LSAC RESEARCH REPORT SERIES

A Method to Determine Targets for Multi-Stage Adaptive Tests

Ronald Armstrong
Rutgers University

Louis Roussos
University of Illinois at Urbana-Champaign

Law School Admission Council
Computerized Testing Report 02-07
December 2005

A Publication of the Law School Admission Council

The Law School Admission Council (LSAC) is a nonprofit corporation whose members are more than 200 law schools in the United States and Canada. It was founded in 1947 to coordinate, facilitate, and enhance the law school admission process. The organization also provides programs and services related to legal education. All law schools approved by the American Bar Association (ABA) are LSAC members. Canadian law schools recognized by a provincial or territorial law society or government agency are also included in the voting membership of the Council.

Table of Contents

Executive Summary

The last two decades have seen paper-and-pencil (P&P) tests being replaced by computerized adaptive tests (CATs) for many standardized test administrations. CATs have several advantages as compared to conventional P&P tests. CATs determine the items to administer in real-time; thus, each form is tailored to the test taker's skill level. A test taker's responses to items are recorded during the test and a regularly updated estimate of the test taker's ability is maintained. A CAT can acquire more information about a test taker's ability while administering fewer items. Other advantages of CAT are immediate scoring, more frequent (flexible) administrations, and the ability to utilize constructed response items.

An MFS is an ordered collection of testlets that allows for adaptation based on a test taker's ability while exposing a pre-set number of items and providing a reasonable number of possible forms. This test structure is a hybrid between the conventional P&P and the CAT where the test is divided into stages and a testlet is administered at each stage. A testlet found in an MFS fits specified target curves. The MFS contains multiple stages and consists of bins in which testlets are placed. The bins at a given stage are arranged in levels corresponding to ability classifications. Each form contains one testlet from each stage.

Unlike most CAT implementations, MFSs utilize Target Testlet Information Functions (*TTIFs*) and Target Testlet Characteristic Curves (*TTCCs*). These targets are based on item response theory. Target curves are common in P&P tests because they facilitate accurate test equating by providing parallel tests. As with P&P targets, the MFS targets are chosen to produce reliable tests while utilizing the item pool effectively. However, an MFS is not a linear test and decisions have to be made as to which items to administer. The targets for each bin and the routing rules are interrelated. Routing rules should depend on the targets and vice versa. Specific aspects of the routing methods will be reviewed here and used in target development. The purpose of this paper is to present a method to create targets for the bins of an MFS Design. The ability distribution of the population and an "ideal" item pool determine the targets.

Abstract

A multi-stage adaptive test (MST) is an ordered collection of testlets. A test taker's progression through the network of testlets adapts to the test taker's ability. The collection of paths through the network yields the set of possible test forms. The assembly of an MST requires target information functions and target characteristic curves for the MST design. The targets are chosen to create tests with limited scoring error and high pool utilization. Matching these targets will yield parallel MSTs, in the sense that standardized paper-and-pencil tests are considered parallel. The objective of this paper is to present a method to determine targets for the MST Design based on an item pool and an assumed distribution of test taker ability. This method can be applied to obtain targets for paper-and-pencil tests.

Introduction

The last two decades have seen paper-and-pencil (P&P) tests being replaced by computerized adaptive tests (CATs) for many standardized test administrations. CATs have several advantages as compared to conventional P&P tests. CATs determine the items to administer in real-time; thus, each form is tailored to the test taker's skill level. A test taker's responses to items are recorded during the test and a regularly updated estimate of the test taker's ability is maintained. This estimate is generally based on Item Response Theory (IRT) (Lord, 1980, Hambleton, Swaminathan & Rogers, 1991). Subsequent items to administer are chosen to be well suited to the test taker. A CAT can acquire more information about a test taker's ability while administering fewer items. Other advantages of CAT are immediate scoring, more frequent (flexible) administrations, and the ability to utilize constructed response items. A general review of CATs can be found in Weiss (1985) and Wainer, et al. (1990).

An MST is an ordered collection of testlets that allows for adaptation based on a test taker's ability while exposing a pre-set number of items. This test structure is a hybrid between the conventional P&P and the CAT. It is a computerized extension of the early attempts at P&P adaptive tests by Lord (1971), where multiple-stage tests were given and the tests to give at later stages depended on performance in earlier stages. Partly because the computer was not used, Lord had two stages with an extended time period between stages. A routing test of average difficulty is administered to gauge the test taker's ability. The second stage consisted of three measurement tests developed for low, medium and high abilities. The number of correct responses on the routing test determined the appropriate test at the second level. The flexi-level test described in Lord (1980, pages 115–127) is closely related to the MST approach. Although the flexi-level testing described did not use testlets, it could have. The flex-test possesses many of the properties of the MST. Items are adapted to the test taker's ability, form review is possible and number-right score can quantify the performance.

A conventional CAT form is highly personalized because items are chosen from an item pool during the test. The multi-stage adaptive test (MST) implementation creates multiple parallel test forms. An adjustment to horizontally-equated forms can be made after the administration because the number of test takers responding to each form is large. Also, forms are created prior to delivery and can be reviewed by test specialists. Luecht and Nungester (1998, 2000) present an overview of the MST approach. Patula (1999) compares the MST approach with conventional CAT. Targeting is among the issues discussed in these works.

While no firm commitment has been made as to how MSTs will be administered in an operational setting, it is currently envisioned to follow more closely the current practices of P&P administration than the open sitting format found in some CAT programs. A large number of test takers can register and take the same MST over a brief time period. It is still likely that items will be reused in future MSTs as computer facilities are not currently available to handle the volume of P&P administrations. The test specialist review time can be reduced if testlets from previous MSTs are kept intact when creating MSTs with exposed items. Exposure issues can be addressed by minimizing the testlet/item overlap across MSTs. MSTs will be carefully tracked to prevent a test taker from seeing the same item twice if retested.

A testlet is a set of items bundled together. A discussion on the benefits of testlets can be found in Wainer and Kiely (1987). The testlets found on an MST path combine to match specified target characteristic curves and information functions for the path. An MST has bins where testlets are placed. The bins at a given stage are arranged in levels corresponding to ability classifications. Each form contains a specified number of testlets from each stage. All MSTs considered in this study have either one or two testlets at a stage with a testlet having a maximum of 8 items. The purpose of this paper is to present a method to create targets for the bins of an MST design. The bin targets are then combined to create the path targets that are used in the MST assembly.

Luecht and Nungester (1998) discuss determining targets by matching the reciprocal test information function to a desired degree of accuracy. The conditional error variance of the ability estimate is this reciprocal. A different approach is taken in this paper. The ability distribution of the population and "ideal" item pools determine the targets. The targets are a weighted average of information functions and characteristic curves of items administered from a simulation to be described. It is possible to develop an operational MST without target characteristic curves, but similar characteristic curves across MSTs promote a similar score distribution across MSTs. The following gives an overview of the steps used to create the targets:

1. Step 1. Simulate multiple administrations of a linear test assembled with knowledge of the test takers' true abilities. All constraints for the MST design must be satisfied and exposure control is enforced. The true abilities for the simulation are drawn for the population's ability distribution. Save the observed results of the simulation once the pool's exposure rate has stabilized.

2. Step 2. Consider the bins sequentially starting at bin 1.

3. Step 3. Calculate the probability of reaching the current bin for each test taker recorded in the simulation. Create test-taker weights by dividing the probability of visiting the bin by the sum of all the probabilities; thus, the weights for all test takers sum to one. Create the target for the bin by computing the weighted sum of the observed characteristic curves and information functions associated with the bin.

4. Step 4. Determine the rules for routing test takers out of the current bin to the next stage.

5. Step 5. Proceed to the next bin and return to step 3, or terminate if all bins have been considered.

The next section gives an example of an MST Design. This is followed by a description of the simulation for an omniscient testing method which provides a nearly optimal utilization of the item pool subject to the constraints. The results of this simulation provide data to be used to create targets. The next sections develop the probabilities needed to create targets. A step-by-step procedure to develop targets is given next. MST evaluations are given to demonstrate the effect of target variation on test reliability and information. Finally, a summary discussion is provided.

