson & Lebiére, 1998), it ignored basic operational principles of that theory, including strengthening, decay, and noise in memory. The mechanisms representing these principles were simply "turned off" in our computational ACT-R model.

Here we present an expanded model of serial attention that explains maintenance cost and offers a preliminary account of switch cost. From the previous model we carry over the premise that serial attention is essentially a memory phenomenon. We now also adopt the premise that memory in serial attention acts like memory in general in that it strengthens with use and decays without, and is subject to noise like any other data channel. The implication is that cognition must employ active processes or *strategies* that manipulate memory strength to cope with these constraints. These memory-manipulation strategies are responsible for the costs of maintaining and switching attention. We thus characterize *serial attention as strategic memory*, or *SASM*.

We first briefly review our serial attention paradigm and the evidence for maintenance cost. Next, we argue that maintenance costs are inevitable given our premises. We then develop the parameters of the SASM model in geometric terms, and develop and test novel predictions against empirical data. We end by discussing implications of the model for such questions as cognitive workload. The appendix presents an algebraic derivation of the model. We have also implemented SASM as a computational ACT-R model and fit Monte Carlo simulations to our empirical data, but this work is not reported here.

The Paradigm and Sample Data

Our serial attention paradigm involves giving participants two simple tasks and periodically issuing an instruction to switch from one task to the other. For example, the tasks might be to judge a single digit (from the set 1, 2, 3, 4, 6, 7, 8, 9) as Odd or Even or as High (> 5) or Low (< 5). In a typical experiment, trials are presented in blocks. Each block begins with an *instruction trial* saying which task to do, for example "Odd Even." This trial is followed by a sequence of classification trials. These are single digits and provide no clue to the current task. The run of classification trials is interrupted by a second instruction trial, which may or may not indicate the same task as the first instruction trial. The second instruction trial is followed by a second run of classification trials, followed by feedback on accuracy and response time for that block. A block contains 20 classification trials, and the second instruction trial occurs randomly between the 7th and 14th classification trials. In a typical experiment, participants receive 192 blocks, for a total of 384 instruction trials (192 for each task). Responses to all trials are self-paced and there is no calibrated inter-trial interval. All stimuli appear at the same location in the center of the screen.

Data from this paradigm appear in Figure 1 (from Altmann & Gray, 1998). The abscissa shows the first seven classification trials after the second instruction trial in a block, with *trial position* meaning position relative to the instruction. Switch cost occurs on P1, in that response time (RT) is substantially slower than on P2. Maintenance cost is the gradual slowing from P2 to P7.

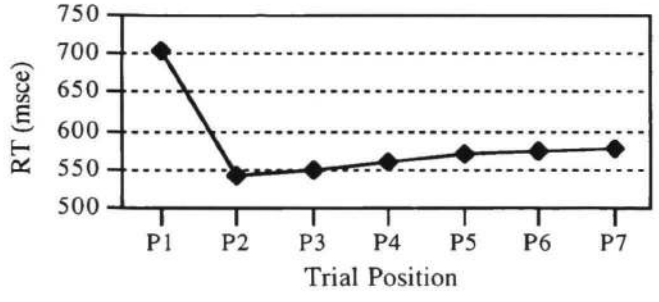

Figure 1: Response time (RT) on post-instruction trials. Maintenance cost is the slowing trend from P2 to P7.

Memory for Instructions

An important distinction is that between a *task* and an *instruction*. A task is semantic and in our paradigm there are only two (e.g., OddEven and HighLow). An instruction is semantic and episodic – there are as many instructions as there are instruction trials (384 in a typical experiment). An instruction specifies what task to do *now*, superseding all previous instructions.

An assumption central to SASM is that each instruction is encoded as a distinct trace in memory. No instruction is instantaneously forgotten or deleted. Rather, old instructions linger and may interfere with the current one. Cognition must cope with this potential interference by encoding each new instruction to resist intrusions from old ones.

