Exposure Control Using Adaptive Multi-Stage Item Bundles

Richard M. Luecht

University of North Carolina at Greensboro

April 2003

Paper presented at the Annual Meeting of the National Council on Measurement in Education, Chicago, IL

Introduction

In most forms of operational computer-adaptive testing (CAT), there are tangible trade-offs between the psychometric goals (maximizing test information), the test development goals (satisfying content and other non-psychometric test specifications), and operational goals (minimizing exposure risks). Policies that differentially weight these goals often lead to fairly predictable outcomes. For example, putting the majority of the weight strictly on maximizing test reliability can lead to content validity problems, over-exposure of the most informative items, and a general under-utilization of much of the item pool. Conversely, severe constraints on content and other test specifications usually leads to statistically less-than-optimal tests and may pose security risks—for example, over-exposure of test materials in critical, hard-to-produce content areas.

This paper presents a multistage adaptive testing test development paradigm that promises to effectively handle content balancing and other test development needs, psychometric reliability concerns, and item exposure. The bundled multistage adaptive testing (BMAT) is a modification of the computer-adaptive sequential testing (CAST) framework introduced by Luecht & Nungester (1998). It is also similar to the multiple forms structures (MFS) approach being independently developed by Armstrong and Little (2003). Under BMAT, banks of parallel testlets are constructed to meet various statistical targets and categorical constraints. BMAT requires automated test assembly (ATA)

technology capable of handling multiple, simultaneous objective functions and constraint systems. In addition to handling item exposure directly via ATA constraints on item overlap across testlets, BMAT incorporates random selection of the testlets and can allow randomization of the item presentation sequence within modules to thwart attempts at memorization and other forms of collaborative cheating. The net result is a secure method of building high-quality adaptive and mastery tests that have severe constraints on test content. In this paper, BMAT will be described and demonstrated in the context of high-stakes professional certification and licensure examinations.

The General BMAT Design Framework

Bundled multistage adaptive testing (BMAT) is a variation on the theme of multistage adaptive testing using testlets (see Adema, 1992; Luecht, Hadadi, and Nungester, 1996; Luecht & Nungester, 1998; Luecht, 2000; and Armstrong and Little, 2003). Each **bundle** is created a type of "wrapper" data object comprised of six components: (1) a series of bins to hold the testlets, (2) a bank of testlets, (3) an item bank; (4) test assembly specification sets, (5) design template, and (5) a routing table.

Each bundle has multiple **bins**. Bins are the basic building blocks for this BMAT framework. Testlets are assigned to particular bins. That is, a bin is a data container that holds some fixed number of testlets. For BMAT to work, it is essential that all testlets within each bin be *exchangeable* with each other. In that respect, each bin has [approximately] parallel testlets in terms of their length

(item count and measurement opportunities), statistical characteristics (information and test characteristic functions), content, and other relevant test assembly attributes.

Testlets[1] are sets of items that are administered to examinees as intact units. Testlets can range in size, statistical characteristics, and content related to the competencies underlying the test. Each testlet is associated with a particular bin in a bundle and has many "siblings" that cohabit the bin.. Operationally, test takers can review the items and change answers within a testlet. Review is usually precluded once a given testlet is completed and submitted for scoring. A bank of testlets is constructed for each bundle. Under BMAT, the testlets are preconstructed, undergo any necessary quality assurance checks and reviews, and are then put into a **testlet bank** for a particular bundle.

The **item bank** is a specific collection of items associated with a testlet bank. Each item bank is a subset of a larger item pool for purposes of test assembly. Once the testlet bank and the item bank are generated, however, the item pool no longer becomes relevant.

Every bin in the bundle has an **automated test assembly (ATA) specifications set** comprised of three associated components: a target test information function (TICF), a target test characteristic function (TCF), and a set of content constraints. These ATA specification sets can vary across bins to

[1] Luecht and Nungester (1998) referred to these units as "modules" to avoid other connotations of the term "testlets".

reflect different statistical targets as well as differences in content across or within stages. The test assembly challenge is to manage simultaneous test assembly specification sets to build the testlets for each bin. Two operational ATA alternatives are briefly described in this paper.