Multi-Stage Adaptive Test Design

A distinction is made between a multi-stage adaptive test design (MSTD) and a multi-stage adaptive test (MST). An MSTD is a framework and the MST is an occurrence (instance) of this framework. An MSTD is defined by:

a) *Bins*. The outline of an MSTD consists of bins. When an MST is constructed, every bin will contain items that will produce one or more testlets. The MSTD defines bins for different ability groups.

b) *Stages*. A test taker visits exactly one bin at each stage of an MST. The testlets of an MST are administered in sequence, one stage at a time.

c) *Targets*. Every bin in the MSTD has a target information function and a target characteristic curve. The targets for the bins are used to obtain targets for the paths which are used for MST assembly. Every MST has its path information functions and characteristic curves within a certain tolerance level of the path targets.

d) *Number of items per bin*. The lower and upper limits on the number of items that can be assigned to a bin are specified by the MSTD.

e) *Routing rules*. A decision must be made at the end of every stage as to next bin, if any, where the test taker will be routed. The rules must be concise and, in some manner, depend on the test taker's ability. An MST created from an MSTD need not have the same routing rules as the MSTD.

f) *Constraints*. Additional constraints, such as cognitive content, answer key, word, topic, and diversity count may be included in an MSTD. In this study, a cognitive skills content distribution was enforced for all MSTDs, and this restricted the number of items appearing on a path.

An MST is a collection of testlets. Every bin in the MSTD is assigned the required number of items and all requirements of the associated MSTD are satisfied. In practice, many MSTs will be assembled from an MSTD. The items assigned to a bin may be broken into two or more testlets for administrative purposes. For example, the design may specify that several items be administered before a routing decision is made. A testlet with more than eight items may be cumbersome for a test taker to review. Also, if items have a common stimulus, it is natural to create a testlet for those items. The administration aspect does not affect the development of the targets.

In this paper, targets are created from a *Partial MSTD* where selected aspects of the MSTD are omitted. Table 1 gives the outline of an MSTD. Constraints must be included before the targets and routing rules can be calculated; however, enforcing the constraints using mathematical programming is not the subject of this paper and will not be stated explicitly. The constraints varied across MSTDs.

TABLE 1
An MST design table with three stages and six bins by population percentile range

Population Percentile Range	Stage (i) 10–14 items	Stage (ii) 5–7 items	Stage (iii) 10–14 items
[67,100]			Bin 6
[50,100]		Bin 3	
[0,100],[33,67]	Bin 1		Bin 5
[0,50]		Bin 2	
[0,33]			Bin 4

Note. Each bin in the MST depicted is targeted for a particular population percentile range as indicated by the numbers in the left margin. The range on the number of items assigned to each bin is given in the column header. This is a set-based design with 5 to 7 items from each set.

Within a stage, bins are indexed using the convention that the lower index describes a targeting for a lower ability group. For discussion purposes, assume that every testlet for this MSTD contains between 5 and 7 items, and there are between 28 and 30 items on each path. The first stage (Bin 1) will contain two testlets (5–7 items each) designed for the complete ability range of the test-taking population. The test taker advances to one of the two bins in the second stage based on the number of correct responses in the first two testlets. A possible method for routing out of Stage (ii) is the following: Proceed to bin 2 if the total number of correct responses to the items in bin 1 is less than 7, and proceed to bin 3 otherwise. The routing can be based on an ability estimate instead of number correct.

Let $\boldsymbol{\theta}$ denote a random variable giving the ability of a test taker. It is assumed that the distribution of $\boldsymbol{\theta}$ is known. The distribution may be represented by a probability density function or empirically derived; for example, a table with the ability estimates of a previous test administration to the population can be used. Stage (i) contains testlets intended for 100% [0,100] of the test-taking population. Stage (ii) has two bins. The testlet in Bin 2 will be intended for test takers with ability levels in the bottom 50th percentile [0,50] of this population and the testlet in Bin 3 will be intended for test takers with ability levels in the top 50th percentile [50,100]. Stage (iii) has three bins, numbers 4, 5, and 6, intended for testlets that target people at progressively increasing ability. In other words, the testlets assigned to bin 4 are created for test takers in the lowest 33rd percentile [0,33], bin 5 for the middle 34th percentile [33,67], and bin 6 for the top 33rd percentile [67,100] of the test-taking population.

The MST approach to CAT can be implemented with classical test theory, but our application uses a three-parameter Item Response Theory (IRT) model. The Bernoulli random variable $\mathbf{U}_i$ indicates whether the i^{th} item is answered correctly or incorrectly, and the random variable $\boldsymbol{\theta}$ gives the true ability of the test taker. This study assumes that $\boldsymbol{\theta}$ has a normal distribution, $N(\mu,\sigma)$, with mean and standard deviation known. The IRT parameters for item i are denoted by a_i, b_i, c_i. Assume that the parameters are accurately calibrated and the $\mathbf{U}_i$'s are independent of each other. The probability of a correct response from a test taker with ability θ is called the item characteristic curve and is given by the following:

$$P(\mathbf{U}_i = 1 \mid \boldsymbol{\theta} = \theta) = CC_i(\theta) = c_i + \frac{1 - c_i}{(1 + \exp(-1.7 a_i (\theta - b_i))} \tag{1}$$

When stating conditional probabilities, the remainder of this paper gives only the value of the random variable when the reference is apparent from the usage. Thus, $P(\mathbf{U}_i = 1 \mid \boldsymbol{\theta} = \theta)$ becomes $P(\mathbf{U}_i = 1 \mid \theta)$.

Let $IF_i(\theta)$ be the information function of item i (Lord, 1980, page 73).

$$IF_i(\theta) = (1.7a_i)^2 \left[\frac{P(\mathbf{U}_i = 0 \mid \theta)}{P(\mathbf{U}_i = 1 \mid \theta)}\right]\left[\frac{P(\mathbf{U}_i = 1 \mid \theta) - c_i}{1 - c_i}\right]^2 \tag{2}$$

The characteristic curve and information function for the items assigned to bin t of stage s is the sum of the information curves for the individual items. Testlet effects (Bradlow, Wainer & Wang, 1999, Lee, 2000 and Lee, Dunbar, & Frisbie, 2001) are not considered in this paper. Studies suggest that a testlet should be used as the unit of analysis, or the local item dependencies induced by the testlet should be modeled. The theory of this paper will not change as long as independence between testlets is assumed.

Targets and Routing

Unlike most CAT implementations, MSTs utilize target bin information functions (*TBIFs*) and target bin characteristic curves (*TBCCs*). These targets are based on the previously-mentioned item response model. Targets are common in P&P tests because they facilitate accurate test equating by providing parallel tests. As with P&P targets, the MST targets are chosen to produce reliable tests while utilizing the item pool effectively. However, an MST is not a linear test and decisions have to be made as to which items to administer. The targets for each bin and the routing rules are interrelated. Routing rules should depend on the targets and vice versa. Since MSTs associated with an MSTD are assembled after the targets are created, this report concentrates on routing for an MSTD and one routing method is developed later.

A good target for a bin will be based on the sub-population visiting the bin. Let $TBIF_t(\theta)$ and $TBCC_t(\theta)$ represent the *TBIF* and *TBCC* at bin t, $t = 1,\ldots,T$. The procedure does not work with the actual $TBIF_t(\theta)$ and $TBCC_t(\theta)$, but with the values at discrete points on the θ-axis. A linear extrapolation is performed to obtain values for points between the discrete points. Our implementation had points between –3 and +3 in steps of .3.

This study looked at the distribution of Law School Admission Test (LSAT) test-taker's ability over recent years. It was found to be approximately $N(\mu',\sigma')$ where μ' was close to 0 and σ' was close to 1; to be explicit, a $N(.122,.932)$ distribution. The results reported in this paper utilize a $N(0,1)$ distribution of ability. The general approach of creating targets can be implemented with any reasonable ability distribution. The target creation process requires an item pool that accurately reflects the characteristics of *future* item pools to be used by the testing agency. This can be an existing pool or one created by simulation to have the desired attributes. The composition of the pool can have a significant effect on the targets. The study to be reported here utilized a subset of an LSAT operational pool with both discrete and set-based items. The pool dimensions are given later.

The objective of this paper is to describe a method for generating targets from an MSTD based on the ability distribution and item pool. A simulation is used to determine items to be administered at each stage under ideal conditions when the test taker has a known ability. The known ability is drawn from the assumed population. The observed $IF(\theta)$'s and $CC(\theta)$'s are used to create the bin targets by weighting them based on the probability of a test taker visiting a bin. The method can be applied to P&P target generation because this is a special case of the MST where there is a single bin and a single stage.

Omniscient Testing

An omniscient testing method is used to create data for deriving the bin targets. Omniscient testing is a modification of the shadow CAT proposed by van der Linden and Reese (1998), and van der Linden (2000b) where an active test, satisfying all constraints, is maintained and items to deliver are chosen from this active test. The constraints are the same as those specified for an MST path, but without the target constraints. Items already administered to the current test taker are forced to be on the active test and cannot be administered again during this test. The items on the active test are updated at specified points in the administration based on responses to those items fixed on the test and the current ability estimate. An application of the shadow CAT to multidimensional adaptive testing can be found in van der Linden (2002).