The mechanism for coping with this interference is grounded in basic laws of memory. Under the law of exercise a memory element becomes stronger with use, and under the law of forgetting a memory element becomes weaker when unused. Both laws are implemented in ACT-R in the functions governing the activation of declarative memory elements (*chunks*). A chunk *use* constitutes either a new *encoding* of that chunk in memory or a *retrieval* of that chunk from memory.

When cognition attempts a retrieval, ACT-R's memory system returns the most active chunk. This reflects the *rational memory* assumption, in which activation represents the memory system's best guess at the chunk most likely to be needed now (Anderson, 1990). Activation in ACT-R has two components, one representing a chunk's history of use and the other representing the chunk's relevance to the current context. *Base-level activation* represents history of use. For example, a period of concentrated rehearsal or encoding makes a chunk very active. *Associative activation*, which we refer to as *priming*, represents relevance to the current context. Priming accounted for maintenance cost in our previous model (Altmann & Gray, 1998); in the current model it complements base-level activation to improve memory accuracy, as we discuss later.

The base-level activation of instructions is a critical factor in serial attention, as illustrated in Figure 2. The abscissa shows two contiguous *runs* of trials, with the *previous* run on the left and the *current* run on the right. Each run is governed by an instruction (I_P and I_C, respectively). The ordinate shows instruction activation. The top two curves (in solid ink) show each instruction at peak activation initially (at $P1_P$ and $P1_C$) then decaying throughout its run. When I_P gives way to I_C (at $P1_C$), it decays somewhat faster because it is no longer being used.

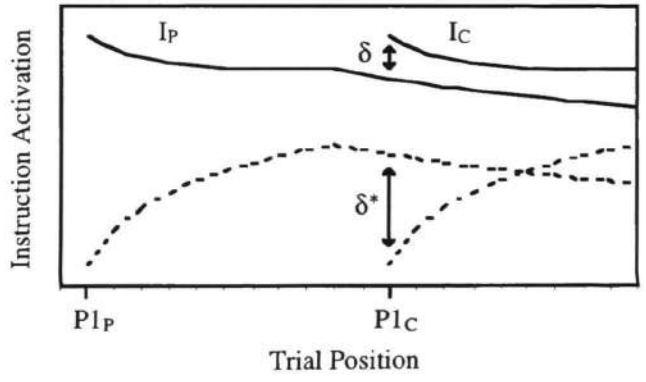

Figure 2: When instructions decay (solid curves), the current instruction (I_C) is always stronger than the previous instruction (I_P), by amount δ at $P1_C$. Were instructions to strengthen (dashed curves), serial attention would fail because I_C would be weaker than I_P, by amount δ^* at $P1_C$.

The important relationship in Figure 2 is between the activation of I_C and I_P. Once I_C is encoded, both instructions coexist in memory because I_P is not completely forgotten. However, the top two curves show I_C always being more active than I_P. Under the rational memory assumption, this ensures that the memory system returns I_C on each trial during the current run, producing correct performance. I_C dominates I_P because both activation curves slope downwards — if all instructions decay from a high initial level, the newest one will always be the most active.

The bottom two curves in Figure 2 (in dashed ink) show an intuitive but problematic interpretation of the law of exercise in this paradigm. The intuition is that if an instruction is retrieved on each trial, it should *gain* activation over trials. However, this would mean that it ends up more active than it begins. At $P1_C$, therefore, I_C would be *less* active than I_P by amount δ^*. Under the rational memory assumption, this would preclude correct performance. Hence, instruction activation must start high and end low.

Thus decay in episodic memory is a necessary condition for serial attention and this decay implies maintenance cost. Time to retrieve a memory element, in ACT-R as in other memory models, is a function of its activation, with higher activation implying faster retrieval. If instructions decay from when they are first used, retrieval time will increase on each subsequent trial within a run.