Every bundle follows a prescribed **design template** that relates the bins to each another in a multistage context. A design template can be viewed as a schema for each bundle. These bundle design templates can vary with respect to three attributes: (1) the number of adaptive testing stages allowed (i.e., two, three, four, or more stages), (2) the number of bins per stage, (3) the relationship of the bins to one another between and within stages. Figure 1 presents a general design template that has four stages with one (A), three (B, C, D), four (E, F, G, H), and four (I, J, K, L) bins per stage.

[INSERT FIGURE 1 ABOUT HERE]

The solid arrows denote the primary pathways taken by the majority of examinees, assuming that an appropriate IRT model is used. The dotted arrows represent auxiliary pathways that allow the design template to adaptively select the optimal next bin, based on the examinee's provisional score. In general, a design template that has more stages and that uses testlets of more varied difficulty per stage will allow for greater adaptation (Luecht, Hadadi, and Nungester, 1996; Luecht and Nungester, 1998).

Finally, the bins and testlet bank associated with each bundle are used to a compute a **routing table**. A routing table is simply a look-up table that uses the

examinee's provisional score, after completing a particular testlet, to select the appropriate bin at the next stage. Routing tables can be based on number-correct scores[2] or IRT scores—e.g., *expected a posteriori* (EAP) scores (see Luecht, Brumfield, and Breithaupt, 2002).

These BMAT components are not constructed independently of each other. In fact, the fundamental challenge in BMAT lies in engineering a design template and test assembly specification sets for the bins that can be used to consistently manufacture testlets that satisfy the relevant statistical, content, and item exposure demands over time. However, it should be realized that research on efficient engineering designs for bundles, given current computer-based systems, the quality of operational item banks (usually derived from paper-and-pencil testing) and adaptive measurement needs, is still in its infancy.

BMAT Data Structures

Although BMAT could technically be implemented as part of a real-time test assembly and delivery system, it is better to think of the bundles, testlet banks, and routing tables a preconstructed data objects. As noted earlier, the BMAT framework is derivative of Luecht and Nungester's (1998) computer-adaptive sequential test (CAST) "panels". Therefore, the framework is highly structured, yet is easily customizable to integrate with any object-oriented or relational database system that supports the use of hierarchical structures (e.g.,

[2] Number-correct scoring simplifies the amount of data and complexity of real-time scoring required by the test delivery driver.

relational data tables). Every component in BMAT is a formal "entity" that can be generated, evaluated, and [if necessary] repaired before an examinee ever sees the bundle. This provides an important capability: explicit quality control over all test delivery entities. BMAT also facilitates testlet and item exposure management by using simple random sampling procedures to deal with assignment of testlets to examinees, as described in the next section.

The full complement of data structures is too detailed to present here. Suffice to say, for test assembly, BMAT uses the ATA specification sets and the larger item pool as the primary inputs to build the testlet bank and associated item bank for one or more bundles. Moving toward test composition, the testlet bank and the item bank are then integrated via the design template to create the routing table for each bundle. The testlet bank, item bank, routing table There are multiple entry points possible at each step in the development process, to allow quality control procedures to be implemented, as needed, to verify the integrity of the bundle(s) under construction.

Exposure Control Mechanisms under BMAT

There are three key aspects to controlling item exposure for a particular BMAT design template: (1) the expected route proportions; (2) the bin statistical targets; and (3) number of testlets produced per bin for a given bundle.

The test developer can specify the proportion of the population expected to follow various routes in a given bundle. Table 1 provides an example where I have assigned fixed proportions to the primary and auxiliary routes within the 1-

3-4-4 bundle design template presented in Figure 1. I used proportions of 0.30, 0.40, and 0.30 for stage 2, implying that 40 percent of the population should be routed to bin C, with the remainder of the population equally split between the other stage two bins. Other proportions could be used. The auxiliary routes (dotted lines in Figure 1 at stages three and four) are somewhat arbitrarily assigned proportions of 0.05, reflecting the fact that we expect most examinees to follow the primary routes. At stage three, 95 percent of the population from the previous bin (stage two) follows the primary routes. At stage four, only 90 percent of the population follow primary routes to bins J and K, since 10 percent of the examinees will be offloaded to the one of the adjacent bins via the auxiliary routes.

