Optimal Reassembly of Shadow Tests in CAT

Applied Psychological Measurement

2016, Vol. 40(7) 469–485

© The Author(s) 2016

Reprints and permissions:

sagepub.com/journalsPermissions.nav

DOI: 10.1177/0146621616654597

apm.sagepub.com

Seung W. Choi¹, Karin T. Moellering², Jie Li²,
and Wim J. van der Linden³

Abstract

Even in the age of abundant and fast computing resources, concurrency requirements for large-scale online testing programs still put an uninterrupted delivery of computer-adaptive tests at risk. In this study, to increase the concurrency for operational programs that use the shadow-test approach to adaptive testing, we explored various strategies aiming for reducing the number of reassembled shadow tests without compromising the measurement quality. Strategies requiring fixed intervals between reassemblies, a certain minimal change in the interim ability estimate since the last assembly before triggering a reassembly, and a hybrid of the two strategies yielded substantial reductions in the number of reassemblies without degradation in the measurement accuracy. The strategies effectively prevented unnecessary reassemblies due to adapting to the noise in the early test stages. They also highlighted the practicality of the shadow-test approach by minimizing the computational load involved in its use of mixed-integer programming.

Keywords

CAT, shadow-test approach, optimal test design

The shadow-test approach to computer-adaptive testing (CAT) provides a flexible framework for adaptive testing solutions requiring complex sets of constraints (van der Linden, 2005; van der Linden & Dia0, 2014, Chapter 7; van der Linden & Reese, 1998). This “Shadow CAT”—approach conceptualizes adaptive testing as a two-stage optimization process. The first stage involves solving a constrained combinatorial optimization problem to meet all content and other pertinent constraints. The second stage involves applying a standard item selection criterion such as maximum information. The optimization problem in the first stage is computing-intensive and generally requires the use of mixed-integer programming (MIP) solvers (e.g., Chen, Batson, & Dang, 2010). The MIP problem in Shadow CAT is essentially an optimal test assembly problem for fixed forms that is performed in real time and for which any items administered to the examinee earlier in the test serve as additional constraints.

¹McGraw-Hill Education CTB, Monterey, CA, USA

²McGraw-Hill Education, Monterey, CA, USA

³Pacific Metrics, Monterey, CA, USA

Corresponding Author:

Seung W. Choi, McGraw-Hill Education CTB, 20 Ryan Ranch Rd., Monterey, CA 93940, USA.

Email: choi.phd@gmail.com

As with many computing-intensive statistical procedures, Shadow CAT is only viable with modern computing techniques and resources. Only with the advent of technologies allowing for pooling of resources, for example, cloud computing, has computing power become cheap, ubiquitous, and (mostly) scalable without limit—enabling formerly utopic large-scale implementations of optimal test assembly in real time. Moreover, modern MIP solvers have become extremely powerful, solving optimal test assembly problems for large numbers of examinees concurrently. For typical assembly problems (e.g., an item pool of several hundred items and several hundred constraints), these solvers are capable of finding solutions in split seconds. Despite these technological advancements, the concurrency requirements for large numbers of examinees taking a CAT at the same time are still hefty (e.g., 250,000 examinees for consortium-based CAT implementations such as Smarter Balanced). In such a setting, with one server, a 32-item CAT would require about 8 million solves or almost 2 days (45 hr) worth of nonstop solving (at a realistic estimate of 20 ms per solve). At that rate, over 45 servers would be needed to support the computational volume within, approximately, 1 hr of test time.

Consequently, special attention is being paid at present to maximize efficiency through justifiable means. This study proposes a theoretically sound approach to maximizing efficiency under the current conceptualization of Shadow CAT. More specifically, the present study aims for determining an effective reassembly plan for shadow tests during a CAT session without compromising optimality. In the standard form, which is optimal by definition, a new shadow test is assembled for each item to be administered so that the number of assembled shadow tests equals the test length. Adaptations of the shadow-test concept to different test formats such as linear-on-the-fly test (LOFT), multi-stage test (MST), or on-the-fly MST have been devised as freezing a shadow test for some duration to select multiple items from it (van der Linden & Diao, 2014).

This study intends to answer three practical questions: At the individual student level, can we reduce the number of shadow tests reassembled within a single CAT session without compromising its measurement quality? That is, for how long can a shadow test be frozen without causing performance degradation? If so, to what extent and through what mechanism can we optimally reduce this number? Can we expand this reassembly logic to take into account concurrency across students to balance computing requirements?

The remainder of this article is organized as follows: In the next section, we will present the methods employed for studying various reassembly policies for a single student. We will then present the results in the context of a mathematics assessment with complex test blueprint constraints. In the last section, we conclude with limitations, implications of the results for practice, and future research directions. We will put a special emphasis on the applicability of the results to address the above mentioned concurrency issues.

Method

The original design of reassembling the shadow test for each item was predicated on the assumption that the preceding shadow test may not be optimal given the updated ability estimate and that a more optimal test can be assembled from the item pool. However, the shadow test reassembled for the k th item could be equivalent to the $(k-1)$ st item. The main reasons for such equivalence are (a) interim theta estimates that are unchanged or remain in close proximity (e.g., ± 0.1 logits), (b) the relative flatness of the item information functions near their maximum, and/or (c) an item pool that is relatively shallow. For the remainder of this study, we will assume a sufficiently deep item pool and focus on the first aspect only.