Parameters of Instruction Memory

We have argued that instructions must decay *ab initio* for serial attention to be possible. We next examine four parameters that determine what initial level of activation is necessary to produce such decay. Here we use a geometric notation, leaving the algebra to the Appendix.

One parameter is noise, which we assume affects memory much as it affects any data channel. Following ACT-R, we take noise to be manifested as transient increases or decreases in chunk activation. Each chunk has an expected level of activation, but its actual level on a given retrieval cycle varies according to a logistic (roughly normal) distribution (Anderson & Lebiére, 1998, ch. 3). This activation variance can cause memory errors and in turn performance errors.

A memory error occurs when noise makes the target less active than some other chunk on a given retrieval cycle. The likelihood of such an error depends both on the amount of noise in the system and on how active the target is, on average, compared to other chunks. This is illustrated in Figure 3, which shows activation distributions for I_C (the target) and I_P. Activation is now on the abscissa and the probability of an instruction having a given activation is on the ordinate. Noise is reflected in the dispersion of each distribution. This dispersion is one factor determining the overlap of the activation distributions. The greater the overlap, the more likely I_P will be retrieved in place of I_C, and hence the greater the number of memory errors.

The other factor affecting memory error is δ, the difference in expected activation between I_C and I_P (Figure 3). This difference, resembling the d' of signal detection theory, is itself a function of three parameters, two affecting base-level activation and one affecting associative activation.

The first parameter affecting δ is the amount of time spent encoding the instruction while it is visible on the display. We assume that more time spent encoding the instruction means more base-level activation for the instruction chunk, consistent with memory paradigms in which stimulus exposure and trace strength are taken to be synonymous. With respect to Figure 3, a longer encoding time would shift I_C to the right, thereby increasing δ. Because instruction trials are self-paced, participants can wait to dismiss the instruction until it is sufficiently encoded. Thus, encoding time is under strategic control.

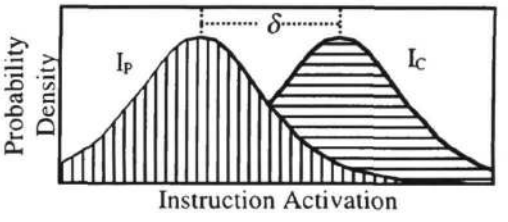

Figure 3: Activation distributions of the previous (I_P) and current (I_C) instructions. The less overlap between them the more likely I_C will be retrieved.

The second parameter affecting δ is run length. As this grows, all else being equal, so does the amount of base-level activation coming from trial-by-trial retrieval as opposed to initial encoding. In Figure 2, each curve decays at first and then flattens out; with more trials per run but no extra encoding, the curve would inflect before the end of the run and begin to slope upwards. In Figure 3, greater run length (all else being equal) would shift I_P to the right, decreasing δ and thus increasing the chance of memory error.

The third parameter affecting δ is associative activation, or priming from cues in the cognitive context. For priming to contribute to δ, some cue must prime I_C more than it primes I_P. This might occur, for example, were the trial stimulus to cue the task. In a variant on our paradigm, the two tasks might be OddEven and ConsonantVowel (instead of OddEven and HighLow), each with a different stimulus set (i.e., numbers vs. letters). In this case, the stimulus itself would be an effective cue for I_C. In contrast, in our paradigm the stimulus (e.g., always a number) affords either task, making it unhelpful as a cue.

However, not all cues need be external. One likely *internal* cue is a residual memory for the previous trial. Although this may be weak, it may play a role in priming the current trial, producing a kind of repetition effect. Within a run, on trial positions after P1, the task performed on the previous trial primes I_C but does not I_P, which specifies the other task. Thus, the previous trial increases δ for the current trial by contributing to the expected activation of I_C.

In sum, we have identified four parameters of memory for the current instruction, or more generally of memory for the current item in serial attention. Memory noise determines activation dispersion; encoding time, run length, and priming affect an instruction's expected total activation. We next examine predictions of a closed-form model built on these parameters.