Omniscient testing knows the true ability of each test taker before the administration, and uses the true ability to choose items for the active test. The motivation is to produce the best average bin information functions and testlet characteristic curves that the pool will support subject to exposure control and other constraints. The abilities of the test takers are drawn from a $N(0,1)$ distribution in this study.

Objective Function

An important issue is item pool usage. There are various methods to promote the usage of all, or at least most, of the items in the pool. The approach employed here is to adjust the information provided by an item with a penalty based on the empirical exposure rate. Consider the sequential assembly of an individualized test for each of $2K$ test takers—the first k to stabilize the parameters and the second k to obtain data for determining targets. The items administered to the k^{th} test taker are assembled based on knowing all tests administered to the previous $k-1$ test takers. All items used in the assembly of the omniscient tests come from a pool where the items are indexed by D = {1,2,3,...}. Consider the problem of assembling a test for the k^{th} test taker. Let x_i denote a zero-one decision variable for item selection, $i \in$ D. The assembly process has $x_i = 1$ when the i^{th} item is present on the test, and $x_i = 0$ when it is not present. Let θ_k be the true ability of the k^{th} test taker, randomly drawn from the population distribution of ability.

A cost is associated with each item. This cost is the value of a function $H_{ik}(\tilde{u}_i, \bar{u}_i, \theta_k)$ where $\tilde{u}_i$ is a random number uniformly distributed between 0 and 1, and $\bar{u}_i$ is the empirical exposure rate of the ith item. The empirical exposure rate is measured as the total number of tests where the item has appeared over the total number of tests administered ($k-1$, in this case).

The objective of the test assembly problem for the kth test taker is the following:

$$Minimize \sum_{i \in D} H_{ik}(\tilde{u}_i, \bar{u}_i, \theta_k) x_i \ . \tag{3}$$

The study reported here uses the following representation for $\mathrm{H}_{ik}(\tilde{u}_i, \bar{u}_i, \theta_k)$:

$$\mathrm{H}_{ik}(\tilde{u}_i, \bar{u}_i, \theta_k) = \alpha_1 \tilde{u}_i + \alpha_2 \bar{u}_i - \alpha_3 I_i(\theta_k) \, . \tag{4}$$

The coefficients α_1, α_2, and α_3 are pre-defined. Sample values and the reasons for their choice are given later in this section. The computational results section provides selected summary results with the values.

The first term, $\alpha_1 \tilde{u}_i$, creates randomization in the item selection process. This assures no discernible pattern in the administration of items. The value of $\alpha_1 \geq 0$ must be large enough to introduce randomness, but not so large as to dominate the other terms. The value of $\tilde{u}_i$ is generated anew for each test taker. Since objective coefficient is relative to the α values assigned to each term, α_1 = 1.0 for all simulations.

The second term, $\alpha_2 \bar{u}_i$, penalizes items that have already been exposed as a linear expression of the empirical exposure rate. The value of $\alpha_2 > 0$ is chosen based on desired pool usage. This approach gives an acceptable method to distribute items over the testing period. The results of the simulation for the first $K/2$ test takers are not recorded, but are used to stabilize the exposure rates. The ultimate goal is to produce targets that will effectively utilize the item pool. This success of achieving this goal can only be evaluated fully after the targets are defined and MSTs are assembled. Experience with the items used in this study indicates that an immediate goal of keeping the maximum exposure rate under 15% and the median exposure rate around 2% produces acceptable pool utilization. In general, α_2 should increase when the pool size increases and decrease when the number of items on an MST path increases.

The third term, $-\alpha_3 I_i(\theta_k)$, is the focus in a standard CAT implementation where the objective is to maximize information. High information items at a θ_k point should be utilized more than items with lower information, but over the course of the simulation, all acceptable items should be administered. There is a trade-off between information and exposure rate. The higher the value of α_3 relative to α_2, the higher the target information curves and fewer non-overlapping MSTs can be assembled.

The constraints on the optimization problem are those usually associated with automated test assembly. They include limitations on cognitive skills, answer key count, topic, diversity and stimulus usage. The constraints will not be stated explicitly here as they can found in Armstrong, Jones and Kunce (1998), Boekkooi-Timminga (1990), Theunissen (1985) and van der Linden (1998).

Constraints and Assembly

A commercial mixed integer programming (MIP) package was used for the assembly of omniscient tests. An introduction to general MIP theory and models are given in Nemhauser and Wolsey (1988). This studied used CPLEX (ILOG, 2002) to solve the MIP problems, but any software for large-scale MIP solution could be used. Computer programs written in C/C++ interfaced directly with the CPLEX library. The objective function for all the problems was given by (3). The details of the assembly and model constraints are not the focus of this paper. The following outlines the two different models that were used. One model is for discrete items where the stimulus and the question can be treated as a unit. The second model is for set-based items where multiple items are associated with a single stimulus. Models for set-based items are discussed by van der Linden (2000a).

An item pool developed for the P&P LSAT was used in the study. All constraints for the omniscient testing were a scaled version of P&P LSAT constraints. For example, if the MST required half the number of items as the corresponding section in the P&P LSAT, the constraints for the upper and lower limits for word count, cognitive skills distribution and key count distribution were halved. The exception was the most general cognitive skill constraint which corresponded to the enforcing the number of items on a form. The requirement for this constraint was randomly chosen to be a value in the range for the number of items on an MST path; thus, for the omniscient testing assembly, the number of items on a form was fixed. If the number of items on the form was not fixed, maximizing the objective would force the maximum allowable number because the information term in the objective function was the dominant term.

A parameter was set in CPLEX to assure a solution within 10% of the optimal solution. Since randomness is built into the problem, the lack of a true optimal solution was not a concern. Time for solving the MIP was not an issue as discrete item problems terminated after less than one second, and the set based problems after about three seconds.

MST with Discrete Items

The following constraints were considered for the discrete items:

- Single occurrence. An item can appear at most once on a form.
- Cognitive skill content. A distribution of the cognitive skills being tested must be satisfied.
- Answer key count distribution. A constraint on the distribution of the multiple-choice answer keys is imposed.
- Word count. A range on the total number of words found on the form is enforced.

The word count constraint was the only constraint that could not be placed in a network flow model. The network flow model facilitates the convergence of the branch-and-cut algorithm used by CPLEX. Williams (1990) presents modeling methods for MIP.

The sample pool for the study had 1,336 discrete items. A representative discrete item MFSD had 3 stages, 6 bins, and between 35 and 37 items per form. The number of zero-one variables was 1,336, the number of constraints was 25 and the number of nonzero entries in the constraint matrix was 4,055. The objective function, (3), had $\alpha_1 = 1.0$, $\alpha_2 = 25.0$, and $\alpha_3 = 75.0$. The rationale used for this choice of objective parameters is given in the next paragraph. Omniscient simulations that were run on a desktop PC with a 2.0GHz CPU took about 25 minutes with $K = 5{,}000$; that is, 10,000 omniscient forms were assembled. The maximum exposure rate was 15.1% and the median exposure rate was 1.9%. All items were used in at least 3 omniscient forms.

To obtain an appropriate value for α_3, the omniscient test simulation was run without any randomization or exposure control for 1,000 test takers; that is, $\alpha_1 = 0$, $\alpha_2 = 0$, and $\alpha_3 = 1$. The solutions yielded an average per item information at the θ_k's of about .67. Randomization alone should not significantly impact the item choice. Consider two candidate items denoted by i_1 and i_2 where $I_{i_1}(\theta_k)$ is at least 10% larger than $I_{i_2}(\theta_k)$. If $\alpha_1 = 1$, $\alpha_2 = 0$, and $\alpha_3 = 25$, randomization alone would rarely cause the assembly to choose i_2 over i_1 for the omniscient test. The value of α_2 was adjusted, with $\alpha_1 = 1$ and $\alpha_3 = 25$, to achieve a desirable exposure rate distribution. This occurred at $\alpha_2 = 75.0$.

MST with Set-Based Items

The following additional constraints were considered for the set-based items:

- Single occurrence. A stimulus can appear at most once on a form.
- Stimulus to form assignment. A specified number of stimuli must be assigned to the form. This number equals the number of testlets on an MST path.
- Item set usage. When a stimulus is assigned to a form, upper and lower bounds on the total number of items from the associated item set are required.
- Priority items in the set. There may be a subset of items within the item set where at least one item from the subset must appear in the MST when the associated stimulus is assigned to a form.
- Topic specifications. The stimuli for set-based items are categorized according to general topics. Every stimulus has a single general topic; for example, “science” might be a topic. Each MST must have a specified number of stimuli of each topic.
- Diversity specifications. Certain stimuli are oriented toward a diversity group. An MST may have a specified diversity representation enforced.

