Exploring Multitasking Strategies in an ACT-R model of a Complex Piloting Task

Garrett Swan (garrett.swan@cubic.com)
Cubic Defense Applications, Inc
San Diego, CA 92123 USA

Christopher A. Stevens (christopher.stevens.28@us.af.mil)
Air Force Research Laboratory
Wright Patterson AFB, OH USA

Christopher R. Fisher (christopher.fisher.27.ctr@us.af.mil)
Cubic Defense Applications, Inc
Beavercreek, OH 45324 USA

Samantha Klosterman (samantha.klosterman@ballaerospace.com)
Ball Aerospace
Fairborn, OH 45324

Abstract

Multitasking is a challenging cognitive task, and there are many factors driving which strategy participants use to complete tasks concurrently. We utilized a model comparison approach to evaluate how participants decide which task to switch to next using the Air Force Multiple Attribute Battery (AF-MATB). We used the cognitive architecture, Adaptive Control of Thought – Rational (ACT-R), to simulate multitasking in the AF-MATB. We varied how the model decided which task to attend to next by comparing a purely top-down strategy, a purely reactive, bottom-up selection strategy, and mixtures of the two. We compared simulations of the model to data from Bowers, Christensen, and Eggemeier (2014). The best combination involved a mixture of top-down and bottom-up selection. Neither the purely top-down nor bottom-up selection models performed well. These results suggest that participants use a complex mixture of strategies to multitasking. The use of a top-down strategy suggests participants could develop efficient strategies to multitask successfully, and that participants may be using a more effortful serial search in the AF-MATB, as indicated by the model's serial processing implementation.

Keywords: ACT-R; AF-MATB; multitasking; cognitive architecture

Introduction

In our daily and professional lives, we often perform multiple concurrent tasks, such as eating while driving, listening to a coworker while reading an email, or piloting aircraft while monitoring numerous instruments. How individuals are able to multitask is an old and ongoing question in research because there is a vast space of human and environmental factors that impact one's ability to multitask (Meyer & Kieras, 1997; Koch, Poljac, Müller, & Kiesel, 2018; Fischer & Plessow, 2015). Multitasking is interesting from a theoretical perspective because it requires multiple cognitive systems to work together in service of a common goal and to adapt to changing circumstances. Moreover, the space of strategies one could use to accomplish multitasking can be quite large (Salvucci, Taatgen, & Borst, 2009; Smith et al., 2008).

There are different aspects of multitasking where strategies may manifest. For example, many studies have examined individuals' decision strategies to stop one task and switch to another, which could be serial without interruptions, when a sufficient amount of time has passed (Kushleyeva, Salvucci, & Lee, 2005), or when there are diminishing benefits of the currently attended task (Payne, Duggan, & Neth, 2007). Here, we are interested in the strategy that determines which task to switch to next. Individuals may search by top-down factors, such as serially moving attention from task to task, "urgency" (Salvucci, Kushleyeva, & Lee, 2004), or activation (Altmann & Trafton, 2002), or by bottom-up factors, such as selective attention (Patsenko & Altmann, 2010). The continuum from a purely top-down strategy to a purely bottom-up selection strategy represents one slice of the problem space of how individuals decide where to allocate attention next. Determining which strategies participants use has theoretical and practical implications in training, the design of realistic simulations of human behavior, and in the development of instruments that could facilitate multitasking.

We used the Adaptive Control of Thought – Rational (ACT-R) cognitive architecture (Anderson et al., 2004) to examine multitasking strategies. One of the primary benefits of using a cognitive architecture such as ACT-R is that it provides a formal framework for developing and testing strategy use in multitasking. We simulated the Air Force Multi-Attribute Task Battery (AF-MATB, Miller 2010), which is a commonly used multitasking environment that has been used to explore different aspects of multitasking, such as the hysteresis effect (Bowers et al., 2014; Kim, House, Yun, & Nam, 2019) and the relationship between performance and physiological measures (Splawn & Miller, 2013). We compared a continuum of models ranging from a purely serial top-down strategy to a purely ballistic strategy driven by bottom-up attention to a combination of the two. When comparing our simulation with behavioral data (Bowers et al., 2014), we found that the best fitting models used a mixture of top-down and bottom-up strategies.