Predictions of the Model

The parameters described above are related to each other and to empirical measures in a system of mutual constraint. For example, increased accuracy might require increased encoding time. The model that captures this system is formalized in the Appendix; here we derive two predictions from it, one empirical and one theoretical.

Errors Increase Within a Run

One possible interpretation of maintenance cost is that it reflects increasingly careful processing across trial positions. People might be shifting their speed-accuracy criterion gradually toward accuracy. Such a shift would imply a constant or decreasing error rate across trials.

In contrast, SASM predicts that error rates will increase within a run for two reasons. First, over the first few trials of a run the activation curves for I_C and I_P approach each other (Figure 2). This decreases δ which increases memory errors and, hence, performance errors. Second, errors early in a run beget more errors later in that run. An error occurs when I_P is used in place of I_C. This use causes I_P to gain in base-level activation at the expense of I_C. Thus, an error decreases δ for all subsequent trials in that run.

Error data from the experiment described earlier are shown in Figure 4. The ordinate shows total errors out of 176 trials and the abscissa shows trial position. Rather than a constant or decreasing error rate, errors increase throughout the run, as SASM predicts. The effect of trial position is significant, F (6, 114) = 4.9, $p < .001$, as is the linear trend, F (1, 114) = 24.2, $p < .001$.

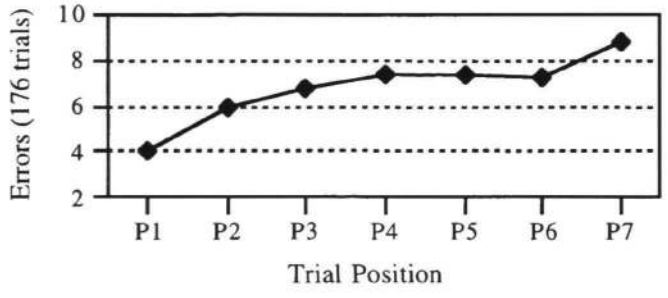

Figure 4: Errors error on post-instruction trials, out of 176 trials. Error maintenance cost spans P1 to P7.

This correct prediction strongly supports our model. We assumed that each instruction is encoded distinctly in episodic memory and decays gradually rather than instantaneously when superseded. These assumptions, together with ACT-R's memory theory, imply the decay trajectory in Figure 2, which in turn predicts the observed error pattern.

Base-Level Activation and Priming are Irreducible

Under the rational memory assumption, activation is composed of two terms: base-level activation (from past use) and priming (from the current cognitive context). This decomposition is based on a Bayesian characterization of the statistical structure of the environment. However, to our knowledge, there has been no analysis of whether both components of activation are functionally necessary. Is it possible that one can be reduced to the other?

By binding the model's parameters we can determine what combinations of base-level and associative activation yield a given error rate. The first parameter, noise, has been estimated many times and typically falls in a limited range (0.30 to 0.85 in terms of the logistic parameter s; Anderson & Lebiére, 1998, p. 217). Hence, although noise is not fixed, it is constrained enough for a sensitivity analysis. As a measure of encoding time we take mean instruction response time, which in our data is 0.97 sec. Run length in our paradigm is 10 trials on average. Finally, the dependent variable, error rate, is 0.023 on trial position P1.

The remaining parameter is priming. Because base-level activation and priming are the only two sources of activation, we can express one as a combination of the other and the δ needed for a given level of accuracy. We designate P_B as the proportion of δ due to base-level activation. P_B is thus nominally defined on an interval of 0 (δ entirely due to priming) to 1 (δ entirely due to base-level activation).

Figure 5 shows a sensitivity analysis of SASM for noise and priming, the two parameters constrained by boundary values. The abscissa shows P_B and the curves predict instruction response time for boundary values of the noise parameter s. Thus the predicted times are those required to achieve δ for given values of P_B and s.