The model for the set-based items is more complicated than the model for discrete items. As with the discrete item case, much of the problem could be modeled with a network flow approach, but fixed charge nodes were required to account for the item set usage and priority item restrictions.

Two separate set-based item pools were used in the study. The first pool had 110 stimuli and 950 items, and the second had 108 stimuli and 1,021 items. The assembly for second pool enforced diversity constraints and the first pool had no diversity field; otherwise, all the constraint types mentioned above were present. The sample MFSD for the first pool was given by Table 1. The sample MFSD for the second pool had the same structure but between 5 and 8 items per testlet, and between 32 and 34 items on a path. The omniscient test assembly problem for the first pool had 1,060 zero-one variables, 1,227 decision variables, 228 constraints and 3,713 non-zero entries in the constraint matrix. The omniscient test assembly problem for the second pool had 1,129 zero-one variables, 1,274 decision variables, 238 constraints and 3,911 non-zero entries in the constraint matrix. The objective parameters for the two pools were $\alpha_1 = 1.0$, $\alpha_2 = 35.0$, and $\alpha_3 = 65.0$, and $\alpha_1 = 1.0$, $\alpha_2 = 25.0$, and $\alpha_3 = 100.0$, respectively. The parameters were chosen using the same method as described for the discrete item types. The total solution time with $K = 5{,}000$ was about 3.5 hours. The maximum item exposure rate for the first pool was 12.8% and the median exposure rate was 2.4%. The second pool yielded a maximum exposure of 13.1% and a median rate of 2.5%.

Omniscient Testing Administration

The omniscient test items should be administered as they would be administered in an MST. Any sequencing of the administration must be enforced. For example, it may be desirable to begin the CAT with items covering specific topics. A testlet for the set-based items corresponds to a stimulus and associated items. The testlets for the discrete items were assembled in a random manner, where each item was equally likely to be placed in any testlet, and the number of items in a testlet was randomly chosen from the permissible number of items for a testlet. The test was administered one testlet at a time. If more than one testlet could be administered at a stage, the testlet to administer was chosen randomly with equal probability as the other eligible testlets. No adaptation was necessary since the test taker's ability was known. In fact, for purposes of developing targets, there was no reason to simulate the responses to the items. This was done for summary statistics used in comparisons.

Data Saved from Omniscient Testing

During the simulated administration of the omniscient test, data is saved and used when deriving the targets. Let S represent the number of stages in a given MSTD and n_{ks}, $s = 1,...,S$ be the number of items administered to test taker k at stage s. The mean of n_{ks} rounded to the nearest integer, denoted by $\bar{n}_s$, $s = 1,...,S$, is saved. The items administered to the kth test taker during the omniscient test are denoted by the following indices:

$$i(k,s,j),\ \ k = 1,...2K;\ \ s = 1,...,S;\ \ j = 1,...,n_{ks}; \tag{5}$$

where s is the stage and j is the sequencing index of the items within the testlet at stage s.

Each item's $CC_i(\theta)$ and $IF_i(\theta)$ can be computed from (1) and (2). Let L represent the number of discrete points along the ability axis where values for the targets, $TBIF_t(\theta)$ and $TBCC_t(\theta)$, will be provided. Label these points $\tilde{\theta}_l$, $l = 1,...L$. This study used 21 points from –3.0 to +3.0 in steps of .3. The same points are used to save the value of the stage characteristic curves, $SCC_{k,s}(\theta)$, and stage information functions, $SIF_{k,s}(\theta)$, observed for the kth test taker at stage s.

$$SCC_{k,s}(\tilde{\theta}_l) = \sum_{j=1}^{n_s} CC_{i(k,s,j)}(\tilde{\theta}_l),\ \ k = K+1,\ldots,2K;\ s = 1,\ldots,S\ \ l = 1,\ldots,L; \tag{6}$$

$$SIF_{k,s}(\tilde{\theta}_l) = \sum_{j=1}^{n_s} IF_{i(k,s,j)}(\tilde{\theta}_l),\ \ k = K+1,\ldots,2K;\ s = 1,\ldots,S\ \ l = 1,\ldots,L. \tag{7}$$

Weighted sums of the $SCC_{k,s}(\tilde{\theta}_l)$ and $SIF_{k,s}(\tilde{\theta}_l)$ are used to create the $TBCC_t(\tilde{\theta}_l)$ and $TBIF_t(\tilde{\theta}_l)$. The weights are derived from the probabilities found in the next section.

Routing and Probabilities

A *path* through the MST is the sequence of bins that a test taker may traverse during the test administration. Each test taker visits exactly one bin from each stage. A path of an MST provides a test form. The collection of all paths in an MST provides the multiple forms derived from the MST. An *incomplete path* is the set of bins traversed up to some stage $s < S$. Let Φ_s be the random variable representing the bin visited by a test taker at stage s. The initial bin is visited with certainty ($\Phi_1 = 1$ with probability 1) because the design has every test taker being administered the same items at stage 1. Suppose that Φ_s takes on the values ϕ_s, $s = 1, ..., S$ during the administration of an MST. The sequence $\{\phi_1, ..., \phi_s\}$ defines a path. For example, referring to the MST of Table 1, a path is given by $\{\phi_1 = 1, \phi_2 = 3, \phi_3 = 5\}$. The possible paths are not known until the routing rules have been obtained.

Path Probabilities

Mislevy and Chang (2000) present the calculation of path probabilities in a CAT by considering the mechanism for administering items. Local item independence does not imply that a path probability can be calculated by the product of the marginal probabilities when the test is adaptive. The approach is specialized for an MST in this section. The bin information function, $BIF_t(\theta)$, and bin characteristic curve, $BCC_t(\theta)$, for bin t can be obtained from the items assigned to bins once an MST has been assembled. The probability distributions of number correct could be computed from the testlets of the MST. However, the creation of the MST requires the paths and targets; therefore, paths and targets must be developed from the design missing these attributes. The routing rules for an MST need not be the same routing rules as used when creating targets.

The method used to obtain the targets for a bin at stage s requires an estimate of the probability distribution of the number of correct responses for all bins in stages less than s. It is assumed that the targets for all bins at stages less than s are known. This will be the case since the targets are obtained in sequence starting at bin 1. The probability of a correct response, conditioned on ability, can be estimated from $TBCC_t(\theta)$. The probability of a correct response by a test taker with ability θ to any item at bin t of stage s is estimated by $p_t(\theta) = TBCC_t(\theta)/\overline{n}_s$. Let $\mathbf{X}_t$ be a binomial random variable with the following distribution conditional on ability:

$$P(X_t = j \mid \theta) = \frac{\overline{n}_s\,!}{j!(\overline{n}_s - j)!} p_t(\theta)^j (1 - p_t(\theta))^{\overline{n}_s - j};\; j = 0, 1, \ldots, \overline{n}_s. \tag{8}$$

The MST assembly does attempt, even though indirectly, to match the bin targets. Thus, assuming that, on the average, the expectation of number of correct responses to an arbitrary testlet at bin t, conditioned on ability, equals $TBCC_t(\theta)$ is reasonable. Assuming the probability of a correct response to all items at this bin is equal may not be justifiable. It does facilitate the computations and provides a reasonable estimate for this application. It is easily shown that taking $p_t(\theta)$ as the probability of every item at bin t maximizes the variance of the distribution, when compared to other estimates of number correct for testlets from bin t where the item probabilities sum to $TBCC_t(\theta)$.

Let $\mathbf{Y}_s$ be a random variable representing the number of correct responses by a test taker after she/he has been administered the items up to and including the testlet of stage s. It is assumed that $\mathbf{X}_t$, the probability distribution as defined by (8), is an accurate representation of correct responses at bin t. The number of correct responses at the completion of the test is denoted $\mathbf{Y} \equiv \mathbf{Y}_S$. Also, the first bin has $\mathbf{Y}_1 = \mathbf{X}_1$. The probability distribution of $\mathbf{Y}_s$, conditioned on both the test taker's ability and the path traversed, will be derived inductively starting at stage 1. The following assumes that the routing rules are based on the *cumulative number of correct responses* at the time the routing decisions are made.

Assume that the probability distribution of $\mathbf{Y}_{s-1}$, conditioned on both the test taker's ability and the visited bins $\{\phi_1, \ldots, \phi_{s-1}\}$, has been computed. To remain on the a specific path after leaving bin ϕ_{s-1} (i.e., be routed to bin ϕ_s), the test taker must have between $\overline{y}$ and $\overline{\overline{y}}$, inclusive, cumulative correct responses after the completion of the items in bin ϕ_{s-1}. (A method to derive $\overline{y}$ and $\overline{\overline{y}}$ will be presented in the next section of this paper.) The lower limit ($\overline{y}$) cannot be less than 0 and upper limit ($\overline{\overline{y}}$) cannot be more than the number of items administered to this point. The path is possible; thus, there is a positive probability that the test taker can be routed to bin ϕ_s regardless of the value of θ.

