Chapter 2
Constrained Adaptive Testing with Shadow Tests

Wim J. van der Linden*
University of Twente, The Netherlands

1. Introduction

The intuitive principle underlying the first attempts at adaptive testing was that a test has better measurement properties if the difficulties of its items match the ability of the examinee. Items that are too easy or difficult have predictable responses and cannot provide much information about the ability of the examinee. The first to provide a formalization of this principle was Birnbaum (1968). The information measure he used was Fisher's well-known information in the sample. For the common dichotomous item response theory (IRT) models, the measure is defined as

\[
I(\theta) = \sum_{i=1}^{n} I_i(\theta) = \sum_{i=1}^{n} \frac{(P'(\theta))^2}{P(\theta)[1 - P(\theta)]},
\]

(1)

where \(P_i(\theta)\) is the probability that a person with ability \(\theta\) gives a correct response to item \(i = 1 \ldots n\).

For the one-parameter logistic (1PL) model, the information measure takes a maximum value if for each item in the test the value of the difficulty parameter \(b_i = \theta\). The same relation holds for the 2PL model, though the maximum is now monotonically increasing in the value of the discrimination parameter of the items, \(a_i\). The empirical applications discussed later in this chapter are all based on response data fitting the 3PL model:

\[
P_i(\theta_j) \equiv c_i + (1 - c_i) \frac{e^{a_i(\theta_j - b_i)}}{1 + e^{a_i(\theta_j - b_i)}}.
\]

(2)

For this model, the optimal value of the item-difficulty parameter is larger than the ability of the examinee due to the possibility of guessing on the items. The difference between the optimal value and the ability

* Part of the research in this chapter received funding from the Law School Admission Council (LSAC). The opinions and conclusions contained in this chapter are those of the author and do not necessarily reflect the position or policy of LSAC.
of the examinee is known to be a monotonically increasing function of the guessing parameter, \( c_t \).

Both test theoreticians and practitioners immediately adopted the information measure in (1) as their favorite criterion of test assembly. The fact that item information additively contributes to the test precision has greatly enhanced its popularity. Though other criteria of item selection have been introduced later (for a review see van der Linden and Pashley, this volume), the most frequently used criterion in computerized adaptive testing (CAT) has also been the one based on the information measure in (1).

Though adaptive testing research was initially motivated by the intention to make testing statistically more informative, the first real-life testing programs to make the transition to CAT quickly discovered that adaptive testing operating only on this principle would lead to unrealistic results. For example, if items are selected only to maximize the information in the ability estimator, test content may easily become unbalanced for some ability levels. If examinees happened to learn about this feature, they may change their test preparations and the item calibration results would no longer be valid. Likewise, without any further provisions, adaptive tests with maximum information can also be unbalanced with respect to such attributes as their possible orientation towards gender or minority groups and become unfair for certain groups of examinees. Furthermore, even a simple attribute such as the serial position of the correct answer for the items could become a problem if the adaptive test administrations produced highly disproportionate use of particular answer keys. Lower ability examinees might benefit from patterned guessing and some of the more able examinees might become anxious and begin second-guessing their answers to previous items. Examinees may get alerted by this fact and start bothering if their answers to the previous questions were correct.

More examples of nonstatistical specifications for adaptive tests are easy to provide. In fact, what most testing programs want if they make the transition from linear to adaptive testing, is test administrations that have exactly the same "look and feel" as their old linear forms but that are much shorter because of a better adaptation to the ability levels of the individual examinees. The point is that adaptive testing will only be accepted if the statistical principle of adapting the item selections to the ability estimates for examinees is implemented in conjunction with serious consideration of many other nonstatistical test specifications.

Formally, each specification that an adaptive test has to meet imposes a constraint on the selection of the items from the pool. As a consequence, a CAT algorithm that combines maximization of statistical information with the realization of several nonstatistical specifications
can be viewed as an algorithm for constrained sequential optimization. The objective function to be optimized is the statistical information in the test items at the current ability estimate. All other specifications are the constraints subject to which the optimization has to take place.

The goal of this chapter is to develop this point of view further and present a general method of constrained sequential optimization for application in adaptive testing. This method has proven to be successful in several applications. The basic principle underlying the method is to implement all constraints through a series of shadow tests assembled to be optimal at the updated ability estimates of the examinee. The items to be administered are selected from these shadow tests rather than directly from the item pool. Use of the method will be illustrated for item pools from several well-known large-scale testing programs.