Establishing a minimum threshold for reassembling the shadow test can reduce the instances where equivalent shadow tests are constructed due to only minor changes in the ability estimate.

Figure 1. Item pool information function (1,000 items).

In general, interim ability estimates change more notably in early than in latter stages of CAT, indicating a reduced need for reassembly later in the test. Conversely, some large changes in the ability estimate in the early stages may reflect more noise than signal. As a result, instead of adapting to often volatile theta estimates, freezing the shadow test in the early stages may actually be beneficial psychometrically—aside from reducing the computational burden.

We have conducted simulation studies employing the following three reassembly strategies: (a) requiring a minimal theta change before reassembling a shadow test, (b) freezing the shadow test for some items, and (c) imposing a freeze interval while also requiring a minimal theta change. For each strategy, we have simulated a range of values to determine the most effective setting.

Simulation Details

We created a hypothetical item pool modeling after real item pools (1,000 items, 918 multiple-choice [MC] and 82 constructed-response [CR] items), with all items calibrated according to the generalized partial credit model (Muraki, 1992). Figure 1 shows the information function of the item pool.

We simulated a 32-item CAT with a complex set of test blueprint constraints based on the Smarter Balanced Math test specification. Table 1 shows the test constraints. The student ability estimates were determined using the Expected a Posteriori (EAP) with a normal prior $N(0,1)$ for interim and maximum-likelihood estimation (MLE) for final estimation. The simulated student population consisted of $n = 500$ replications at each of the following ability levels $\theta_{\text{true}} =$

Table 1. Test Blueprint Constraints.

Constraints	Lower bound	Upper bound
Test length	32	32
Content Level 1 = 1	17	20
Content Level 1 in 2 or 4	6	6
Content Level 1 = 3	8	8
Content Level 2 = 1A	2	3
Content Level 2 in 1B, 1C, 1I, or 1G	5	6
Content Level 2 in 1D or 1F	5	6
Content Level 2 in 1E, 1J, or 1K	3	4
Content Level 2 = 1H	1	1
Content Level 2 = 2A	2	2
Content Level 2 in 2B, 2C, or 2D	1	1
Content Level 2 in 4A or 4D	1	1
Content Level 2 in 4B or 4E	1	1
Content Level 2 in 4C or 4F	1	1
Content Level 2 in 3A or 3D	3	3
Content Level 2 in 3B or 3E	3	3
Content Level 2 in 3C or 3F	2	2
Content Level 1 = 1 & DOK $\geq$ 2	7	
Content Level 1 in (2 or 4) & DOK $\geq$ 3	2	
Content Level 1 = 3 & DOK $\geq$ 3	2	
ITEM TYPE = DRAG	2	4
ITEM TYPE = EQUATION	12	15
ITEM TYPE = FILL IN	1	2
ITEM TYPE = GRAPH	1	3
ITEM TYPE = HOT SPOT	1	3
ITEM TYPE = MATCHING	2	4
ITEM TYPE = SR MU	1	2
ITEM TYPE = SR SI	5	8

Note. Content Level (1, 2) refers to two primary sub-domains. DOK = Depth of Knowledge, SR MU = Selected Response Multiple Selections; SR SI = Selected Response Single Selection.

$\{-2.5, -2.0, -1.5, -1.0, \dots, 2.5\}$, totaling 5,500 examinees. The starting ability for all students was $\theta_0 = 0$.

For the adaptive formats, the objective function for the selection of the k th shadow test (subject to the constraints in Table 1) was

$$\text{maximize } \sum_{i=1}^{1,000} I_i(\hat{\theta}_{k-1}^{EAP})x_i,$$

where $x_i, i = 1, \dots, 1,000$, is the binary decision variable for the selection of Item i , and $I_i(\theta)$ is the value of Fisher information for Item i at θ .

Fixed Theta Threshold

In our study, we examined the effects of requiring theta change thresholds of 0.1, 0.3, 0.5, 0.7, and 0.9 before reassembling a shadow test for a specific student. The thresholds were fixed throughout the test. We call this policy the *Theta Threshold Policy*. Let l be the item after which we last reassembled a shadow test for student n . To trigger a reassembly prior to deciding on

Figure 2. Audit trail of a hypothetical examinee ($\theta_{\text{true}} = -2.5$, $\theta_{\text{threshold}} = 0.1$).

Note. The horizontal line denotes the true theta value; the bar chart at the bottom shows the score point received for each item (min = 0 and max = 2).

the k th item, the absolute difference between this student's current theta estimate, $\hat{\theta}_{k-1,n}$, and the theta estimate at the last reassembly, $\hat{\theta}_{l,n}$, needed to exceed the set threshold $\theta_{\text{threshold}}$, that is, we require

$$\Delta\hat{\theta}_{k-1,l,n} = |\hat{\theta}_{k-1,n} - \hat{\theta}_{l,n}| \geq \theta_{\text{threshold}}.$$

Note that a Theta Threshold Policy with $\theta_{\text{threshold}} = 0$ yields the standard Shadow CAT logic.

Implicitly, we assumed the following sequence of events: A student completes the $(k-1)$ st item, the ability estimate $\hat{\theta}_{k-1,n}$ is updated and the change in theta $\Delta\hat{\theta}_{k-1,l,n}$ determined, and the algorithm decides on reassembly. Finally, the next item, that is, the k th item, is administered either based on the existing or a newly assembled shadow test.