Let y represent possible number of correct responses after stage s given a specific path, and n the number of items to administer at stage s. Since the distribution is conditional on the routing, the probability distribution of $\mathbf{Y}_s$ is the following:

$$P(\mathbf{Y}_s = y \mid \theta, \phi_1, \ldots, \phi_s) = \begin{cases} \displaystyle\sum_{x = \max\{y - \overline{\overline{y}}, 0\}}^{\min\{y - \overline{y}, n\}} \mathrm{P}(\mathbf{Y}_{s-1} = y - x \mid \theta, \phi_1, \ldots, \phi_{s-1}) P(\mathbf{X}_s = x \mid \theta)/\Delta_s(\theta), \\ \quad \textit{for } y = \overline{y}, \ldots, \overline{\overline{y}} + n \\ 0, \textit{ otherwise} \end{cases} \tag{9}$$

where

$$\Delta_s(\theta) = P(\phi_s \mid \theta, \phi_1, \ldots, \phi_{s-1}) = \sum_{j = \overline{y}}^{\overline{\overline{y}}} P(\mathbf{Y}_{s-1} = j \mid \theta, \phi_1, \ldots, \phi_{s-1}). \tag{10}$$

The value of $\Delta_s(\theta)$ is the probability of a test taker with ability θ staying on this path at stage s, given that they have been on the path to stage $s - 1$. Multiple paths could have the same sequence of bins up to stage $s < S$. Define $\Delta_1(\theta) = 1.0$.

A sample MST based on Table 1 is utilized to help explain the derivation of (9). Take $s = 3$ and the path to be $\{\phi_1 = 1, \phi_2 = 3, \phi_3 = 5\}$. Consider the routing from bin 3 to bin 5. Assume $\overline{y} = 7$ and $\overline{\overline{y}} = 10$, and $n = 5$. The values $\mathbf{Y}_{s-1}$ with a positive probability are between 7 and 15, inclusive. Take two values for y to illustrate. First, $y = 9$. There are three ways to obtain 9 correct responses at the completion of bin 3. The test taker can enter bin 3 with 7, 8 or 9 correct responses and obtain 0, 1 or 2 correct responses from the bin 3 testlet; therefore, the summation over x is 0, 1 and 2. Next, consider the case where $y = 14$. There are two ways to obtain 14 correct responses at the completion of bin 3. The test taker can enter bin 3 with 9 or 10 correct responses and obtain 5 or 4 correct responses from the bin 3 testlet; therefore, the summation over x is 4 and

5. The division by $\Delta_s(\theta)$ is an application of Bayes formula (see Ross (1997), page 79) and causes the probabilities to sum to 1.

When $s = S$, the probability (9) is the conditional path scoring distribution. The probability of a test taker with ability θ being routed on a path is the following:

$$P(\phi_1, \ldots, \phi_S \mid \theta) = \prod_{q=1}^{S} \Delta_q(\theta). \tag{11}$$

The joint probability distribution of **Y** and a path conditioned only on ability is the following:

$$P(\mathbf{Y} = y, \phi_1, \ldots, \phi_s \mid \theta) = P(\mathbf{Y} = y \mid \theta, \phi_1, \ldots, \phi_s)\, P(\phi_1, \ldots, \phi_s \mid \theta). \tag{12}$$

Probabilities for Populations

The probability of number correct through stage s of path r without conditioning on ability, but knowing the density function of $\boldsymbol{\theta}$ (denoted by $f(\theta)$), is the value of an integral.

$$P(\mathbf{Y} = k \mid \phi_1, \ldots, \phi_s) = \int_{-\infty}^{+\infty} f(\theta) P(\mathbf{Y} = k \mid \theta, \phi_1, \ldots, \phi_s)\, d\theta. \tag{13}$$

It is assumed that $\boldsymbol{\theta}$ is distributed $N(\mu, \sigma)$; thus, numerical integration is required. Gauss-Hermite weights can be used to closely approximate the true value of the integral. The tables by Abramowitz and Stegun (1965) can be used to obtain the values of weights for the integration.

Let Θ_t represent the subpopulation targeted by bin t; for example from Table 1, Θ_t is the abilities between the 33*rd* and 67*th* percentiles. Define $\Delta_s(\Theta_t)$ to be the probability that a test taker from Θ_t is routed to bin ϕ_s given they have been on the specified path through stage $s - 1$, and $\Delta_s(\Theta)$ be the same event but for the complete population. Thus, $\Delta_s(\Theta)$ is the fraction of test takers who have been on the path through stage $s - 1$ to reach bin ϕ_s. The probabilities for the populations are analogous to $\Delta_s(\theta)$ of (10), but it is not conditioned on a specific ability, but on the ability coming from a population. The calculation is similar and is given by the following:

$$\Delta_s(\Theta_t) = P(\Phi_s = \phi_s \mid \theta \in \Theta_t, \phi_1, \ldots, \phi_{s-1}) = \sum_{j=\bar{y}}^{\bar{\bar{y}}} P(\mathbf{Y}_{s-1} = j \mid \theta \in \Theta_t, \phi_1 \ldots, \phi_{s-1}) \tag{14}$$

$$\Delta_s(\Theta) = P(\Phi_s = \phi_s \mid \phi_1, \ldots, \phi_{s-1}) = \sum_{j=\bar{y}}^{\bar{\bar{y}}} P(\mathbf{Y}_{s-1} = j \mid \phi_1 \ldots, \phi_{s-1}). \tag{15}$$

Given bin $t = \phi_s$ at stage s, the probability of a test taker from the subpopulation of traveling the bin sequence $\{\phi_1, \ldots, \phi_s\}$ is the following:

$$P(\phi_1 \ldots, \phi_s \mid \theta \in \Theta_t) = \prod_{q=1}^{s} \Delta_q(\Theta_t). \tag{16}$$

The probability of this routing for the complete population is the following:

$$P(\phi_1 \ldots, \phi_s) = \prod_{q=1}^{s} \Delta_q(\Theta). \tag{17}$$

To develop the routing rules, the fraction of the total population that comes from Θ_t and follows $\{\phi_1,\ldots,\phi_s\}$ is needed. In other words, the probability of $\theta \in \Theta_t$ conditioned on visiting bins $\{\phi_1,\ldots,\phi_s\}$. The fraction of abilities coming from Θ_t and traversing bins $\{\phi_1,\ldots,\phi_s\}$ is given by the following:

$$\psi_s(\Theta_t) = \frac{P(\phi_1\ldots,\phi_s \mid \theta \in \Theta_t)P(\theta \in \Theta_t)}{P(\phi_1\ldots,\phi_s)}. \tag{18}$$

For example, assume the following values for the components of (18):

$$P(\phi_1\ldots,\phi_s \mid \theta \in \Theta_t) = 0.6,\ \ P(\phi_1\ldots,\phi_s) = 0.4 \text{ and } P(\theta \in \Theta_t) = 0.5 \tag{19}$$

Bin t contained 50% of the total population and 0.6 of this 50% would travel the specified path up to stage s. These numbers yield 30% of the test takers arriving at bin t via the specified path and have ability within Θ_t. Forty percent of the total population follows $\{\phi_1,\ldots,\phi_s\}$; thus, 75% of the test takers traversing this incomplete path come from Θ_t, and $\psi_s(\Theta_t) = .75$.

A Routing Rule

Routing rules are necessary to direct a test taker to a bin at the next stage of the MST. Consider the case where the test taker has visited bins $\{\phi_1,\ldots,\phi_s\}$. The routing from a bin ϕ_s at stage s to each bin at stage $s+1$ is considered. The following provides a routing rule for a design using the probabilities developed earlier.

All bins of the MSTD are intended to attract a specific subpopulation. We assume the total population is $N(\mu,\sigma)$. The routing from bin ϕ_s at stage s to a bin at the next stage is considered. The path taken to arrive at ϕ_s is critical for the routing rule. First, the probability of a randomly chosen test taker from the overall population arriving at bin ϕ_s is calculated. This is $\Delta_s(\Theta)$ from (15). Each subpopulation at the next stage is considered. The expected fraction of the test takers visiting bin ϕ_s and whose θ value comes from each targeted subpopulation is computed. This is $\psi_s(\Theta)$ from (18). If any fraction is small (less than .075 say), the fraction is allocated proportionally to the other subpopulation fractions and the small fraction set to zero. Routing will not be permitted to bins when its subpopulation has little chance of arriving at bin ϕ_s. Let $\overline{\psi}_s(\Theta_t)$ be the adjusted expected fraction of test takers at bin ϕ_s arriving via the incomplete path $\{\phi_1,\ldots,\phi_s\}$ and coming from the subpopulation defined by bin t. Test takers are routed out of bin ϕ_s based on these fractions and the population distribution of $\mathbf{Y}_s$ conditioned on the path. Each path will have its own routing rules.