Methods

Participants

We tested our model against behavioral data from Bowers et al. (2014). Sixteen participants (11 male, 5 female, ages 18 to 28) from neighboring universities (Air Force Institute of Technology, Wright State University, University of Dayton, and Wright Site Junior Force Council) participated in the study. Participants were unfamiliar with the task and completed informed consent prior to participation. The study was approved by Air Force Research Laboratory Institutional Review Board.

AF-MATB Task Description

The AF-MATB is a laboratory environment designed to investigate multitasking behavior in tasks similar to some of those encountered while operating aircraft. Full details regarding the AF-MATB can be found in Miller et al. (2010; 2014). Participants monitored subtasks for scripted events and responded to those events with keyboard presses and a joystick. In Bowers et al. (2014), the subtasks included System Monitoring, Tracking, Communications, and Resource Management. In the System Monitoring subtasks, participants had a limited time (3 and 6 seconds) to press a key on the keyboard when a Light (color change) or Gauge (exceeding a y-axis threshold) malfunctioned, respectively. In the Tracking subtask, participants used a joystick to adjust the position of a randomly moving reticle. In the Communications subtask, participants listened for audio files, adjusted and submitted the frequency and channel if the audio matched the participant's callsign. In the Resource Management subtask, participants monitored fluid levels in two tanks and adjusted the state of 8 pumps to control the fluid levels.

Parameters underlying these events and the frequency of these events were controlled by the experimenter. Events were distributed pseudorandomly, such that the same events could not overlap. Events from other subtasks could occur concurrently. Difficulty was primarily determined by increasing the frequency of events, which was the case in Bowers et al. (2014), resulting in greater overlap between events for the Hard difficulty compared to the Easy difficulty.

ACT-R Model

The ACT-R architecture consists of discrete modules (e.g., visual, auditory) that are acted upon by production rules (if-then statements) that control behavior. Cognition manifests as information flows between the different modules.

Our model was designed to detect and respond to events in the AF-MATB task environment. The model interacted with a custom built version of the AF-MATB in Python, which had reduced visual fidelity but the same timing and visual properties as the AF-MATB. We designed the simplest model that was similar to human behavior, given that a more complex model designed specifically to fit the data would theoretically be less generalizable. The structure described below is the core version of the model used in all of the simulations.

Core Model The core model selected subtasks in a strictly serial (i.e., top-down) manner. The model responded to the subtasks primarily through ACT-R productions, given that participants generally receive training in the AF-MATB prior to participation (i.e., participants completed six training sessions at 2 hours each in Bowers et al. 2014), which suggests that the rules for detecting and responding were well-learned and practiced. See Figure 1 for a high-level overview of how the model works.

First, the model turned on pumps (in this case, pumps 1 to 6) in the Resource Management subtask, which every participant did in Bowers et al. (2014). Next, the model searched for subtasks serially (in this case, clockwise) using two productions: (1) find a visual-location to attend to and (2) move visual attention to that location. This serial search was the