[INSERT TABLE 1 ABOUT HERE]

Since our goal is to maximize test information, it makes sense to choose IRT test information function targets that will be maximally informative for the expected proportions of examinees in the population following the various routes after each stage. Building on suggestions presented in Luecht, Brumfield, and Breithaupt (2002), Luecht and Burgin (2003) provide a rather straightforward algorithm for finding these types of targets and demonstrate a simple implementation of the algorithm with operational test data. Essentially, their algorithm finds pair-wise sets of test information function (TIF) targets that intersect at prescribed values on an IRT θ scale. The expected routing

proportions assigned by the test developer (for example, the values from Table 1) can be used to determine those fixed points.

The final exposure control component is the number of unique testlets assigned to each bin. Since the testlets assigned to each bin are assumed to be exchangeable, the exposure probability for a given testlet, *i:j*, within a given bin, j, becomes

$$P(testlet_{i:j} \mid bin_j) = \frac{P(bin_j)}{K_j} \quad (1)$$

We can also work backward from a prescribed exposure rate to the (rounded) quantity of testlets needed per bin. For example, suppose that we want to ensure a maximum exposure rate of 0.1 for the population. Table 2 is a augmentation of Table 1 that now provides the [rounded] minimum number of testlets needed per bin to keep the maximum exposure at 0.1. In sum, a minimum of 40 testlets would be needed to achieve this goal.

Finally, Table 3 shows what happens when we change the sizes of the bundles (short testlets of five items to long testlets of 25 items). Clearly, using larger testlets significantly increases the bundle size and the overall item inventory demands. Using larger testlets can also limit the degree of potential adaptation (Luecht, 2000) and the viability of building testlets that meet the test assembly specifications sets per bin.

It is important to realize that, by preconstructing the testlets [for a well-engineered bundle] in sufficient numbers, BMAT eliminates to need for formal

exposure controls while the test is running. That is, test delivery engine only need to implement three simple mechanisms in real-time: (i) a scoring algorithm to produce a provisional score after each stage is completed (number-correct scoring is possible); (ii) a routing algorithm that can read the routing table and choose the optimal bin at the next stage; and (iii) a random selection mechanism to randomly choose a testlet from within the current bin. Also, since the testlets and bundles are formal data objects, it is possible to effectively "block" test retakers from seeing particular testlets (or even items) they may have seen before.

Automated Test Assembly Options for BMAT

Automated test assembly (ATA) is an essential part of BMAT. ATA involves the use of mathematical optimization procedures to select items from an item bank for one or more "test forms," subject to multiple constraints related to the content and other qualitative features. van der Linden (1998) presents an excellent overview of the more popular ATA heuristics and mathematical programming techniques.

A simple example may help for purposes of illustration. We start by specifying a quantity to minimize or maximize. This quantity is called the *objective function* and can be formulated as a mathematical function to be optimized by linear programming algorithms or heuristics. *Constraints* are imposed on the solution, usually reflecting the content blueprint or other qualitative features of the items that we wish to control (e.g., word counts). The

constraints are typically expressed as equalities (exact numbers of items to select) or inequalities (upper or lower bounds on the number of items to select).

For example, suppose that we want to maximize the IRT test information at a fixed cut point, denoted θ_0, with a fixed test length of 20 items. We need to define a binary decision variable, x_i, i=1,…,I that indicates that item i is selected (x_i=1) or not (x_i=0) from the item bank. Given this decision variable, the objective function to be maximized is the IRT test information function for the selected items; that is,

$$I(\theta_0) = \sum_{i=1}^{I} I(\theta_0, \xi_i) x_i \qquad (1)$$

where ξ_i denotes the item parameters from the item bank, i=1,…,I (e.g., $\xi_i = \{a_i, b_i, c_i\}$ for the three-parameter logistic model). Now, suppose that we have two content areas, C_1 and C_2, and wish to have at least 5 items from content area C_1 and no more than 10 items from content area C_2. This ATA problem can be modeled as follows:

$$\text{maximize} \quad \sum_{i=1}^{I} I(\theta_0, \xi_i) x_i \qquad \text{(maximum information)} \qquad (2)$$

subject to:

$$\sum_{i \in C_1} x_i \geq 5 \qquad \text{(constraint on } C_1\text{)} \qquad (3)$$

$$\sum_{i \in C_2}^{I} x_i \leq 10 \qquad \text{(constraint on } C_2\text{)} \qquad (4)$$

$$\sum_{i=1}^{I} x_i = 10 \qquad \text{(test length)} \qquad (5)$$

$$x_i \in \{0,1\},\ i=1,\ldots,I. \qquad \text{(range of variables)} \qquad (6)$$