Figure 2 illustrates the impact of imposing such a limit for a hypothetical examinee with true theta $\theta_{\text{true}} = -2.5$, and threshold $\theta_{\text{threshold}} = 0.1$. The shadow tests were reassembled 15 times in this illustration. Reassembly points are marked with an “S” for shadow test.

The size of the threshold has a direct impact on the number of shadow tests to be reassembled and so should be determined rationally, for example, in accordance with the precision of measurement. For example, the threshold may not be set markedly below the anticipated level of precision to avoid the risk of chasing the noise, especially in the early stages. Conversely, setting it too high may undercut adaptability and lead to performance degradation.

This is not the only way to set a threshold. Other possibilities include variable thresholds that (a) change as the test progresses or (b) depend on the student ability. For example, the threshold can be set to a larger value for the first few items and then switch to a smaller value (or gradually decrease) afterward. Another possibility is to keep a small threshold until the student receives at least one correct and one incorrect response. While the first kind takes into account the increased noise in the early test stages, the second one might offer quicker adaptation for

Item Position	$t_{\text{freeze}} = 0$	$t_{\text{freeze}} = 1$	$t_{\text{freeze}} = 2$	$t_{\text{freeze}} = 3$
1	Assemble	Assemble	Assemble	Assemble
2	Assemble	Freeze	Freeze	Freeze
3	Assemble	Assemble	Freeze	Freeze
4	Assemble	Freeze	Assemble	Freeze
5	Assemble	Assemble	Freeze	Assemble
6	Assemble	Freeze	Freeze	Freeze
7	Assemble	Assemble	Assemble	Freeze
8	Assemble	Freeze	Freeze	Freeze
9	Assemble	Assemble	Freeze	Assemble
10	Assemble	Freeze	Assemble	Freeze
11	Assemble	Assemble	Freeze	Freeze
12	Assemble	Freeze	Freeze	Freeze
13	Assemble	Assemble	Assemble	Assemble
14	Assemble	Freeze	Freeze	Freeze

Figure 3. Assembly schedule based on Freeze Policy (conceptual).
Note. $t_{\text{freeze}} = 0$, a freeze period of zero, is equivalent to the standard Shadow CAT. CAT = computer-adaptive testing.

students for which the initial student ability estimate did not fit well (e.g., students for whom no previous test results exist). Our results, however, indicate that for the individual student, these more elaborate approaches using variable thresholds are not warranted. That is, for the conditions and constraints examined in the current study the differential performance between the fully optimal Shadow CAT and the simpler approaches was not of practical import to call for additional improvements.

Freeze Intervals

As mentioned, the early stages of an adaptive test typically exhibit some fluctuation in the ability estimate. It is far more likely to exceed any imposed theta threshold in the early stages than in the later ones. However, this might largely just be a reaction to noise. Therefore, we studied imposing a freeze period t_{freeze} as follows. With l denoting the item for which we last reassembled a shadow test and k denoting the next item to be administered, we only reassemble if

$$k - l > t_{\text{freeze}}$$

and refer to the resulting policy as the *Freeze Policy*. Note that this is essentially an MST setting, albeit with the sub-test assembled at interim estimates of θ rather than predetermined fixed

values. The columns in Figure 3 visualize the resulting reassembly schedule for three different freeze periods (0, 1, and 3) stating if the decision for item at the next item position requires a newly assembled shadow test or not. Implicitly, the standard Shadow CAT imposes a freeze period $t_{\text{freeze}} = 0$. For the current study, we examined four freeze periods, $t_{\text{freeze}} = \{3, 7, 15, 31\}$, which effectively triggered 8, 4, 2, and 1 reassemblies, respectively.

Hybrid

As before, we fix freeze intervals of $t_{\text{freeze}} = \{3, 7, 15, 31\}$ items, but we additionally condition reassembly on a threshold in the theta change since the last reassembly, just as in the Theta Threshold Policy. Thus, we only reassemble a shadow test for student n after item $(k-1)$ if

$$\Delta\hat{\theta}_{k-1, l, n} = |\hat{\theta}_{k-1, n} - \hat{\theta}_{l, n}| > \theta_{\text{threshold}} \text{ and } k - l > t_{\text{freeze}}.$$

This policy imposes an additional condition on the reassembly schedule and can further reduce the number of reassemblies. We refer to this policy as the *Hybrid Policy*. Note that the Freeze Policy can be regarded as a special case of the Hybrid Policy with $\theta_{\text{threshold}} = 0$.

During the early stages of the test, we will often see that the freeze period prevents reassembly since the threshold condition will be met more regularly. During later stages, although a reassembly would be permissible as the freeze period is over, there might not be a need for a reassembly due to only low theta changes. Therefore, although the freeze period is in theory in effect throughout the whole test and not just in the early stages, it mostly reduces the number of reassemblies during the early stages.

As benchmarks, we simulated two additional testing formats: (a) the standard Shadow CAT (i.e., reassembled a shadow test for each item administered) and (b) LOFT at true theta, θ_{true} (i.e., constructing a shadow test once at each test taker's θ_{true} and freezing it for the duration of the test). We ran all simulations in R with IBM ILOG CPLEX Optimizer.