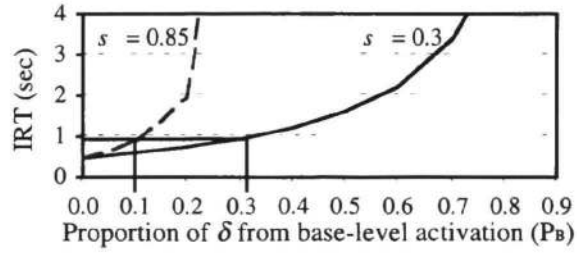

Figure 5: Instruction response time (IRT) as predicted by P_B and upper and lower bounds on activation noise s (see Appendix, Eqn. 4). The empirical IRT of 0.97 is predicted by $P_B = 0.1$ for $s = 0.85$ and $P_B = 0.32$ for $s = 0.30$.

Figure 5 shows that neither base-level activation nor priming alone can achieve the δ needed for high-accuracy serial attention. A P_B of zero predicts a minimum encoding time of roughly 500 msec (regardless of s). Even if δ is entirely due to priming, this amount of initial encoding is

needed to bring the initial base-level activation of I_C up to the final base-level activation of I_P.

For large values of P_B, predicted instruction response times go off the scale. That is, even large amounts of initial encoding cannot completely replace priming. As encoding time increases, δ due to base-level activation approaches a limit (see Appendix, Eqn. 3). Performance accuracy requires δ to be above this limit, so priming must supply the balance of the needed activation. In sum, high-accuracy serial attention depends on both base-level activation and priming.

If some amount of δ must come from priming, what cues could provide this? The environment offers no effective cues; the classification-trial stimulus (a number) equally primes both kinds of instruction (OddEven and HighLow) and thus fails to contribute to δ. Therefore, any effective cues must be internal. We proposed earlier that residual memory for the previous trial is a likely cue. Indeed, it is unclear what other internal cues there could be that affect I_C and I_P differentially. Thus, SASM predicts an inherent inertia to serial-attention performance. Priming by the previous trial implies an architectural propensity to do the same task over again. However, because priming alone cannot produce the needed δ, perseveration is not a danger.

Upper and lower bounds on P_B can be estimated from our data. Figure 5 shows that P_B between 0.1 and 0.32 predicts the empirical instruction response time of 0.97 sec. We interpret this to mean that inertia from trial-to-trial priming substantially facilitates serial attention.

In sum, serial attention depends on both base-level and associative activation — neither is reducible to the other. The general implication is that memory retrieval relies not on one but on two sources of information about the target. This is an axiom of Bayesian analyses in general and ACT-R in particular, but SASM suggests that two sources of information are *required* for reliable retrieval in a dynamic environment. To the extent that serial attention is a building block of higher-level cognition, base-level and associative activation are building blocks as well and earn their designation as atomic components of thought.

Discussion

The SASM model reduces serial attention to a set of memory phenomena, going some way toward banishing the homunculous of mental attention. Switching attention involves rapidly strengthening a new item to be temporarily dominant, and maintaining attention involves letting the current item decay slightly to prevent it from intruding later.

The signature evidence for the model is maintenance cost, as measured by the gradual increase in response times across trials in a run. Although to our knowledge this effect is novel, we have replicated it under a variety of situations (Altmann & Gray, 1999). We explain this effect in terms of short-term, trial-by-trial decay of instructions encoded in episodic memory. This decay is a feature, not a bug, in that it contributes to δ, the activation difference that makes the current instruction always the most active.

The complement to maintenance cost is switch cost. This is typically measured on the first classification trial governed by the new task (Allport, Styles, & Hsieh, 1994; Garavan, 1998; Gopher et al., 1996; Rogers & Monsell, 1995). Switch cost is often (but not always; Allport et al., 1994) interpreted as the cost of moving an attentional spotlight from one location to another, with no related account of maintenance cost. SASM suggests a broader interpretation of switch cost as the cost of processes that increase δ. These processes may occur on classification trials, instruction trials, or elsewhere. On this view, the largest switch cost in our paradigm is instruction response time. On instruction trials, participants strategically encode the displayed instruction to decay through use.