It is relatively straightforward to extend these basic ATA procedures to a multistage, adaptive testlet environment like BMAT, using TIF targets and even test characteristic functions (TCFs; Luecht, 2000). However, these types of problems are not trivial to solve with operation data.

BMAT requires specifically automated test assembly (ATA) software that can handle simultaneous construction of multiple testlets (replications for the bins) using multiple ATA specification sets. Each ATA specification set is comprised of a test information function (TIF) target, a test characteristic function (TCF) target (optional), and a collect of constraints that are intended to manage the content covering the competencies measured by the test as well as and related test design attributes that need to be controlled for each bin. The goal of the ATA process is to generate the required number of non-overlapping testlets needed for each bundle.

Fortunately, there are a number of methods that can handle this type of ATA problem, including mixed integer programming (MIP), using the with shadow test technology (van der Linden and Adema, 1998), generalized network flow algorithms (also solved using MIP, Armstrong and Little, 2003), and

heuristics like the normalized weighted absolute deviation heuristic (NWADH; Luecht, 1998, 2000). An in-depth discussion of these methods is beyond the scope of this paper.

Discussion

This paper is more of prescription for research and development than a solution, per se. BMAT is derivative of another multistage method, CAST, co-developed by the author (Luecht and Nungester, 1998). Other methods such as Armstrong and Little's (2003) multiple forms structures approach are also on the horizon in terms of potential operational use. The obvious question is, "So what?". The corollary is, how are these models different or better than CAT?

In fact, it is interesting to compare a design framework like BMAT to operational CAT and to adaptive testlets methods. In CAT, items are individually selected by a heuristic to maximize the test information at the provisional score, while satisfying various content constraints and other test specifications. Exposure control mechanisms are implemented as indirect "penalty functions" that stochastically restrict the tendency of the heuristic to choose the most informative items for the examinee population. If the test administration unit is changed form a single item to a testlet, CAT would still function in this manner, with the possibility of eliminating the need to check content and other test specifications for the preconstructed testlets. Operationally, the majority of the test assembly and test administration is being

handled in real time, by computer software. The question is, where is quality control implemented in CAT?

In BMAT, we go a step further and implement a formal structure for the testlets that controls the proportion of examinees routed to various bins in the bundle and the quantity of testlets available to administer to a given examinee at a particular bin. The test is therefore engineered to work with three simple mechanisms in real-time (during the test): (i) a scoring algorithm to produce a provisional score after each stage is completed; (ii) a routing algorithm that can read the routing table and choose the optimal bin at the next stage; and (iii) a random selection mechanism to randomly choose a testlet from within the current bin. In contrast to CAT, these multistage models hold the promise of potentially important performance gains by employing simple scoring and routing mechanisms, especially for web-based testing (WBT). Also, because the majority of BMAT components are preconstructed, there is almost unlimited opportunity for quality control.

However, multistage models like BMAT are still in their infancy insofar as research and development. Certainly, these types of models a still some distance from recommended operational use. Fortunately, recent advances in ATA technology have finally made these types of test design models viable, however, there are still many practical issues to be resolved. Some of those issues to be addressed by future research include: (a) determining appropriate item inventory models to generate optimal item pools (e.g. required item counts and

characteristics); (b) investigating optimal design templates and bin ATA specification sets to maximize the production of BOTH the number of tests and the extent of feasible adaptation, from real item pools; and (c) developing ATA methods that allow for controlled item overlap across bins and replications of testlets within bins.

References

Armstrong, R. D. & Little, J. (2003, April). *The Assembly of Multiple Form Structures*. Paper presented at the Annual Meeting f the National Council on Measurement in Education.

Luecht, R. M. (1998). Computer-assisted test assembly using optimization heuristics. *Applied Psychological Measurement, 22*, 224-236.