Results

In this section, we will present the results of our simulations applying the policies explained above. For each condition, the root mean square error (RMSE) and bias of the final θ estimates, $\hat{\theta}_{32}^{MLE}$, are calculated as follows:

$$RMSE(\theta) = \left[n^{-1} \sum (\hat{\theta}_{32}^{MLE} - \theta)^2 | \theta \right]^{1/2}$$

and

$$Bias(\theta) = n^{-1} \sum [\hat{\theta}_{32}^{MLE} - \theta | \theta],$$

respectively. In addition, we computed the mean number of reassemblies for each condition. Tables 2 to 4 present these statistics for all reassembly conditions along with the statistics for the two benchmarks: the standard Shadow CAT and LOFT targeting true theta.

Theta Threshold Policy

For a simulated CAT of 32 items from an item pool of 1,000 items with about 200 constraints (counting lower and upper bounds as separate constraints), we have found that even imposing a low theta threshold of 0.1 reduces the mean reassembly rates to below 50% after eight to 12

Table 2. RMSE for Different Refresh/Freeze Policies by Theta ($N = 5,500$).

	$\Theta = -2.5$	$\Theta = -2.0$	$\Theta = -1.5$	$\Theta = -1.0$	$\Theta = -0.5$	$\Theta = 0.0$	$\Theta = 0.5$	$\Theta = 1.0$	$\Theta = 1.5$	$\Theta = 2.0$	$\Theta = 2.5$
LOFT at true theta	0.205	0.183	0.182	0.174	0.161	0.167	0.164	0.173	0.182	0.173	0.205
At every item (32 times)	0.225	0.191	0.190	0.183	0.168	0.168	0.169	0.168	0.187	0.185	0.230
At every 4th item (8 times)	0.236	0.195	0.179	0.183	0.171	0.167	0.162	0.168	0.187	0.192	0.242
At every 8th item (4 times)	0.248	0.205	0.190	0.187	0.173	0.168	0.164	0.173	0.193	0.215	0.271
At every 16th item (2 times)	0.497	0.265	0.222	0.193	0.164	0.164	0.165	0.187	0.221	0.317	0.511
At every 32nd item (1 time)	2.094	2.131	1.487	0.468	0.189	0.167	0.196	0.565	1.721	2.097	2.174
If $\Delta\theta_{k-1,t,n} \geq 0.1$	0.228	0.192	0.192	0.180	0.170	0.167	0.167	0.169	0.189	0.183	0.228
If $\Delta\theta_{k-1,t,n} \geq 0.3$	0.225	0.193	0.189	0.186	0.169	0.169	0.164	0.172	0.190	0.184	0.232
If $\Delta\theta_{k-1,t,n} \geq 0.5$	0.232	0.190	0.184	0.188	0.174	0.167	0.169	0.171	0.193	0.184	0.232
If $\Delta\theta_{k-1,t,n} \geq 0.7$	0.228	0.197	0.185	0.187	0.163	0.167	0.170	0.182	0.192	0.196	0.241
If $\Delta\theta_{k-1,t,n} \geq 0.9$	0.254	0.202	0.197	0.195	0.163	0.177	0.166	0.192	0.192	0.209	0.256
At every 4th if $\Delta\theta_{k-1,t,n} \geq 0.1$	0.234	0.193	0.178	0.188	0.171	0.165	0.163	0.168	0.186	0.192	0.244
At every 4th if $\Delta\theta_{k-1,t,n} \geq 0.3$	0.234	0.197	0.187	0.193	0.168	0.171	0.169	0.173	0.188	0.196	0.244
At every 4th if $\Delta\theta_{k-1,t,n} \geq 0.5$	0.239	0.198	0.198	0.189	0.170	0.171	0.167	0.175	0.194	0.202	0.249
At every 4th if $\Delta\theta_{k-1,t,n} \geq 0.7$	0.255	0.202	0.190	0.191	0.174	0.173	0.167	0.190	0.199	0.207	0.258
At every 4th if $\Delta\theta_{k-1,t,n} \geq 0.9$	0.261	0.209	0.193	0.200	0.180	0.175	0.172	0.191	0.199	0.216	0.265
At every 8th if $\Delta\theta_{k-1,t,n} \geq 0.1$	0.247	0.204	0.191	0.189	0.174	0.166	0.163	0.172	0.193	0.214	0.271
At every 8th if $\Delta\theta_{k-1,t,n} \geq 0.3$	0.251	0.205	0.192	0.190	0.170	0.169	0.166	0.177	0.195	0.215	0.269
At every 8th if $\Delta\theta_{k-1,t,n} \geq 0.5$	0.255	0.208	0.198	0.192	0.172	0.169	0.164	0.185	0.198	0.218	0.271
At every 8th if $\Delta\theta_{k-1,t,n} \geq 0.7$	0.255	0.215	0.194	0.199	0.176	0.169	0.164	0.192	0.199	0.230	0.276
At every 8th if $\Delta\theta_{k-1,t,n} \geq 0.9$	0.247	0.230	0.201	0.206	0.183	0.168	0.170	0.194	0.202	0.234	0.283

Note. $\Delta\theta_{k-1,t,n}$: The absolute difference between Student n 's current theta estimate after item $k-1$, $\theta_{k-1,n}$, and the theta estimate at the last reassembly, $\theta_{l,n}$. RMSE = root mean square error; LOFT = linear-on-the-fly test.