2. Review of Existing Methods for Constrained CAT

2.1. Item-Pool Partitioning

An adaptation of the maximum-information criterion to make item selection balanced with respect to test content was presented in Kingsbury and Zara (1991). Their proposal was to partition the item pool according to the item attributes. While testing, the numbers of items selected from each class in the partition are recorded. To maintain content balance, the algorithm is forced to follow a minimax principle selecting the next item from the class for which the largest number of items is lacking. A further modification was proposed to prevent items from being readministered to examinees that have taken the same test earlier. Finally, to reduce the exposure of the most informative items in the pool, these authors suggested not to select the most informative item from the current class in the partition but to pick one at random from among the $k$ best items in the class. The last adaptation has also been used in an early version of the CAT-ASVAB (Hetter & Symponson, 1997).

2.2. Weighted-Deviation Method

A more general approach is the weighted deviation method (WDM) by Stocking and Swanson (1993). In their approach, all content specifications for the CAT are formulated as a series of upper and lower bounds on the numbers of items to be selected from the various content classes. The objective of maximum information is reformulated as a series of upper and lower bounds on the values of the test information
function. A weighted sum of the deviations from all bounds is taken as the objective function, with the weights reflecting the desirability of the individual specifications. The items in the adaptive test are selected sequentially to minimize the objective function.

2.3. Testlet-Based Adaptive Testing

Some of the first to address the necessity of combining content specification and statistical criteria in item selection for CAT were Wainer and Kiely (1987). Their solution was to change the size of the units in the item pool. Rather than discrete items, the pool instead comprised "testlets"; that is, bundles of items related to sets of content specifications that can be selected only as intact units. Testlets are preassembled to have a fixed item order. For example, they may be organized according to a hierarchical branching scheme with examinees proceeding with items at a higher (lower) difficulty level if their ability estimate goes up (down). Examinees may proceed from one testlet to another in a hierarchical fashion, a linear fashion, or a combination thereof. New psychometric theory for testlet-based adaptive testing is offered by Wainer, Bradlow, and Du (this volume) and Glas, Wainer, and Bradlow (this volume).

2.4. Multi-Stage Testing

The idea of testlet-based adaptive testing is closely related to the older format of multi-stage testing (Lord, 1980). In multi-stage testing, examinees proceed through a sequence of subtests, moving to a more difficult subtest if they do well but to an easier one if their previous performances are low. Though the earlier literature discussed a paper-and-pencil version of this format with nonstatistical scoring of the ability of the examinees after each subtest, the advent of computers in testing practice has made an implementation with statistical estimation of ability after each subtest possible. Adema (1990) van der Linden and Adema (1998) offer 0-1 linear programming models for the design of multi-stage testing systems based on the maximum-information criterion that allow for a large variety of constraints on the composition of the subtests. The possibility to include such constraints into multistage testing systems has rekindled the interest in this testing format (Luecht & Nungester, this volume; 1998).

2.5. Evaluation of Existing Approaches

The above approaches differ in an important way. The first two approaches implement the constraints through a modification of the item
selection algorithm. The last two approaches build all constraints directly into the units in the pool from which the test is administered. This distinction has consequences with respect to:

1. the degree of adaptation possible during the test;
2. the extent of item coding necessary;
3. the possibility of expert review of actual test content;
4. the nature of constraint realization; and
5. the possibility of constraint violation.

Both the item-pool partitioning and WDM approach allow for an update of the ability estimate after each item. Thus, they offer the maximum degree of adaptation possible. However, to be successful, both approaches assume coding of all relevant item attributes. If a potentially important attribute is overlooked, test content may still become unbalanced. The WDM minimizes only a weighted sum of deviations from the constraints. Some of its constraints can therefore be violated even with complete coding of the items. In either approach, both the selection of the items and the realization of the constraints is sequential. Though sequential item selection allows for optimal adaptation, sequential realization of constraints is less than ideal. Algorithms with this feature tend to favor items with an attractive value for the objective function early in the test. However, the choice of some of these items may turn out to be suboptimal later in the test. If so, the result is less than optimal adaptation to the ability estimates and/or the impossibility to complete the test without constraint violation.