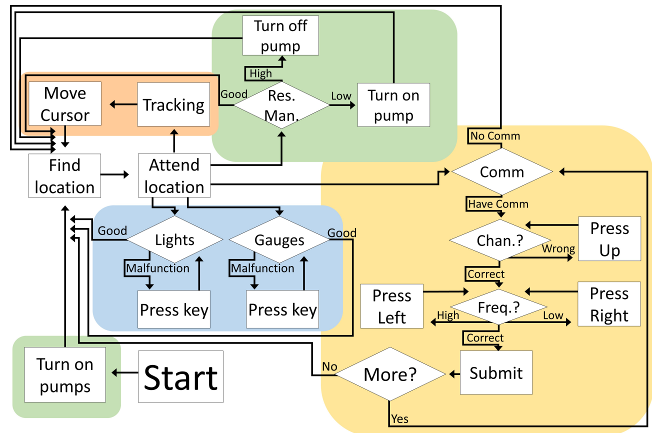

Figure 1: A high-level diagram of how the ACT-R model interacts with the AF-MATB task. The colored boxes correspond to productions for the subtasks. Green = Resource Management (Res. Man.), Yellow = Communications (Comm), Orange = Tracking, Blue = System Monitoring, Boxes = processes, Diamonds = decision points.

main loop that brought the model's attention to each subtask.

If the model was attending to one of the Lights or Gauges in the System Monitoring subtask and there was a malfunction, then the model responded by pressing the appropriate key on the keyboard with the left hand index finger.

If the model attended the Tracking subtask reticle, then the model moved the cursor towards the reticle, with the cursor simulating the behavior of a joystick by adding a constant x and y value in the direction of the center of the tracking panel to the reticle. The model's right hand was kept on the mouse.

If the model attended to one of the tank levels in the Resource Management subtask and the tank level was either too high or too low (i.e. 100 L), then the model cycled attention through the pumps with the intention of turning on pumps if the level was too low or turning off pumps if the level was too high. If too high, then the model checked and turned off pump 2 or 4 to slowly decrease the tank level. If too low, then the model checked and turned on pumps 1, 5, then 2 or 3, 6, then 4 to increase the level.

While attending to the above subtasks, the model listened for audio. If audio was played and started with the correct callsign, then the model stored the upcoming channel and frequency information in declarative memory. The next model production attempted to retrieve the channel and frequency from declarative memory. Successful retrieval of this memory switched the model's attention towards the Communications subtask and moved the model's left hand to the left-arrow key such that the model could reach the relevant keys. If necessary, the model first adjusted the channel using the up arrow key. Then, if necessary, the model adjusted the frequency using the left and right arrow keys. Once the frequency was correct, the model pressed the return key to submit the response and moved the left hand back to the 5 key to be able to reach keys for the System Monitoring and Resource Management subtasks.

Errors manifested in a few different ways. In the System Monitoring and Resource Management subtasks, errors could occur in two ways: (1) when the model initiated the production to press the key as the malfunction returned to normal automatically (3 seconds for Lights and 6 seconds for Gauges) and (2) there was motor noise (see below) such that the model could press the wrong key. If the model pressed the incorrect key, then the attended subtask would still be in a malfunctioned state and the model would attempt to press the correct key again. In the Tracking subtask, we assumed there was noise in the motor movements, which was modeled by activating the ACT-R cursor-noise parameter. In the Communication subtask, the model could fail to retrieve the stored chunk corresponding to the channel and frequency if the declarative memory was not sufficiently activated during retrieval.

Model Parameters

The majority of the ACT-R model parameters were kept at their default level. We enabled subsymbolic (:esc = t) and full base level learning computations (:ol = nil). Given that malfunctions maintained in their state until corrected, we set the visual-onset-span parameter to 3.0 seconds, which represented the model being able to detect the malfunction after it had occurred in the model's peripheral vision.

For the Communications subtask, the model stored and retrieved from declarative memory. We set base-level learning (bll) to the recommended level (0.5). To achieve a retrieval rate that was approximately the same average as the behavioral data in Bowers et al. (2014) (approximately 92%), we altered the activation noise (ans = 0.2), base-level constant (blc = 2), and retrieval threshold (rt = 2.9) based on a grid-based search of a partial dataset from one of the participants in Bowers et al. (2014).