Table 3. Bias for Different Refresh/Freeze Policies by Theta ($N = 5,500$).

	$\Theta = -2.5$	$\Theta = -2.0$	$\Theta = -1.5$	$\Theta = -1.0$	$\Theta = -0.5$	$\Theta = 0.0$	$\Theta = 0.5$	$\Theta = 1.0$	$\Theta = 1.5$	$\Theta = 2.0$	$\Theta = 2.5$
LOFT at true theta	-0.026	-0.014	0.005	-0.007	-0.009	-0.002	0.008	-0.018	0.006	0.014	0.016
At every item (32 times)	-0.026	0.002	0.012	-0.015	-0.005	-0.001	0.008	-0.017	0.003	0.024	0.009
At every 4th item (8 times)	-0.026	-0.004	0.011	-0.011	-0.006	-0.002	0.005	-0.016	0.006	0.025	0.014
At every 8th item (4 times)	-0.028	0.002	0.005	-0.013	-0.009	-0.004	0.007	-0.017	0.006	0.015	0.021
At every 16th item (2 times)	-0.116	-0.010	-0.012	-0.018	-0.003	-0.002	0.005	-0.009	0.018	0.044	0.097
At every 32nd item (1 time)	-1.501	-1.344	-0.589	-0.081	-0.015	-0.002	0.023	0.120	0.812	1.256	1.620
If $\Delta\theta_{k-1,i,n} \geq 0.1$	-0.027	0.001	0.006	-0.013	-0.007	-0.001	0.008	-0.018	0.005	0.025	0.010
If $\Delta\theta_{k-1,i,n} \geq 0.3$	-0.034	-0.001	0.012	-0.007	-0.009	-0.002	0.010	-0.013	0.004	0.015	0.014
If $\Delta\theta_{k-1,i,n} \geq 0.5$	-0.030	-0.002	0.015	-0.004	-0.006	-0.006	0.011	-0.011	-0.007	0.010	0.009
If $\Delta\theta_{k-1,i,n} \geq 0.7$	-0.029	-0.009	0.009	-0.006	-0.001	-0.004	0.008	-0.019	-0.011	0.014	0.008
If $\Delta\theta_{k-1,i,n} \geq 0.9$	-0.021	0.008	-0.002	0.003	-0.002	-0.003	0.014	-0.032	0.009	-0.011	0.002
At every 4th if $\Delta\theta_{k-1,i,n} \geq 0.1$	-0.026	-0.003	0.006	-0.012	-0.007	-0.004	0.005	-0.015	0.004	0.023	0.015
At every 4th if $\Delta\theta_{k-1,i,n} \geq 0.3$	-0.029	-0.005	0.012	-0.005	-0.013	-0.007	0.010	-0.019	0.003	0.015	0.013
At every 4th if $\Delta\theta_{k-1,i,n} \geq 0.5$	-0.032	0.002	0.007	-0.004	-0.007	-0.005	0.015	-0.019	-0.001	0.008	0.010
At every 4th if $\Delta\theta_{k-1,i,n} \geq 0.7$	-0.026	0.014	0.000	-0.004	0.001	-0.007	0.012	-0.025	0.002	0.002	0.010
At every 4th if $\Delta\theta_{k-1,i,n} \geq 0.9$	-0.026	-0.003	0.006	-0.012	-0.007	-0.004	0.005	-0.015	0.004	0.023	0.015
At every 8th if $\Delta\theta_{k-1,i,n} \geq 0.1$	-0.028	0.001	0.004	-0.009	-0.010	-0.005	0.006	-0.016	0.005	0.016	0.021
At every 8th if $\Delta\theta_{k-1,i,n} \geq 0.3$	-0.027	0.005	0.008	-0.012	-0.009	-0.002	0.004	-0.015	0.003	0.010	0.018
At every 8th if $\Delta\theta_{k-1,i,n} \geq 0.5$	-0.024	0.008	0.006	-0.007	-0.001	-0.004	0.003	-0.014	0.005	0.013	0.019
At every 8th if $\Delta\theta_{k-1,i,n} \geq 0.7$	-0.019	0.004	0.005	-0.002	0.000	-0.004	0.002	-0.020	0.009	0.019	0.014
At every 8th if $\Delta\theta_{k-1,i,n} \geq 0.9$	-0.012	-0.005	0.008	0.007	-0.012	-0.003	0.015	-0.025	0.005	0.019	0.006

Note. $\Delta\theta_{k-1,i,n}$: The absolute difference between Student n 's current theta estimate after item $k-1$, $\theta_{k-1,n}$, and the theta estimate at the last reassembly, $\theta_{l,n}$. LOFT = linear-on-the-fly test.

Table 4. Mean Number of Assemblies for Different Refresh/Freeze Policies by Theta ($N = 5,500$).