From an applied perspective, SASM may help operationalize cognitive workload in real-time dynamic tasks. Excess workload in a dynamic environment means essentially that too much is happening too fast, causing the operator's performance to suffer. SASM provides a way to analyze such excess workload as a memory phenomenon. For example, in modeling sustained operations, fatigue may be instantiated as increased noise in memory, and accuracy and run length may map directly to measures of task performance. Thus SASM might be used, for example, to predict need for external memory aids as a function of task complexity and opacity (Brehmer & Dorner, 1993).

An open question concerns the first trial position of a run (P1). This position appears not to benefit from high initial instruction strength, in that RT is substantially slower than on later trial positions (Figure 1). One possible explanation is that this slowdown reflects a final phase of the encoding process. If initial encoding stops when activation reaches criterion, then transient noise may boost activation to this criterion prematurely and bring an early end to the instruction trial. However, if the instruction persists in iconic memory, a final encoding phase would be possible as P1 begins. In our computational ACT-R model, this final encoding phase regresses activation toward its criterion value, because instruction chunks not made active enough during the instruction trial get a second chance. In future research it will be important to investigate empirically the extra processing that seems to take place on P1, and the role of this processing in the general scheme of serial attention.

Acknowledgments

This work was supported by a grant from the Air Force Office of Scientific Research (F49620-97-1-0353) to Wayne D. Gray. We thank Wai-Tat Fu, Michael J. Schoelles, Brian D. Ehret, Willard Larkin, and an anonymous reviewer for their comments.

References

Allport, A., Styles, E. A., & Hsieh, S. (1994). Shifting intentional set: Exploring the dynamic control of tasks. In C. Umilta & M. Moscovitch (Eds.), *Attention and performance IV*. Cambridge, MA: MIT Press. 421-452.

Altmann, E. M., & Gray, W. D. (1998). Pervasive episodic memory: Evidence from a control-of-attention paradigm. In M. A. Gernsbacher & S. J. Derry (Eds.), *Proceedings of the twentieth annual conference of the Cognitive Science Society* (pp. 42-47). Hillsdale, NJ: Erlbaum.

Altmann, E. M., & Gray, W. D. (1999). The within-run slowing effect in serial attention. *Manuscript in preparation.*

Anderson, J. R. (1990). *The adaptive character of thought.* Hillsdale, NJ: Erlbaum

Anderson, J. R., & Lebiére, C. (1998). *The atomic components of thought.* Hillsdale, NJ: Erlbaum.

Brehmer, B. & Dorner, D. (1993). Experiments with computer-simulated microworlds: Escaping both the narrow straits of the laboratory and the deep blue sea of the field study. *Computers in Human Behavior*, **9**, 171-184.

Garavan, H. (1998). Serial attention within working memory. *Memory and Cognition*, **26**, 263-276.

Gopher, D., Greenshpan, Y., & Armony, L. (1996). Switching attention between tasks: Exploration of the components of executive control and their development with training. *Proc. HFES 40th Annual Meeting.* Philadelphia: HFES. 1060-1064.

Posner, M. I. (1980). Orienting of attention. *Quarterly Journal of Experimental Psychology*, **32**, 3-25.

Rogers, R. D., & Monsell, S. (1995). Costs of a predictable switch between simple cognitive tasks. *Journal of Experimental Psychology: General*, **124**, 207-231.