In simulating the Tracking subtask, we activated the incremental-mouse-moves parameter to more realistically capture joystick behavior. In addition, we activated the cursor-noise parameter to add motor movement noise. In simulating the joystick, the mouse cursor position in Cartesian coordinates were converted into Polar coordinates. The radius was multiplied by 0.125, then converted back into Cartesian coordinates. The resulting x and y values were capped at 10 pixels given physical limitations in joystick movements. The x and y were added to the Cartesian coordinates of the reticle each update.

Given the false alarm rate in System Monitoring (4.4% in Bowers et al.) and the percentage of times participants turned on/off pumps that took the fluid level away from the intended direction (9.2%), the model randomly pressed a key (F1-F6 and 1-8) on 5.0% of responses when responding to System Monitoring and Resource Management subtasks.

Strategy Space

Here, we introduce reactive, event-driven strategies driven by bottom-up selection. In these models, the sequential selection process was interrupted if the model noticed a malfunction. That is, instead of the next subtask in the sequence determined by clockwise position, the model would attend to a different subtask. Each variant allowed interruption from a specific subset of the subtasks. Finally, we included a purely reactive variant of the model that only responded to tasks when malfunctions were passively noticed.

The different variants were referred to by which subtask had bottom-up selection (L = Lights and G = Gauges in System Monitoring, R = Resource, T = Tracking, N/A = for no bottom-up selection, and Only LGRT for only bottom-up selection and no top-down strategy)[1]. In total, there were 17 different variants (N/A, L, G, R, T, LG, LR, LT, GR, GT, RT, LGR, LGT, LRT, GRT, LGRT, and Only LGRT).

Bottom-up Selection Simulation We simulated bottom-up selection by using an ACT-R feature called "buffer stuffing", which is when the visual-location buffer is automatically populated with the location of a new stimulus instead of the model needing to search for a stimulus to add to the visual-location buffer (i.e., skipping the "find location" production in Figure 1). Note that following bottom-up selection, the next subtask would continue clockwise from the currently attended subtask and not from the previous subtask.

For the System Monitoring subtasks, "buffer stuffing" occurred any time there was a malfunction. For the Tracking subtask, "buffer stuffing" occurred when the tracking object was 27.5 pixels away from the origin. For the Resource Management subtask, "buffer stuffing" occurred when the tank levels were 700 L above or below the middle of the tank. These thresholds were tested on partial datasets to ensure they improved performance for that subtask.

Overall, as intended, each subtask had improved performance when that subtask had bottom-up selection (Table 1), indicating that the bottom-up selection was prioritizing that subtask.

Table 1: Average measures with and without bottom-up selection when that subtask has bottom-up selection active for all of the trials. Sys. Mon. = System Monitoring

Subtask	With	Without
Sys. Mon. Accuracy	0.76	0.66
Sys. Mon. RT	1.76	2.51
Tracking	36.06	46.73
Resource	181.74	230.73

Performance Measures

Trial Simulation We simulated the same number of participants (n = 16) and trials (t = 12) in Bowers et al. (2014). We used the same event lists generated from the participants, given that the event numbers differed between participants in

[1]There was no bottom-up selection for the Communication subtask since the model already switches to that task as soon as it can

the Hard difficulty. The transitions from Easy to Hard (6 trials) and Hard to Easy (6 trials) were counterbalanced. Two of the participants did not fully complete a trial, so the total number of simulations was 190 trials for each of the 17 different variants.

Performance Evaluation We evaluated the model variants by calculating accuracy (accuracy = correct / total) and reaction time for correct responses in the System Monitoring and Communications subtasks. For the Tracking subtask, we averaged the Euclidean distance of the reticle to center across the trial. For the Resource Management subtask, we averaged the deviations of both tank levels across the trial.