	$\Theta = -2.5$	$\Theta = -2.0$	$\Theta = -1.5$	$\Theta = -1.0$	$\Theta = -0.5$	$\Theta = 0.0$	$\Theta = 0.5$	$\Theta = 1.0$	$\Theta = 1.5$	$\Theta = 2.0$	$\Theta = 2.5$
LOFT at true theta	1	1	1	1	1	1	1	1	1	1	1
At every item (32 times)	32	32	32	32	32	32	32	32	32	32	32
At every 4th item (8 times)	8	8	8	8	8	8	8	8	8	8	8
At every 8th item (4 times)	4	4	4	4	4	4	4	4	4	4	4
At every 16th item (2 times)	2	2	2	2	2	2	2	2	2	2	2
At every 32nd item (1 time)	1	1	1	1	1	1	1	1	1	1	1
If $\Delta\theta_{k-1,i,n} \geq 0.1$	14.716	14.034	12.900	11.742	11.436	11.316	10.956	10.612	11.430	12.230	11.896
If $\Delta\theta_{k-1,i,n} \geq 0.3$	6.236	5.588	5.112	4.832	4.740	4.624	4.660	4.870	5.118	5.504	6.048
If $\Delta\theta_{k-1,i,n} \geq 0.5$	4.828	4.106	3.518	3.030	2.850	2.990	2.898	2.700	3.152	3.562	4.176
If $\Delta\theta_{k-1,i,n} \geq 0.7$	3.520	3.010	2.520	2.138	2.052	1.872	1.972	2.190	2.700	3.136	3.744
If $\Delta\theta_{k-1,i,n} \geq 0.9$	3.108	2.812	2.258	1.948	1.616	1.384	1.534	1.932	2.160	2.700	3.016
At every 4th if $\Delta\theta_{k-1,i,n} \geq 0.1$	5.434	5.242	4.844	4.678	4.290	3.874	4.120	4.450	4.686	5.128	5.248
At every 4th if $\Delta\theta_{k-1,i,n} \geq 0.3$	3.946	3.606	3.008	2.850	2.500	1.934	2.458	2.786	2.868	3.430	3.718
At every 4th if $\Delta\theta_{k-1,i,n} \geq 0.5$	3.296	2.964	2.328	2.216	1.972	1.384	1.978	2.306	2.348	2.886	3.200
At every 4th if $\Delta\theta_{k-1,i,n} \geq 0.7$	2.996	2.572	2.126	1.988	1.588	1.134	1.570	2.002	2.100	2.528	3.012
At every 4th if $\Delta\theta_{k-1,i,n} \geq 0.9$	2.850	2.300	2.014	1.836	1.268	1.032	1.280	1.842	2.008	2.304	2.802
At every 8th if $\Delta\theta_{k-1,i,n} \geq 0.1$	3.688	3.508	3.314	3.186	2.972	2.624	2.968	3.194	3.340	3.464	3.658
At every 8th if $\Delta\theta_{k-1,i,n} \geq 0.3$	3.206	2.800	2.518	2.410	2.138	1.414	2.110	2.386	2.482	2.842	3.148
At every 8th if $\Delta\theta_{k-1,i,n} \geq 0.5$	2.976	2.404	2.158	2.094	1.734	1.086	1.714	2.118	2.142	2.524	2.952
At every 8th if $\Delta\theta_{k-1,i,n} \geq 0.7$	2.702	2.198	2.036	1.960	1.398	1.024	1.358	1.960	2.034	2.266	2.698
At every 8th if $\Delta\theta_{k-1,i,n} \geq 0.9$	2.374	2.056	1.996	1.798	1.170	1.006	1.136	1.774	1.996	2.106	2.426

Note. $\Delta\theta_{k-1,i,n}$: The absolute difference between Student n 's current theta estimate after item $k-1$, $\theta_{k-1,n}$, and the theta estimate at the last reassembly, $\theta_{i,n}$. LOFT = linear-on-the-fly test.

Figure 4. Mean shadow-test refresh rates by item position for two theta thresholds (0.1 and 0.3).
 Note. The shaded area around the solid line represents the range for different theta points (i.e., $-2.5, -2.0, \dots, 2.0, 2.5$).

items depending on the theta location (and to 20% after about 14-20 items). As is intuitively clear (and can be seen in Figure 4), the mean refresh rate drops faster with a higher threshold ($\theta_{\text{threshold}} = 0.3$). After about 10 items, the refresh rate drops to 20%.

We also studied the refresh rate (the number of reassemblies) for students of different ability level with different theta threshold values, $\theta_{\text{threshold}} = 0.1, 0.3, \dots, 0.9$. We found that the mean refresh rates dropped the fastest for students with a true theta close to 0 (see Figure 5). This is not surprising as our starting ability θ_0 equals 0 so that students in this theta group immediately get items at their ability level. For our 32-item CAT, the mean number of reassembled shadow tests per examinee ranged from 11 to 15 with $\theta_{\text{threshold}} = 0.1$, again depending on the true theta of the students (see Figure 6). The mean number further reduced to 5 to 6 with $\theta_{\text{threshold}} = 0.3$. That is roughly 80% reduction in the number of reassembled shadow tests per examinee.

Despite these substantial reductions in reassembly rates, little degradation in the measurement accuracy has been observed as evidenced by the RMSE values in Table 2 for a Theta Threshold Policy with $\theta_{\text{threshold}} = 0.1$ or 0.3. Table 2 additionally lists the RMSE for the standard Shadow CAT. While the RMSE (averaged across the θ values) was 1.88 for the standard Shadow CAT, it was 1.88 and 1.89 for $\theta_{\text{threshold}} = 0.1$ and 0.3, respectively. The RMSE values across different theta levels demonstrate that with the exception of extreme values ($\theta \geq |2.0|$) the policies triggering 4 to 6 assemblies produced a comparable level of accuracy as the standard Shadow CAT.