Let y denote the observed number correct after the completion of bin ϕ_s. The objective of the routing is to maximize the number of test takers from Θ_t visiting bin t. Consider the cumulative distribution function of the probability of (13). Let bin t be the bin meant for the lowest ability at stage $s+1$ and having $\overline{\psi}_s(\Theta_t)$ as positive. The value of $\underline{y}$ of (9) for the bin is the smallest number of correct responses possible after the completion of bin ϕ_s. The upper limit on the branching range, $\overline{\overline{y}}$, is the value of y where the cumulative distribution is closest to $\overline{\psi}_s(\Theta_t)$. The next bin to consider at stage $s+1$ is meant for the next highest ability group. The lowest number correct for this routing is the previous $\overline{\overline{y}}$ plus one. The upper limit can be obtained from the cumulative distribution function by having the fraction of those test takers on the path and being routed to the bin as close as possible to the $\overline{\psi}_s(\Theta_t)$ of the bin.

Consider a routing out of bin 1 from an MST based on the design of Table 1 to illustrate the process. Assume that ten items have been administered. The probability distribution and cumulative distribution of $\mathbf{Y}_2$ can be computed from (9). The first routing attempts to divide the test takers in half, with the lower 50% going to bin 2 and the top 50% going to bin 3. The design has the arrival at bin 1 certain for all subpopulation and fractions from the subpopulations arriving at bin 1 are $\psi_2(\Theta_2) = .5$ and $\psi_2(\Theta_3) = .5$. For the sample MST chosen, 11 items were administered at Stage *i* and $P(\mathbf{Y}_1 \le 5 \mid \phi_1 = 1) = .369$ and $P(\mathbf{Y}_1 \le 6 \mid \phi_1 = 1) = .515$; therefore, the best alternative for the attempt to split the population is to send all test takers with a total number correct less-than-or-equal-to 6 to bin 2, and greater than 6 to bin 3.

A more complex routing decision is illustrated considering routing out of bin 3 from this design. Table 2 gives the cumulative distribution of $\mathbf{Y}_3$ conditioned on the path $\{\phi_1 = 1, \phi_2 = 3\}$. There is only one path to bin 3 (bin 2) at stage *ii*. A total of 16 items have been administered to the test taker at the end of stage *ii*. If the test taker had obtained fewer than 7 correct responses at the end of stage *i*, they would have been routed to bin 2; thus, the minimum number of correct responses after the completion of the testlet in bin 3 is 7.

TABLE 2

A cumulative distribution function of the number of correct responses after a random test taker from a N(0,1) population has completed bin 3 of a design from Table 1.

y	7	8	9	10	11	12	13	14	15	16
$\mathbf{Y}_2 \leq$ y	0.013	0.081	0.236	0.443	0.642	0.799	0.905	0.966	0.993	1.00

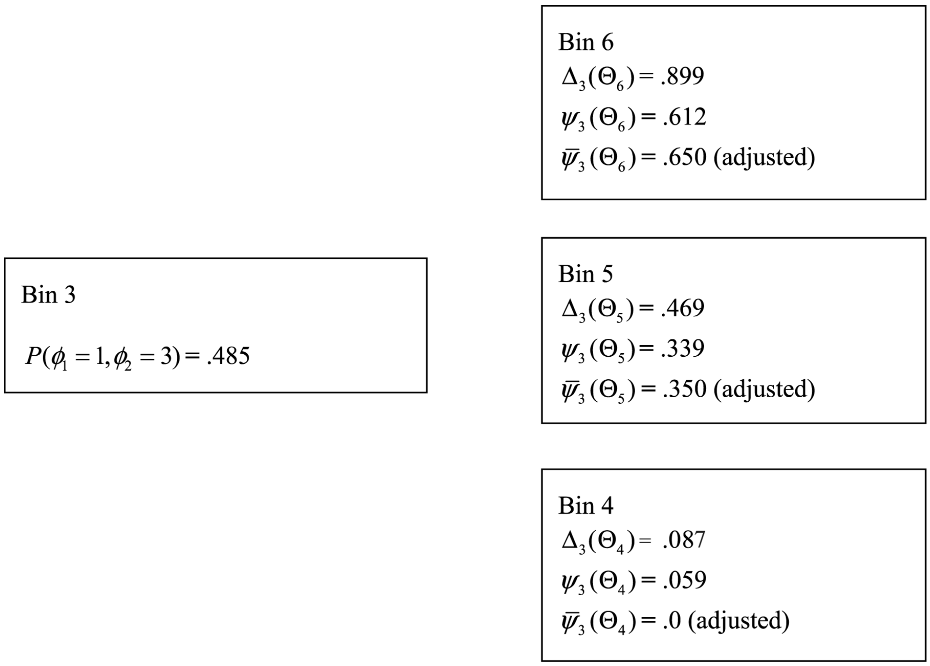

FIGURE 1. *The probability of a randomly chosen test taker following the path to bin 4 is given in the bin 4 box. The boxes for bins 5, 6, and 7 give the probability of a test taker from the associated percentile group arriving at bin 4, the fraction of test takers arriving at bin 4 and coming from the associated percentile group, and the adjusted fractions used for routing.*

There are three subpopulations targeted by bins at the next stage. These are the percentile groups [0,33], [33,67] and [67,100]. Figure 1 gives the expected fraction of the total population to visit bin 3 and the expected fraction to come from each of the subpopulations.

There is no routing from bin 3 to bin 4 because the fraction of those arriving at bin 3 and coming from the lowest 33^{rd} percentile is small ($\psi_2(\Theta_4) = .059$). Routing to bin 5 is considered next. The probability of someone from the percentile group targeted by bin 5 being routed to bin 3 is .469 and 33.9% of the test takers arriving at bin 3 are from Θ_5 . The option of a test taker arriving at bin 3 from the subpopulation of bin 6 (top 33 percentile) is .899, and ($\psi_3(\Theta_6) = .612$. The adjusted numbers are $\overline{\psi}_3(\Theta_4) = 0$, $\overline{\psi}_3(\Theta_5) = .350$ and $\overline{\psi}_3(\Theta_6) = .650$; in other words, the routing decision is based on the assumption that 35.0% of the test takers arriving at bin 3 have an ability from Θ_5. Thus, the test takers with lowest 35.0% of the scores should be routed from bin 3 to bin 5. From the cumulative distribution presented in Table 2, it can be seen that this occurs closest to 10 correct. The rule is to route test takers from bin 3 to bin 5 if he/she has 10 or fewer correct responses, and the remainder to bin 6.

If the adjustment had not been made with the fractions, test takers with 7 and 8 correct responses after bin 3 would have been routed to bin 4. The likelihood of someone from the bottom 33% of the ability scale being routed to bin 3 is small. One classification error is the routing of a test taker with ability $\theta \in \Theta_4$ to bin 5 or 6. Another classification error is the routing of a test taker with $\theta \in \Theta_6$ or $\theta \in \Theta_5$ to bin 4. If the rule were modified to route test takers with 7 and 8 correct responses after bin 3 to bin 4, the probability of a misclassification would be greater than the misclassification error attained by the stated rule. Also, the suppression of possible routings simplifies the final structure. A structure that is easy to work with is important for a successful implementation.

Target Generation

If there is more than one path to bin t, then $\Delta_t(\theta_k)$ given below is the sum of possible $\Delta_t(\theta_k)$ from (10). Let T represent the total number of bins. The following summarizes the approach to generate targets.

6. Step 1. Execute the omniscient test simulation with 2K test takers randomly drawn from the $N(\mu,\sigma)$ population. Results are not recorded to the database for the first K test takers, but are used to stabilize the exposure rate. For each of the next K test takers, save the expected number correct and information obtained over the ability range [–3, +3] with steps of .3 for the items administered at each stage. This is given by (6) and (7). Also, save the mean number of items administered at each stage during the simulation and the true ability of each test taker.

7. Step 2. Let t represent the current bin in the target development process. Targets will be developed sequentially beginning at stage 1; thus, initially, $t = 1$ and $s = 1$.