The testlet-based and multi-stage approach have the option of expert review of all intact testing material prior to administration. Explicit coding of relevant item attributes is not always necessary. However, these approaches lack full adaptation of item selection to the ability estimates (multi-stage testing and linear testlet-based CAT) or allow only for partial adaptation (testlet-based CAT with hierarchical branching). Also, the task of assembling a pool of testlets or a multi-stage testing system such that any path an examinee takes satisfies all constraints involves a huge combinatorial problem that can quickly become too complicated to be solved by intuitive methods. The result may be a suboptimal branching system and/or constraint violation. However, as already noted, formal methods for assembling multi-stage testing systems do exist. Use of these methods does assume explicit coding of all relevant item attributes. Both in the testlet-based CAT and multi-stage testing approaches, groups of items are selected sequentially. Hence, adaptation of item selection to ability estimation is suboptimal. On the other hand, the constraints are realized simultaneously when assembling the testlets or subtests prior to the test administration, pro-
vided formal methods and adequate item coding are used to assemble the pool of testlets or the multi-stage testing system.

This evaluation of the existing methods for constrained CAT reveals an important dilemma. An algorithm with optimal properties would need to select its items sequentially to allow for optimal adaptation but realize all constraints simultaneously to prevent violation of certain constraints or suboptimal adaptation later in the test. Possible solutions to the dilemma are: (1) to allow the algorithm to work backwardly to improve on previous decisions or (2) to have the algorithm project forwardly to take future consequences of decisions into account. In adaptive testing, backtracking is impossible; the algorithm is applied in real time and earlier choices can not be undone. Thus, the only possibility left is to have the algorithm project forwardly each time a new item is selected. This is exactly what the new class of algorithms presented in this chapter do.

3. Constrained CAT with Shadow Tests

The basic concept of a shadow test is illustrated in Figure 1. The selection of each new item in the CAT is preceded with the on-line assembly of a shadow test. A shadow test is a full-length test that: (1) meets all the test constraints; (2) contains all items administered previously; and (3) has maximum information at the current ability estimate. The item to be administered is the one with maximum information in the shadow test. The next actions are updating the ability estimate of the examinee, returning the unused items to the pool, and repeating the procedure.

The following pseudo-algorithm gives a more precise summary of the idea:

Step 1: Initialize the ability estimator.
Step 2: Assemble a shadow test that meets the constraints and has maximum information at the current ability estimate.
Step 3: Administered the item in the shadow test with maximum information at the ability estimate.
Step 4: Update the ability estimate
Step 5: Return all unused items to the pool.
Step 6: Adjust the constraints to allow for the attributes of the item already administered.
Step 7: Repeat Steps 2-6 until n items have been administered.

Observe that test length has been fixed in this algorithm. This choice is in agreement with practice in nearly all existing CAT programs. Though a stopping rule based on a predetermined level of accuracy
for the ability estimator is desirable from a statistical point of view, it seems impossible to guarantee always the same specifications for all examinees for a test with random length.

The ideal in constrained adaptive testing is a test that is feasible (i.e., meets all constraints) and has an optimal value for the objective function in (1) at the true ability of the examinee. Since the true ability is unknown, all one can hope for is an item-selection mechanism with a result approximating this ideal case. The proposed adaptive testing scheme has this feature; it yields feasible adaptive tests converging to the optimal value for the information function at the true ability of the examinees.

This claim can be shown to hold as follows. The algorithm realizes all constraints simultaneously for each shadow test. Each next shadow tests contains all items administered previously. Thus, the last shadow test is the actual adaptive test and always meets all constraints. Further, each shadow test is assembled to have a maximum value for the information function in (1) and the item selected from the shadow test has a maximum contribution to this function. For a consistent ability estimator, it follows from Slutsky’s theorems (e.g., Ferguson, 1996) that the value for the function in (1) converges to the maximum value possible at the true ability of the examinee. Mild conditions for the ML
estimator with maximum-information item selection to be consistent are formulated in Chang and Ying (in press).

This argument assumes an infinitely large item pool with known item parameter values. However, the conclusion is expected to hold closely enough for all practical purposes for any well-designed, finite item pool. Of course, the speed of convergence depends on size and nature of the set of constraints. For a severely constrained adaptive test from a small item pool, convergence may be slower than for a test from a large pool involving only a few constraints. The empirical examples later in this chapter will shed some light on the question how fast the ability estimator converges in typical applications of this procedure for constrained CAT.

4. Technical Implementation

The idea of constrained adaptive testing with shadow tests was introduced in van der Linden and Reese (1998), who used the technique of 0-1 linear programming (LP) to assemble the shadow tests. The same idea was explored independently in Cordova (1998), whose test assembly work was based on the network-flow programming approach introduced in Armstrong and Jones (1992). In principle, any algorithm for automated test assembly that generates an optimal feasible solution and is fast enough for application in real time can be used to implement the above adaptive testing scheme. Even for test assembly heuristics that tend to provide suboptimal solutions, such as the WDM or the normalized weighted absolute deviation heuristic (NWADH) by Luecht (1998; Luecht & Hirsch, 1992), considerable gain over the existing methods of constrained adaptive testing can be expected when implemented in the proposed scheme. A recent review of approaches to automated test assembly is given in van der Linden (1998).