Freeze Policy

Figure 6 graphically displays the RMSE values for different freeze intervals and contrasts them to the two benchmarks (as boundaries of the shaded area shown at the bottom), the most optimal reassembly schedule manifested in the standard Shadow CAT and the theoretical maximum manifested

Figure 5. Mean number of reassemblies by theta for different theta threshold values.

in LOFT targeting true theta. The freeze intervals examined in this study, $t_{\text{freeze}} = \{3, 7, 15, 31\}$, resulted in eight, four, two, and one reassemblies per examinee, respectively. As evident in Figure 6, the RMSE function for eight reassemblies ($t_{\text{freeze}} = 3$) was virtually indistinguishable from that for the standard Shadow CAT even at extreme theta values. The RMSE function for four reassemblies ($t_{\text{freeze}} = 7$) showed some deviations only at the extreme theta values.

Hybrid Policy

The two bottom plots in Figure 7 show the RMSE functions for two freeze intervals ($t_{\text{freeze}} = 3$ and 7) in conjunction with two theta thresholds ($\theta_{\text{threshold}} = 0.1$ and 0.3). Recall that $t_{\text{freeze}} = 3$ is equivalent to reassembling eight times per examinee or at every fourth item after the initial assembly (i.e., prior to deciding on Items one, five, nine, . . . , 29) and that $t_{\text{freeze}} = 7$ to reassembling four times per student or every eighth item after the initial assembly (i.e., 1, 9, 17, and 25). It is not surprising to see that the RMSE functions for the hybrid condition with at most eight reassemblies show substantial similarities with those for the freeze policy of exactly eight reassemblies ($t_{\text{freeze}} = 3$). However, it is interesting to note that the mean number of shadow-test reassemblies for the hybrid setting was 4.7 for $\theta_{\text{threshold}} = 0.1$, and 3.0 for $\theta_{\text{threshold}} = 0.3$. That is, of the eight reassembly opportunities, the shadow test was not reassembled three to five times (depending on the size of the threshold) because the theta threshold was not satisfied. That is a reduction of 85% to 90% in the number of shadow-test reassemblies, compared with refreshing at each item, that is, the standard Shadow CAT.

Conclusion

Even in times of cloud computing and advanced solvers, concurrency requirements put an undisturbed computer-adaptive testing experience at risk. This risk is relatively higher for the

Figure 6. RMSE for different fixed reassembly schedules (1, 2, 4, and 8 times) compared with standard Shadow CAT and LOFT at true θ .
Note. The shaded area was formed by the boundaries of two RMSE functions for the standard Shadow CAT (reassembled 32 times) and the LOFT at true theta values. RMSE = root mean square error; CAT = computer-adaptive testing; LOFT = linear-on-the-fly test.

computing-intensive shadow-test approach that is using MIP in real time. In this study, we have explored various policies reducing the calculation load by minimizing the number of shadow-test reassemblies.

The Theta Threshold Policy requires a certain minimal change in the ability estimate since the last recalculation before triggering a reassembly. While we see differences in the decrease of the refresh rate as the test progresses depending on the true theta of a student, the effect on the overall RMSE is very limited, even for high thresholds.

The Freeze Period introduces fixed intervals between reassemblies, an idea primarily motivated by the noise in the early test stages that caused frequent but unwarranted reassemblies under the Theta Threshold Policy. The Freeze Policy can be further improved by introducing progressively increasing intervals. Freezing the shadow test at a certain point and not refreshing for the remainder of the test can be considered in conjunction with the Freeze Policy.

The Hybrid Policy finally imposes freezing the existing shadow test for a fixed number of items in addition to requiring a minimal theta change. This policy particularly reduces the number of recalculations during the early stages of a test when theta changes are typically the highest—often enough due to a lot of noise. This policy can yield the largest reduction in the number of reassembled shadow tests without lowering the measurement accuracy.

Though these policies are useful in their own right, we can apply the findings to create more flexible versions of these policies directly taking into account the overall system state, early on preventing unpleasant testing experiences due to technical issues. While we have discussed the

Figure 7. RMSE functions for different theta thresholds (0.1 and 0.3), freeze intervals (every fourth item and every eighth item), and hybrids (fourth and 0.1 or 0.3; and eighth and 0.1 or 0.3). Note. The shaded area was formed by the boundaries of two RMSE functions for the standard Shadow CAT (reassembled 32 times) and the LOFT at true theta values. RMSE = root mean square error; CAT = computer-adaptive testing; LOFT = linear-on-the-fly test.

potential approaches, this area needs additional work and empirical studies to fully understand how best to adapt the two policies.