In our model comparison to select the best model, we accounted for the difference in scales in the dependent measures from each subtask, n, by computing a mean normalized root mean square errors (NRMSE):

$$\mathrm{NRMSE} = \frac{1}{n}\sum_{i=1}^{n}\frac{\sqrt{(\hat{x}_i - \bar{x}_i)^2}}{\bar{x}_i}$$

with $\hat{x}$ being the mean predicted measure, $\bar{x}$ being the mean observed DV, and i being the index for subtask. Given the division of the RMSE by the mean from the behavioral data, values closer to 0 indicate less error between the model and behavioral data.

After selecting the best model using NRMSE, we conducted a Bayesian repeated measures ANOVA using JASP (JASP Team, 2022) to analyze the effect of difficulty (Easy vs Hard) for the model and the behavioral data (Bx) to determine if the model is qualitatively showing the same effect of difficulty as the participants. Then, we conducted a Bayesian mixed factors ANOVA with difficulty (repeated: Easy vs. Hard) and group (Model vs. Bx) to see if the effect of difficulty was the same for the model and behavioral data.

Results

We evaluated the model's performance for the dependent measures. In the figures below, the ordering of the model variants is based on ascending NRMSE. Based on NRMSE, the best models included bottom-up selection for one of the System Monitoring subtasks (Light or Gauge) and Resource Management subtasks. Specifically, the LR (NRMSE: 0.054) and R (NRMSE: 0.11), and GR (NRMSE: 0.14) models performed best.

The purely top-down model (N/A, NRMSE: 0.17) performed better than the model with solely bottom-up selection (Only LGRT, NRMSE: 0.37). This difference was largely driven by the very poor performance in the Resource Management subtask for the Only LGRT model, which was likely because the interruptions in the other subtasks took attention away from the Resource Management subtask.

There was a wide range of System Monitoring performance (see Figure 2). As indicated above, including either Light or Gauge bottom-up selection increased System Monitoring subtask accuracy and decreased reaction time. The LG (NRMSE: 0.33) variant had the closest System Monitoring accuracy to the behavioral data, which was expected given that the model attended more frequently to the System Monitoring subtask, but was a poor fit otherwise.

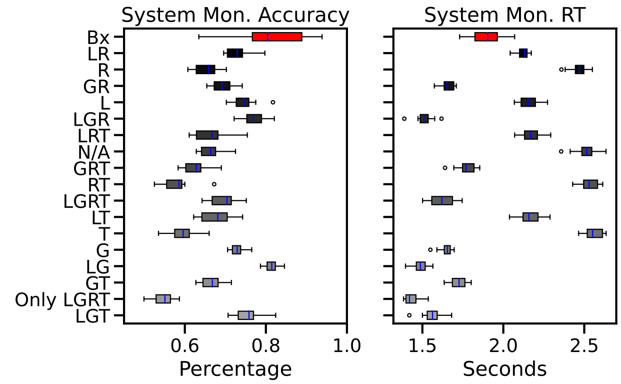

Figure 2: Simulation of the System Monitoring subtask for the different models (y-axis) and behavioral data (Bx, in red). System Mon. = System Monitoring

The Communications subtask simulation had the least amount of variability (see Figure 3). This was expected given there is no bottom-up selection to affect performance and once in the Commmunication subtask, the model was not interrupted by other subtasks. Some of the variance in the behavioral data suggests that some participants may have interleaved this subtask with other subtasks.

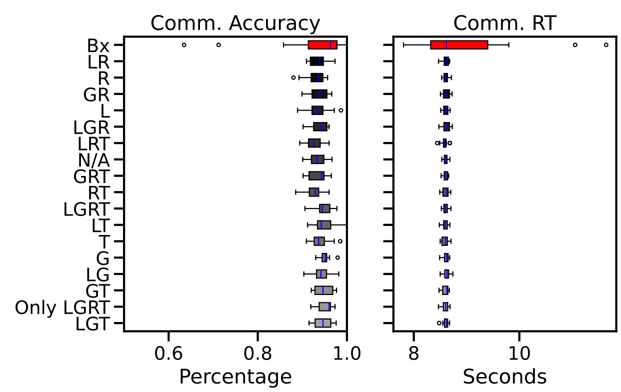

Figure 3: Simulation of the Communications subtask for the different models (y-axis) and behavioral data (Bx, in red). Comm. = Communications

Tracking performance tended to be precise (see Figure 4 Left), even without bottom-up selection. If there was bottom-up selection in the Resource Management but not Tracking subtask, then the Tracking performance was significantly less precise but closer to the behavioral data. This likely occurred because of the clockwise search pattern of the model, which could effectively skip the Tracking subtask if the model attended to the Resource Management subtask.