The examples later in this chapter are all based on the technique of 0-1 LP. This technique allows us to deal with virtually any type of constraint that can be met in test assembly and thus offers maximum flexibility when modeling the problem of shadow test assembly. In addition, general software for 0-1 LP is available that can be used to solve such models in real time for the type of CAT pools met in real life.

4.1. Basic Notation and Definitions

To maintain generality, a 0-1 LP model for the assembly of shadow tests from an item pool with some of its items organized as sets with
a common stimulus is formulated. This testing format has become increasingly popular; several of the item pools used in the empirical examples later in this chapter involved this format. Typically, in testing with set-based items, the numbers of items per stimulus available in the pool are larger than the numbers to be selected in the test.

The following notation will be used throughout the remainder of this chapter:

- **items in the pool:** \( i = 1, \ldots, I; \)
- **stimuli in the pool:** \( s = 1, \ldots, S; \)
- **set of items in the pool with stimulus** \( s \): \( U_s, s = 1, \ldots, S; \)
- **items in the adaptive test:** \( k = 1, \ldots, n; \)
- **stimuli in the adaptive test:** \( l = 1, \ldots, m. \)

Thus, \( i_k \) and \( s_l \) are the indices of the \( k \)th item and \( l \)th stimulus in the adaptive test, respectively. Using this notation, \( S_{k-1} \equiv \{ i_1, \ldots, i_{k-1} \} \) is defined as the set of the first \( k - 1 \) items administered in the test. Consequently, \( R_k \equiv \{ 1, \ldots, I \} \setminus S_{k-1} \) is the set of items remaining in the pool after \( k - 1 \) items have been selected in the test.

The \( k \)th shadow test is denoted as \( T_k \equiv \{ i_1, \ldots, i_{k-1}, i'_k, \ldots, i'_n \} \) whereas \( S_l \equiv \{ s_1, \ldots, s_l \} \) is defined as the set of the first \( l \) stimuli in the test. If the constraints on the number of items for \( l \)th stimulus in the adaptive test have not yet been satisfied, \( s_l \) is called the active stimulus and \( U_{s_l} \) the active item set. If \( s_l \) is active, the next item is selected from \( U_{s_l} \cap \{ i'_k, \ldots, i'_n \} \). Otherwise, it is selected from \( \{ i'_k, \ldots, i'_n \} \). Therefore, the list of eligible items in the \( k \)th shadow test is defined as

\[
A_k = \begin{cases} 
U_{s_l} \cap \{ i'_k, \ldots, i'_n \} & \text{if the } l \text{th stimulus is active;} \\
\{ i'_k, \ldots, i'_n \} & \text{otherwise.}
\end{cases} \tag{3}
\]

Let \( \hat{\theta}_{k-1} \) denote the ability estimate updated after the first \( k - 1 \) items in the adaptive test. It thus holds that the \( k \)th item in the adaptive test is

\[
i_k = \max_i \{ I_i(\hat{\theta}_{k-1}); i \in A_k \}. \tag{4}
\]

When assembling the shadow test, the objective function should be maximized only over the set of items eligible for administration. In particular, if the \( l \)th stimulus is active, it may be disadvantageous to maximize the information in the shadow test over items not in \( U_{s_l} \) (even though such items are needed to complete the shadow test). To implement this idea for the objective function in the model below, the following set is defined:

\[
O_k = \begin{cases} 
U_{s_l} & \text{if the } l \text{th stimulus is active;} \\
R_k & \text{otherwise.}
\end{cases} \tag{5}
\]
4.2. 0-1 LP Model for Shadow Test

The model is an adapted version of the one presented in van der Linden and Reese (1998). To formulate its objective function and constraints, 0-1 valued decision variables \( x_i \) and \( z_s \) are introduced. These variables take the value 1 if item \( i \) and stimulus \( s \) is selected in the shadow test, respectively, and the value 0 otherwise.