Discussion

Theoretically, maximum adaptation is realized only in fully adaptive testing, with reassembly of the shadow test for each examinee after each administered item (van der Linden & Diao, 2014). However, practical limitations owing to shallow item pools, and excessive content constraints and exposure control needs may overshadow the potential benefits of maximum adaptation. Moreover, the adaptability may be intentionally regulated/constrained for some testing formats, for example, MST, on-the-fly MST, to accommodate some non-psychometric considerations,

Figure 8. Concurrent requests and latency.
Note. Upper panel shows the concurrency in terms of the total number of concurrent requests, and the lower panel shows the pertinent latency in milliseconds for the same period.

for example, skipping and reviewing items within modules or item sets. It is often in the interest of the validity of the test to not allow item review, as most of our real-life decisions and interactions are adaptive and do not allow us to get back. In addition, most empirical research shows that the difference in test scores between conditions with and without review are negligible (Revuelta, Ximénez, & Abad, 2000). Nonetheless, test takers in general prefer to have the

ability to navigate back and forth to review items and their responses. This preference may originate from text anxiety (Stocking, 1997) or being accustomed to paper-and-pencil tests.

In this study, we examined practical and efficient reassembly strategies for Shadow CAT with discrete items. However, the same methodology can be applied to testing situations with both stand-alone items and item sets (e.g., passage-based reading assessments). With item sets as units, however, there are additional considerations and constraints as to where and how frequently reassemblies should/can occur. For example, freezing the shadow test for the duration of a passage-based item set can not only be natural but also conducive to allowing for p/reviewing of items.

Targeted Reassembly to Reduce Concurrency

As already noted, the concurrency requirement for large-scale CAT implementations is immense in current U.S. K-12 consortium assessments. Unfortunately, even in a cloud setting, machines may fail and latency might vary. Figure 8 shows the concurrency and latency that we observed for a relatively small, operational testing program employing Shadow CAT with, approximately, 4,000 students per day and an average testing time of 60 min per student. We observe that there is no direct connection between the concurrency and the latency, which is generally favorably low (less than 50 ms per assembly). But we also note the latency spikes for such a small scale. These spikes can also be due to other factors such as suddenly invoked security features or (temporary) limits when scaling up in the cloud. Consortia settings can increase the number of students per day by a factor 50, increasing not only concurrency but also the likelihood that other factors come into play.

We thus aim at taking the outlined approach a step further and not only reduce the number of reassembled shadow tests per student but also coordinate reassembly across concurrent students. The Hybrid Policy can be regarded as a first step in this direction. If a lot of students start a test within a few minutes, a pure Theta Threshold Policy would still see a large number of concurrent solves during the initial phase until the drop in refresh rate takes effect. As seen above, the Hybrid Policy reduces the number of concurrent solves in this initial phase and maintains the total number of reassemblies to a fraction of the number needed under the standard Shadow CAT. In fact, when we allow for coordination of reassembly across concurrent students, a reverse strategy may even be possible in which next shadow tests are assembled ahead both for a correct and incorrect response for a given student when the demand for reassembly from the other students is low.

Another approach under investigation is a Theta Threshold Policy with variable threshold. As discussed above, the results of our study to date have shown that while the refresh rate is significantly affected by an increase in the applied threshold, the effect on the RMSE is limited. Thus, it seems straightforward to combine this information with available information on the current state of the system, that is, the number of concurrently testing students, their progress in the test, and the observed latency, to increase the threshold, especially for students in the early stages of a test, to reduce the load on the system.

Another potential approach at the macro level, similar to scheduling techniques from Operations Research, involves prioritizing students based on the observed theta gap and imposing a maximum for concurrent reassemblies or solves depending on the state of the system. Unfortunately, in this case, the decision to reassemble a shadow test depends on the whole group of currently testing students, which might introduce a delay in the evaluation of the theta gap.

Large-scale, real-time implementation of automated test assembly, once considered computationally too intensive and hence impractical, has become just another routine in today's

psychometric and computational machinery. With the arrival of cloud computing, the real-time constrained combinatorial optimization problem inherent in Shadow CAT has become scalable practically without limit and solutions are virtually instantaneous. These technological advancements may suggest that freezing the shadow test for computational reasons for any duration at any performance degradation (however small) might not be justifiable. However, the human factor considerations discussed previously, for example, allowing for item reviews, may suggest further research and offer a compelling justification for more economical and yet practical reassembly strategies.

Authors' Note

An earlier version of this article was presented at the annual meeting of the National Council on Measurement in Education, Chicago, Illinois, April 15 to 19, 2015.

Acknowledgments

The authors thank Hao Ren for programming support and Michelle Boyer for editorial assistance.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

References

- Chen, D.-S., Batson, R. G., & Dang, Y. (2010). *Applied integer programming: Modeling and solution*. Hoboken, NJ: John Wiley.
- Muraki, E. (1992). A generalized partial credit model: Application of an EM algorithm. *Applied Psychological Measurement, 16*, 159-176.
- Revuelta, J. O. J., Ximénez, M. C., & Abad, F. J. (2000). Psychometric and psychological effects of review on computerized fixed and adaptive tests. *Psicológica, 21*, 157-173.
- Stocking, M. L. (1997). Revising item responses in computerized adaptive tests: A comparison of three models. *Applied Psychological Measurement, 21*, 129-142.
- van der Linden, W. J. (2005). *Linear models for optimal test assembly*. New York, NY: Springer.
- van der Linden, W. J., & Diao, Q. (2014). Using a universal shadow-test assembler with multistage testing. In D. Yan, A. A. von Davier, & C. Lewis (Eds.), *Computerized multistage testing: Theory and applications* (pp. 101-118). New York, NY: CRC Press.
- van der Linden, W. J., & Reese, L. (1998). A model for optimal constrained adaptive testing. *Applied Psychological Measurement, 22*, 259-270.