The model tended to be imprecise in the Resource Man-

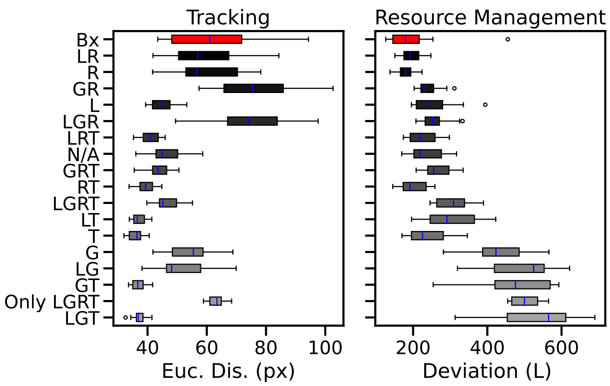

Figure 4: Simulation of the Tracking (Left) and Resource Management (Right) subtasks for the different models (y-axis) and behavioral data (Bx, in red). Px = Pixels

agement subtask (see Figure 4 Right). This may be due to the specific strategy of the current model. For example, participants tended to change the status of the pumps on average 53.7 times (sd = 12.3), whereas the model ranged from 40.7 (LR, best fit) to 27 (LGT, worst fit) changes.

The Effect of Task Difficulty

We evaluated the best fitting model's (LR) ability to show the task difficulty effect found in Bowers et al. (2014). In Table 2, we show descriptive results for the effect of difficulty and the Bayes Factor when difficulty was added to the null model. There was decisive evidence for the inclusion of difficulty in the model and behavioral comparisons (BF > 421.6) except for the following: for Communications subtask accuracy, there was moderate evidence for not including difficulty for the model (BF = 0.436) and moderate evidence for including difficulty for the behavioral data (BF = 2.6), and there was anecdotal evidence for including difficulty for Resource Management subtask (BF = 5.2). These results overall show the model captured the qualitative worsening of performance when the task became more difficult.

When quantitatively comparing the model to the behavioral data and the effect of difficulty using Bayesian mixed factor ANOVAs, there was decisive evidence for the effect of difficulty (BF $> 1.1e+4$) except for Communications subtask accuracy (BF = 0.43), which was expected given the lack of a difference in Table 2. Evidence was leaning towards the null hypothesis for the effect of group (BF < 0.69) except for anecdotal and decisive evidence for the effect of group in System Monitoring subtask accuracy (BF = 1.9) and reaction time (BF =$5.1e+2$), respectively. This was expected given the minimal error between the best fitting model and the behavioral data. Interestingly, there tended to be decisive evidence for an interaction term (BF > 231.9) except for Communication subtask reaction time (BF = 58.1) and accuracy (BF = 2.4), which had very strong and anecdotal evidence respectively. These results suggest that the predicted magnitude of the effect of difficulty differed from that of the participants. Visual inspection of the table suggests that the model tended to be more accurate and faster on the Easy difficulty and less accurate and slower on the Hard difficulty.

Table 2: Average values for Easy and Hard difficulty for the model (M) and behavioral data (Bx). The Bayes Factor (BF) for including difficulty in a Bayesian repeated measures ANOVA is also shown, with higher values indicating evidence for the difficulty factor.