8. Step 3. Calculate the probability that a test taker with ability θ_k, $k = K + 1,..., 2K$ will be routed to bin t. This is based on (10) adjusted for possible multiple paths. The values obtained for the targets are the following:

$$TBCC_t(\tilde{\theta}_l) = \frac{\sum_{k=K+1}^{2K} \Delta_t(\theta_k) SCC_{k,s}(\tilde{\theta}_l)}{\sum_{k=K+1}^{2K} \Delta_t(\theta_k)}; l = 1,\dots,L. \tag{20}$$

$$TBIF_t(\tilde{\theta}_l) = \frac{\sum_{k=K+1}^{2K} \Delta_t(\theta_k) SIF_{k,s}(\tilde{\theta}_l)}{\sum_{k=K+1}^{2K} \Delta_t(\theta_k)}; l = 1,\dots,L. \tag{21}$$

9. Step 4. Determine the MSTD routing rules out of bin t from the procedures found in the preceding section.

10. Step 5. Let $t = t + 1$. If $t \le T$, set s to be the stage where bin t is found and return to Step 3; otherwise, terminate the process as all targets have been determined.

Figures 2 and 3 show an attempt to fit a symmetric function given by $g(\theta) = H \exp[\bar{\theta} - \theta]^2 / \beta]$ to possible target information functions. Both H and β are parameters adjusted for the fit. Figure 2 presents a target for a percentile group whose ability is centered about 0, and Figure 3 presents a target for a high ability group. The function $g(\theta)$ provides reasonable fit. The characteristics of the item pool are reflected in the fit errors. For example, the high ability target drops below $g(\theta)$ because the pool does not have enough good items with a difficulty close to 3. Figure 2 shows the target a little below $g(\theta)$ to the left of 0 and a little above $g(\theta)$ to the right of 0. The average difficulty of items from the pool is .31, and this can explain the capability to achieve more information with the positive abilities. The pool affects the targets and the targets, in turn, affect the routing.

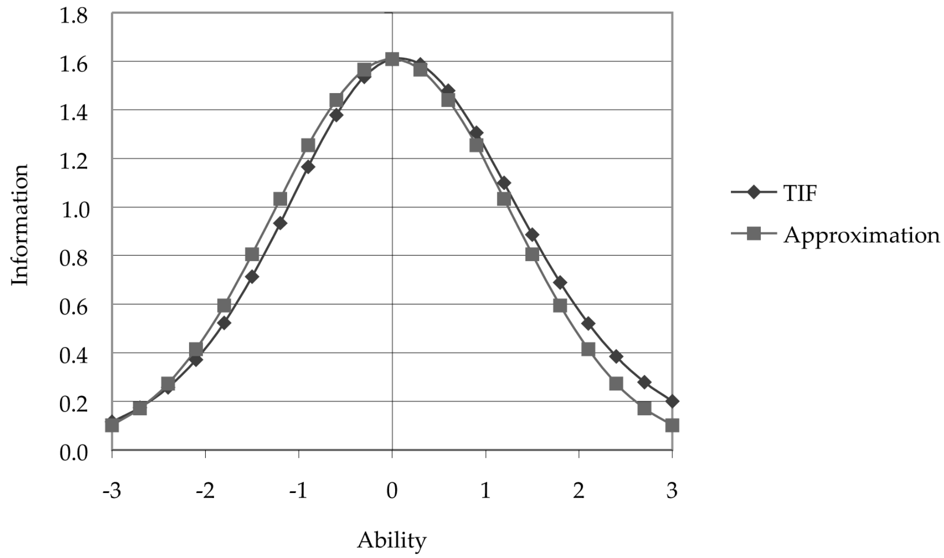

FIGURE 2. *A possible target information function for an ability group with a central tendency close to 0 is plotted against a symmetric function of the form given by* $H\exp[\bar{\theta}-\theta]^2/\beta$.

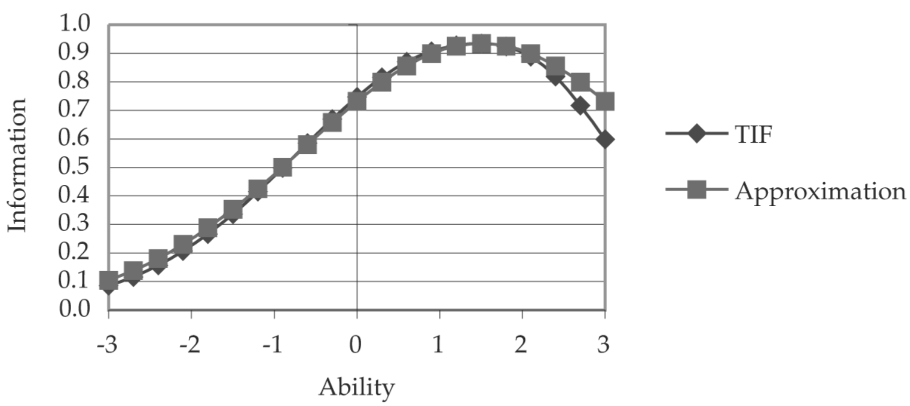

FIGURE 3. *A possible target information function for a high ability group is plotted against a symmetric function of the form given by* $H\exp[\bar{\theta}-\theta]^2/\beta$.

Computational Results

The omniscient test simulations were conducted for each of the three item pools with the α_1, α_2, and α_3 values stated in the omniscient testing section. The targets were created with the process of the previous section. Eight non-overlapping MSTs were assembled with CPLEX for each design. The constraints on each path were the same as the constraints enforced in the omniscient test, but the target constraints were added for the paths. Path information functions and characteristic curves for an MST were required to be within plus or minus 10% of the path targets. No attempt was made to minimize the distance from the target curves. At each point $\theta_l, l = 1, \ldots, L$ the conditional probability of traversing each path was approximated. If the probability was less than 0.1, the target constraint at that point was omitted. This resulted in 22, 25 and 33 target constraint pairs appearing for the three pools, respectively. Once the MSTs were assembled, routing rules were determined for each MST individually.

The MSTs were evaluated based on the scaled score derived from true score equating (Kolen & Brennan, 1995). Scaled scores ranged from 120 to 180. The conditional number correct distribution for each path and a conditional expected scaled score was obtained. The expected squared error of the conditional score was calculated by the following:

$$\sum_y P(Y_s = y | \theta)(SCS(y) - SCS(y_\theta))^2; \quad (22)$$

where $SCS(y)$ is the function creating a scaled score from a true score of y and y_θ is the true score evaluated at θ. Numerical integration was used to derive the scoring error for the population from (22). The square root of this error is defined as the MST's standard error of the scaled score. The fidelity coefficient is the correlation between the $SCS(y_\theta)$ and $SCS(y)$ over the population. Summary results from the assembled forms are given in Table 3.

TABLE 3
Fidelity, unconditional standard error of the scaled score and expected scaled score of eight MSTs

	Discrete Item Pool			Set Based Pool 1			Set Based Pool 2		
MFS #	Fidelity	S.E.	Score	Fidelity	S.E.	Score	Fidelity	S.E.	Score
1	.93	3.69	149.85	.88	5.14	149.90	.92	4.35	149.70
2	.93	3.69	149.86	.89	4.95	149.84	.92	4.26	149.74
3	.93	3.69	149.86	.88	5.27	149.86	.92	4.22	149.79
4	.93	3.75	149.78	.89	4.85	149.86	.92	4.40	149.73
5	.93	3.76	149.78	.89	5.03	149.80	.92	4.24	149.78
6	.93	3.68	149.84	.88	5.09	149.92	.92	4.32	149.62
7	.94	3.64	149.84	.89	4.88	149.90	.91	4.43	149.77
8	.93	3.74	149.83	.89	5.02	149.79	.92	4.17	149.68

Note. The MSTs were assembled from the three pools described in the report.

The effect of the percentiles on the MST was studied by varying the percentiles for the last stage of the MSTD for Table 1. The new design had percentiles of [0,10] for bin 4, [10,90] for bin 5, and [90,100] for bin 6; otherwise, the new design was identical to the design previously studied for the first set-based pool. Since only the percentiles changed, the simulation results from the omniscient test simulation for the pool could be used to create the new design's targets. Eight non-overlapping MSTs were assembled from the new design. The conditional expected information for each of 21 abilities is given for the 16 MSTs in Table 4. The conditional expected information is computed by summing, over all paths, the total information on a path times the probability of a test taker with the given ability traversing the path. The change in information was not dramatic. The three highest ability values obtained consistently more information with the new design. Overall, the original design provided more information for the middle abilities. The change from varying the percentiles would be more pronounced if the exposure control in the omniscient testing was relaxed. This, however, makes the assembly of non-overlapping MSTs more difficult.