Appendix

SASM is predicated on the notion that, to be reliably retrieved, the current item (e.g., an instruction) must be more active than the previous item by some amount δ:

$$B_C = B_P + \delta \qquad \textbf{Serial Attention Equation} \qquad (1)$$

δ depends on the probability of retrieving the current instruction, $P(I_C)$. This is given by the Chunk Choice Equation (Anderson & Lebiére, 1998, p. 77):

$$P(i) = e^{m_i/t} / [\sum_j e^{m_j/t}]^{-1}$$

$P(i)$ is the probability of retrieving chunk i given noise t and given j chunks in memory each with expected activation m_j. We assume an infinite number of previous instructions of which the most recent few materially affect the probability of a previous instruction intruding on the current one. The expected activations of these few previous instructions are roughly δ apart, so we estimate $m_j \approx m_i - j\delta$. This allows for a closed-form solution to the Chunk Choice Equation, $P(I_C) = 1 - e^{-\delta/t}$. Rearranging, we get $\delta = -t \ln[1 - P(I_C)]$.

$P(I_C)$ is also constrained by performance accuracy, A. We assume an accurate response if the retrieved instruction is (a) I_C, which always specifies the appropriate task; (b) a previous instruction I_{PA} that specifies the appropriate task; or (c) a previous instruction I_{PN} that specifies the not-appropriate task but whose response for the current stimulus is the same as that of the appropriate task. Thus, $A = P(I_C) + P(I_{PA}) + P(I_{PN})$. The paradigm is structured such that $P(I_{PA}) = 2P(I_{PN})$ and, assuming that an instruction is always retrieved when retreival is attempted, $P(I_C) = 1 - [P(I_{PA}) + 2P(I_{PN})]$. These constraints together imply that $P(I_C) = 4A - 3$. Substituting this for $P(I_C)$ in the equation for δ, and using the error rate $E = 1 - A$ instead of A, yields the equation below, where P_B, defined on [0...1], limits δ to the proportion due to base-level activation.

$$\delta = -tP_B \ln(4E) \qquad (2)$$

With δ bound by E, we can find how many initial uses are needed to achieve that δ. Assuming one use per trial, average run length R, and N initial uses, we can express the age of an instruction in terms of uses. At $P1_C$, the age of I_P is one instruction trial and R classification trials from the previous run, plus another instruction trial and one classification trial in the current run, or R+3. The age of I_C is only 2. The number of uses of I_P is N+R and of I_C is N+1. We can now expand Eqn. 1 using the Base-Level Learning Equation (Anderson & Lebiére, 1998, p. 124), which defaults to $B = \ln(2nT^{-0.5})$ for n uses over chunk age T. Eqn. 1 in terms of N, R, and δ (and exponentiated) is then:

$$(N+1)2^{-0.5} = (N+R)(R+3)^{-0.5} e^{\delta} \qquad (3)$$

To estimate the maximum contribution of initial use to δ, we can solve Eqn. 3 for δ and take the limit as N goes to infinity. This produces $\ln\sqrt{0.5(R+3)}$, or 0.94 for R=10. However, E with minimal noise entails a δ of at least 1.02 (via Eqn. 2 with P_B=1; E and t are bound below). Therefore, some δ must come from differential priming of I_C over I_P.

To estimate encoding time as a function of P_B, we can solve Eqn. 3 for N and substitute for δ from Eqn. 2. Two ACT-R production firings serve to encode a chunk once, one to create the chunk and push it onto the goal stack and one to pop it into memory. Default firing time is 50 msec, so time per use is 100 msec. Thus predicted encoding time, as measured by instruction response time (IRT), is:

$$IRT = 100\frac{\sqrt{R+3} - R\sqrt{2}(4E)^{-tP_B}}{\sqrt{2}(4E)^{-tP_B} - \sqrt{R+3}}, \text{ with } t = \sqrt{2}s \qquad (4)$$

Eqn. 4, with E=0.023 (bound empirically), s=0.30 and 0.85 (Anderson & Lebiére, 1998, ch. 7), R=10 (task-specific), and P_B = [0...1], produces the curves in Figure 5.