Subtask	Easy(m, sd)	Hard(m, sd)	BF
M: SysMon. Acc.	0.96(0.01)	0.68(0.03)	2.7e23
Bx: SysMon. Acc.	0.93(0.06)	0.79(0.1)	4.9e5
M: SysMon. RT	1.63(0.05)	2.28(0.05)	2.4e23
Bx: SysMon. RT	1.57(0.2)	1.98(0.1)	1.3e10
M: Comm. Acc.	0.92(0.06)	0.94(0.02)	0.463
Bx: Comm. Acc.	0.97(0.1)	0.92(0.1)	2.6
M: Comm. RT	8.4(0.11)	8.7(0.07)	5.4e6
Bx: Comm. RT	8.32(1.0)	9.21(1.2)	421.6
M: Tracking	26.1(2.3)	93.2(23.7)	2.1e10
Bx: Tracking	42.4(13)	81.8(17.8)	2.1e9
M: ResMan	85.5(4.9)	306(53.8)	3.6e14
Bx: ResMan	173.1(72)	221(97.0)	5.2

General Discussion

Multitasking has captured the interest of researchers because it provides a rich environment for understanding the strategies people use to manage and prioritize multiple competing goals. We contributed to the understanding of strategy use in multitasking by comparing a continuum of strategies ranging from purely top-down (i.e. selecting tasks in a fixed order) to purely bottom-up (i.e., only selecting tasks that malfunctioned or changed). These strategies were instantiated in the ACT-R cognitive architecture in order to test their predictions quantitatively. Overall, we found that the best fitting model was neither using a strict top-down nor bottom-up selection strategy. Instead, the best model used a mixture of the two. That is, the model serially searched, but the serial search could be interrupted if a malfunction was detected in the model's peripheral vision.

Our simulation suggests a few things. First, these results indicate that task switching in the AF-MATB was largely driven by top-down strategies. While the best fitting model had bottom-up selection for two of the subtasks, the majority of the models we simulated with top-down strategies performed adequately in capturing the behavioral data. These findings corroborate other findings in the literature that suggest top-down factors, such as task instruction (Lehle & Hübner, 2009), affect and alter task performance, which highlights the needs for a better understanding of strategy use in multitasking. A better understanding of how strategies affect performance could result in the development of strategies and instruments that improve multitasking performance.

Second, it suggests that a model in which tasks are completed in a serial fashion provides a satisfactory account of multitasking in the AF-MATB. This is consistent with prior work in which individuals opt for an effortful, serial strategy instead of a parallel approach to improve performance on some tasks (Lehle, Steinhauser, & Hübner, 2009). It is possible that an account in which task processing overlaps to a greater degree (e.g. Salvucci and Taatgen, 2008) could also provide a satisfactory account. Further research is needed to determine the extent to which concurrent processing is needed to account for performance in the AF-MATB and whether the AF-MATB is sensitive enough to distinguish these accounts.

There were limitations. First, the magnitude of the effect of difficulty was significantly different from the behavioral data. One explanation is that participants may be switching their strategy when the task became more difficult, such as using a bottom-up selection strategy when the task was difficult and a top-down strategy when the task was easy. Second, strategies likely differ between participant. It may be that some participants only serially searched while others only used bottom-up selection. There is likely not enough data in Bowers et al. (2014) to determine if this is the case or not.

Acknowledgments

The opinions expressed herein are solely those of the authors and do not necessarily represent the opinions of the United States Government, the U.S. Department of Defense, the U.S. Air Force, or any of their subsidiaries, or employees. The contents have been reviewed and deemed Distribution A. Approved for public release. Case number: AFRL-2022-1713.

References

Altmann, E. M., & Trafton, J. G. (2002). Memory for goals: An activation-based model. *Cognitive science*, *26*(1), 39–83.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological review*, *111*(4), 1036.

Bowers, M. A., Christensen, J. C., & Eggemeier, F. T. (2014). The effects of workload transitions in a multitasking environment. In *Proceedings of the human factors and ergonomics society annual meeting* (Vol. 58, pp. 220–224).