TABLE 4
Expected information conditioned on ability for 16 MSTs

	MSTs [0,33], [33,67], [67,100]								MSTs [0,10], [10,90], [90,100]							
θ	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
-3.0	1.06	1.04	0.90	1.06	0.90	1.06	1.08	0.96	1.20	1.00	0.95	1.13	1.03	0.99	1.08	1.03
-2.7	1.38	1.43	1.25	1.48	1.26	1.42	1.48	1.28	1.53	1.35	1.29	1.48	1.38	1.41	1.44	1.35
-2.4	1.75	1.93	1.69	1.99	1.71	1.84	1.98	1.67	1.89	1.77	1.70	1.90	1.80	1.92	1.87	1.72
-2.1	2.18	2.51	2.21	2.57	2.23	2.31	2.56	2.13	2.29	2.26	2.15	2.36	2.26	2.48	2.37	2.15
-1.8	2.67	3.14	2.79	3.18	2.77	2.82	3.16	2.65	2.70	2.79	2.64	2.84	2.74	3.02	2.89	2.61
-1.5	3.18	3.74	3.39	3.79	3.29	3.33	3.70	3.18	3.12	3.31	3.16	3.32	3.20	3.50	3.38	3.08
-1.2	3.71	4.24	3.95	4.34	3.73	3.82	4.10	3.67	3.59	3.77	3.68	3.81	3.63	3.91	3.79	3.51
-0.9	4.22	4.56	4.40	4.78	4.09	4.24	4.33	4.06	4.10	4.11	4.12	4.30	4.01	4.26	4.08	3.88
-0.6	4.66	4.66	4.71	5.03	4.40	4.54	4.46	4.33	4.63	4.32	4.41	4.67	4.30	4.55	4.28	4.20
-0.3	4.96	4.63	4.84	5.08	4.62	4.71	4.54	4.51	5.04	4.43	4.53	4.78	4.52	4.76	4.44	4.50
0.0	5.09	4.57	4.82	4.96	4.70	4.74	4.61	4.64	5.20	4.53	4.55	4.68	4.68	4.86	4.58	4.79
0.3	5.01	4.56	4.70	4.74	4.66	4.72	4.61	4.70	5.09	4.64	4.58	4.54	4.78	4.83	4.67	5.00
0.6	4.79	4.60	4.55	4.53	4.56	4.76	4.56	4.67	4.77	4.74	4.65	4.45	4.77	4.70	4.68	5.01
0.9	4.51	4.64	4.42	4.41	4.51	4.91	4.49	4.55	4.40	4.76	4.66	4.42	4.66	4.55	4.57	4.80
1.2	4.32	4.60	4.32	4.39	4.54	5.05	4.43	4.40	4.17	4.71	4.58	4.43	4.52	4.44	4.38	4.49
1.5	4.27	4.45	4.25	4.36	4.65	5.02	4.34	4.23	4.18	4.61	4.46	4.45	4.42	4.44	4.26	4.24
1.8	4.29	4.17	4.16	4.23	4.71	4.76	4.15	4.06	4.37	4.49	4.38	4.46	4.40	4.47	4.31	4.12
2.1	4.20	3.81	3.95	3.96	4.49	4.34	3.84	3.86	4.51	4.30	4.34	4.38	4.35	4.39	4.45	4.06
2.4	3.83	3.41	3.57	3.55	3.92	3.82	3.41	3.58	4.31	3.96	4.39	4.13	4.13	4.08	4.38	3.92
2.7	3.23	2.97	3.06	3.05	3.18	3.25	2.89	3.17	3.72	3.47	4.06	3.70	3.64	3.57	3.95	3.64
3.0	2.54	2.49	2.50	2.51	2.49	2.65	2.33	2.66	2.93	2.87	3.49	3.15	2.99	2.96	3.25	3.21

Note. Eight MSTs were assembled with final stage percentiles as [0,33], [33,67] and [67,100], and the other 8 MSTs with final stage percentiles [0,10], [10,90] and [90,100].

Conclusions

This paper has presented a method to obtain targets for multi-stage adaptive tests. The method utilizes the ability distribution of the population being tested and the data from item pools to create targets for bins where a population percentile is specified. The item pools must be representative of future pools. Knowledge of the ability distribution of the test-taking population is assumed. This population can be represented by a probability density function, as was used for the examples of the paper, or a table giving the abilities of test takers from a previous administration of a related test.

The results indicate the capability of the targets to capture the desired attributes. If more scoring accuracy is desired at certain ability levels, a bin targeting a narrower percentile of the population can be specified in the design. In general, it has been found that three, or at most four, target levels are desirable at the final stage. The small benefit of the additional levels in improving scoring accuracy is outweighed by the added complexity.

The omniscient test simulation has parameters to create randomness, control item exposure rate and improve information. An attempt has been made to balance the accuracy of the test scores and the effective usage of the item pool. It does not appear to be practical to define “optimal” targets. The future distribution of ability and the components of the future item pools would be needed and, even then, the multiple objectives make a precise definition of optimality difficult. A lengthy appraisal process is required to set targets. Once a test based on the targets for the bins has been made operational, it is difficult to change the targets. The significant time and effort to create and evaluate targets prior to implementation is highly justified.

References

Abramowitz, M., & Stegun, I. A. (1965). *Handbook of Mathematical Functions*. New York: Dover Publications, Inc.

Bradlow, E., Wainer, H., & Wang, X. (1999). A Bayesian random effects model for testlets. *Psychometrika, 64*, 153–168.

Boekkooi-Timminga, E. (1990). The construction of parallel tests from IRT-based item pools. *Journal of Educational Statistics, 15*, 129–145.

Hambleton, R. K., Swaminathan, H., & Rogers, H. J. (1991). *Fundamentals of Item Response Theory*. Sage Publications, Newbury Park, CA.

ILOG (2002). *CPLEX 8.0 User's Manual*. Incline Village NV.

Kolen, M. J., & Brennan, R. L. (1995). *Test Equating Methods and Practices*. New York, Springer.

Lee, G. (2000). A comparison of methods of estimating conditional standard errors of measurement for testlet-based test scores using simulation techniques. *Journal of Educational Measurement, 36*, 91-112.

Lee, G. , Dunbar, S., & Frisbie, D. (2001). The relative appropriateness of eight measurement models for analyzing scores from tests composed of testlets. *Educational and Psychological Measurement, 61*, 958–975.

Lord, F. (1971). A theoretical study of two stage testing. *Psychometrika, 36*, 227-242.

Lord, F. (1980). *Applications of Item Response Theory to Practical Testing Problems*. Lawrence Erlbaum Associates, Hillsdale, NJ.

Luecht, R. M., & Nungester, R. J. (1998). Some Practical Examples of Computer-Adaptive Sequential Testing. *Journal of Educational Measurement, 35*, 229–247.

Luecht, R. M. & Nungester, R. J. (2000). Computer-Adaptive Sequential Testing. In W. van der Linden & C. A. Glas (eds.). *Computerized Adaptive Testing: Theory and practice* (pp. 117–128). Boston, Kluwer.

Mislevy, R. J., & Chang, H. (2000). Does adaptive testing violate local independence? *Psychometrika, 65*, 149–156.

Nemhauser, G., & Wolsey, L. (1988). *Integer and Combinatorial Optimization*. New York: John Wiley & Sons.

Patula, L. N. (1999). *A comparison of computerized adaptive testing and multi-stage testing*. Doctoral Dissertation, University of Massachusetts, Amherst.

Ross, S. (1997). *A First Course in Probability*. Upper Saddle River, NJ, Prentice Hall.

Theunissen, T. J. J. M. (1985). Binary Programming and Test design. *Psychometrika, 50*, 411–420.

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

van der Linden, W. J. (2000a). Optimal assembly of tests with item sets. *Applied Psychological Measurement, 24*, 225–240.

van der Linden, W. J. (2000b). Constrained adaptive testing with shadow tests. In W. van der Linden & C. A. Glas (eds.), *Computerized adaptive testing: Theory and practice* (pp. 27–52). Boston, Kluwer.

van der Linden, W. J., & Reese, L. M., (1998). A model for optimal constrained adaptive testing. *Applied Psychological Measurement, 22*, 259–270.

Veldkamp, B. P., & van der Linden, W. (2002). Multidimensional adaptive testing with constraints on test content. *Psychometrika, 67*, 575–588.

Wainer, H. (1995). Precision and differential item functioning on a testlet-based test: The 1991 Law School Admissions Test as an example. *Applied Measurement in Education, 8*, 157–186.

Wainer, H., & Kiely, G. L. (1987). Item clusters and computerized adaptive testing: A case for testlets. *Journal of Educational Measurement, 24*, 185–201.

Wainer, H., Dorans, N. J., Flaugher, R., Green, B. F., Mislevy, R. J., Steinberg, L., & Thissen, D. (1990). *Computerized Adaptive Testing: A Primer*. Hillsdale NJ: Lawrence Erlbaum Associates.

Weiss, D.J. (1985). Adaptive testing by computer. *Journal of Consulting and Clinical Psychology, 53*, 774–789.

Williams, H.P. (1990). *Model Building in Mathematical Programming*. (3rd edition), Chicester England, John Wiley & Sons, Inc.