Luecht, R. M. (2000, April). *Implementing the computer-adaptive sequential testing (CAST) framework to mass produce high quality computer-adaptive and mastery tests.* Symposium paper presented at the Annual Meeting of the National Council on Measurement in Education, New Orleans, LA.

Luecht, R. M., Brumfield, T., & Breithaupt, K. (2002, April). *A Testlet Assembly Design for the Uniform CPA Examination*. Paper presented at the Annual Meeting of the National Council on Measurement in Education. New Orleans.

Luecht, R. M. & Burgin, W. (2003, April). *Test Information Targeting Strategies for Adaptive Multistage Testing Designs*. Paper presented at the Annual Meeting of the National Council on Measurement in Education, Chicago, IL.

Luecht, R. M., Hadadi, A., & Nungester, R. J. (1996, April). *Heuristic-Based CAT: Balancing Item Information, Content, and Exposure*. Paper presented at the Annual Meeting of the National Council on Measurement in Education. New York, NY.

Luecht, R. M. & Nungester, R. J. (1998). Some practical applications of computerized adaptive sequential testing. *Journal of Educational Measurement*, 35, 229-249.

van der Linden, W. J. (1998). Optimal assembly of psychological and educational tests. *Applied Psychological Measurement, 22*, 195-211.

van der Linden, W.J., & Adema, J.J. (1998). Simultaneous assembly of multiple test forms. *Journal of Educational Measurement, 35*, 185-198.

Table 1. Possible Expected Population Proportions per Route for a 1-3-4-4

Stage	Bin Letter	Population Proportion
1	A	1.000
2	B	0.300
2	C	0.400
2	D	0.300
3	E	0.285
3	F	0.215
3	G	0.215
3	H	0.285
4	I	0.282
4	J	0.219
4	K	0.219
4	L	0.282

Table 2. Minimum Required Testlet Counts per Bin for a Maximum Exposure Rate of 0.1

Stage	Bin Letter	Population Proportion	Testlet Count*
1	A	1.000	10
2	B	0.300	3
2	C	0.400	4
2	D	0.300	3
3	E	0.285	3
3	F	0.215	2
3	G	0.215	2
3	H	0.285	3
4	I	0.282	3
4	J	0.219	2
4	K	0.219	2
4	L	0.282	3

Table 3. Bundle Sizes at Various Testlet Sizes

Stages	Testlet Sizes (Possible Item Counts per Testlet)					
1	5	10	10	10	15	20
2	5	5	10	10	15	20
3	5	5	5	10	15	20
4	5	5	5	10	15	20
Bundle Size	200	250	300	400	600	800

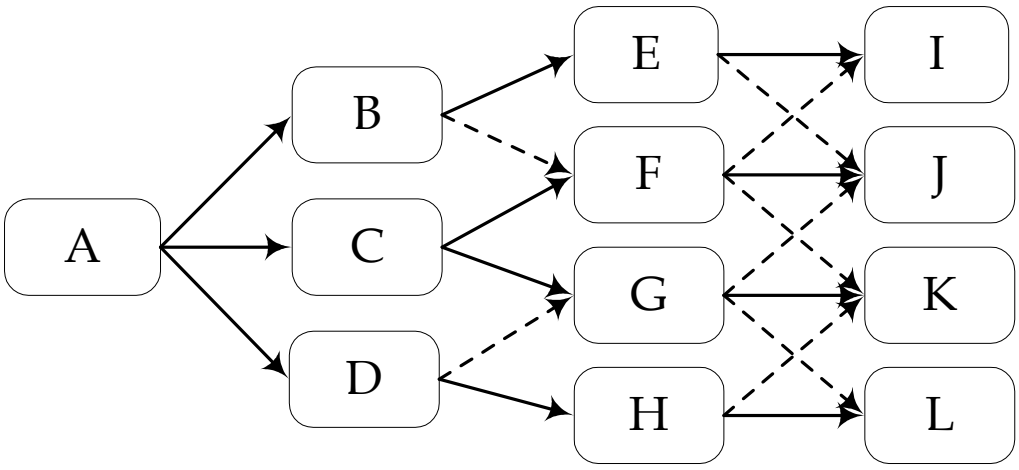

Figure 1. A Design template for a 1-3-4-4 Bundle Design