Fischer, R., & Plessow, F. (2015). Efficient multitasking: parallel versus serial processing of multiple tasks. *Frontiers in psychology*, *6*, 1366.

JASP Team. (2022). *JASP (Version 0.16.1)[Computer software].* Retrieved from `https://jasp-stats.org/`

Kim, N. Y., House, R., Yun, M. H., & Nam, C. S. (2019). Neural correlates of workload transition in multitasking: An act-r model of hysteresis effect. *Frontiers in Human Neuroscience*, *12*, 535.

Koch, I., Poljac, E., Müller, H., & Kiesel, A. (2018). Cognitive structure, flexibility, and plasticity in human multitasking—an integrative review of dual-task and task-switching research. *Psychological bulletin*, *144*(6), 557.

Kushleyeva, Y., Salvucci, D. D., & Lee, F. J. (2005). Deciding when to switch tasks in time-critical multitasking. *Cognitive systems research*, *6*(1), 41–49.

Lehle, C., & Hübner, R. (2009). Strategic capacity sharing between two tasks: Evidence from tasks with the same and with different task sets. *Psychological Research PRPF*, *73*(5), 707–726.

Lehle, C., Steinhauser, M., & Hübner, R. (2009). Serial or parallel processing in dual tasks: What is more effortful? *Psychophysiology*, *46*(3), 502–509.

Meyer, D. E., & Kieras, D. E. (1997). A computational theory of executive cognitive processes and multiple-task performance: Part i. basic mechanisms. *Psychological review*, *104*(1), 3.

Miller, W. D. (2010). *The us air force-developed adaptation of the multi-attribute task battery for the assessment of human operator workload and strategic behavior.* (Tech. Rep. No. AFRL-RH-WP- TR-2010-0133). Retrieved from the Defense Technical Information Center website: http://apps.dtic.mil/dtic/tr/fulltext/u2/a537547.pdf.

Miller, W. D., Schmidt, K. D., Estepp, J. R., Bowers, M., & Davis, I. (2014). *An updated version of the us air force multi-attribute task battery (af-matb).* (Tech. Rep. No. AFRL-RH-WP-SR-2014-0001). Retrieved from the Defense Technical Information Center website: https://apps.dtic.mil/sti/pdfs/ADA611870.pdf.

Patsenko, E. G., & Altmann, E. M. (2010). How planful is routine behavior? a selective-attention model of performance in the tower of hanoi. *Journal of Experimental Psychology: General*, *139*(1), 95.

Payne, S. J., Duggan, G. B., & Neth, H. (2007). Discretionary task interleaving: heuristics for time allocation in cognitive foraging. *Journal of Experimental Psychology: General*, *136*(3), 370.

Salvucci, D. D., Kushleyeva, Y., & Lee, F. J. (2004). Toward an act-r general executive for human multitasking. In *Iccm* (pp. 267–272).

Salvucci, D. D., & Taatgen, N. A. (2008). Threaded cognition: an integrated theory of concurrent multitasking. *Psychological review*, *115*(1), 101.

Salvucci, D. D., Taatgen, N. A., & Borst, J. P. (2009). Toward a unified theory of the multitasking continuum: From concurrent performance to task switching, interruption, and resumption. In *Proceedings of the sigchi conference on human factors in computing systems* (pp. 1819–1828).

Smith, M. R., Lewis, R. L., Howes, A., Chu, A., Green, C., & Vera, A. (2008). More than 8,192 ways to skin a cat: Modeling behavior in multidimensional strategy spaces. In *Proceedings of the 30th annual conference of the cognitive science society* (pp. 1441–1446).

Splawn, J. M., & Miller, M. E. (2013). Prediction of perceived workload from task performance and heart rate measures. In *Proceedings of the human factors and ergonomics society annual meeting* (Vol. 57, pp. 778–782).