In addition, the following notation is needed to denote the various types of item and stimulus attributes that may play a role in the assemble of the shadow test:

- **categorical item attribute** : \( C^i_e, e = 1, ..., E; \)
- **categorical stimulus attribute** : \( C^s_f, f = 1, ..., F; \)
- **quantitative item attribute** : \( q_i, i = 1, ..., I; \)
- **quantitative stimulus attribute** : \( q_s, s = 1, ..., S; \)
- **sets of mutually exclusive items** : \( V^i_g, g = 1, ..., G; \)
- **sets of mutually exclusive stimuli** : \( V^s_h, h = 1, ..., H. \)

Categorical item attributes, such as item content or format, partition the item pool into classes \( C^i_e, e = 1, ..., E. \) Note that each attribute involves a different partition. For simplicity, only the case of one attribute is discussed; adding more constraints is straightforward. The model below requires the number of items in the test from class \( C^i_e \) to be between \( n^l_e \) and \( n^u_e. \) Likewise, the set of stimuli in the pool is partitioned by a categorical attribute \( C^s_f, f = 1, ..., F, \) and the number of stimuli from class \( C^s_f \) is required to be between \( n^l_f \) and \( n^u_f. \) In addition, the items and stimuli are assumed to be described by categorical attributes, such as a word count or an item difficulty parameter. For simplicity, the case of one attribute with value \( q_i \) for item \( i \) and \( r_s \) for stimulus \( s, \) respectively, is discussed. The model requires the sum of these values for the items and stimuli in the test to be between in the intervals \( (q^{il}, q^{iu}) \) and \( (q^{st}, q^{su}), \) respectively. Note that in the model below in fact average values are constrained since the model also fixes the total numbers of items and stimuli in the test. Finally, the use of logical constraints to model test specifications is illustrated. It is assumed that the item pool has sets of items, \( V^i_g, g = 1, ..., G; \) and sets of stimuli, \( V^s_h, h = 1, ..., H; \) that exclude each other in the same test, for instance, because knowing one of them facilitates solving the others.

The model is formulated as follows:

\[
\text{maximize } \sum_{i \in O_k} I_i(\hat{\theta}_{k-1})x_i \quad \text{(maximum information)} \quad (6)
\]
subject to

\[
\sum_{i=1}^{I} x_i = n, \quad \text{(\# of items)} \tag{7}
\]
\[
\sum_{s=1}^{S} z_s = m, \quad \text{(\# of stimuli)} \tag{8}
\]
\[
\sum_{i \in S_{k-1}} x_i = k - 1, \quad \text{(items already selected)} \tag{9}
\]
\[
\sum_{i \in U_s} x_{i \geq n_s^l} z_s, \quad s = 1, \ldots, S, \quad \text{(# of items per stimulus)} \tag{10}
\]
\[
\sum_{i \in U_s} x_{i \leq n_s^u} z_s, \quad s = 1, \ldots, S, \quad \text{(\# of items per stimulus)} \tag{11}
\]
\[
\sum_{i \in C^l_e} x_{i \geq n_e^l}, \quad e = 1, \ldots, E, \quad \text{(categorical attribute)} \tag{12}
\]
\[
\sum_{i \in C^u_e} x_{i \leq n_e^u}, \quad e = 1, \ldots, E, \quad \text{(categorical attribute)} \tag{13}
\]
\[
\sum_{i=1}^{I} q_i x_i \geq q^l, \quad \text{(quantitative attribute)} \tag{14}
\]
\[
\sum_{i=1}^{I} q_i x_i \leq q^u, \quad \text{(quantitative attribute)} \tag{15}
\]
\[
\sum_{i \in C^l_f} z_s \geq n_f^l, \quad f = 1, \ldots, F, \quad \text{(categorical attribute)} \tag{16}
\]
\[
\sum_{i \in C^u_f} z_s \leq n_f^u, \quad f = 1, \ldots, F, \quad \text{(categorical attribute)} \tag{17}
\]
\[
\sum_{i=1}^{I} q_s z_s \geq q^s_l, \quad \text{(quantitative attribute)} \tag{18}
\]
\[
\sum_{i=1}^{I} q_s z_s \leq q^s_u, \quad \text{(quantitative attribute)} \tag{19}
\]
\[
\sum_{i \in V^l_g} x_i \leq 1, \quad g = 1, \ldots, G, \quad \text{(mutually exclusive items)} \tag{20}
\]
\[
\sum_{s \in V^l_h} z_s \leq 1, \quad h = 1, \ldots, H, \quad \text{(mutually exclusive stimuli)} \tag{21}
\]
\[
x_i = 0, 1, \quad i = 1, \ldots, I, \quad \text{(domain of variables)} \tag{22}
\]
\[ z_s = 0, 1, \quad s = 1, \ldots, S. \quad \text{(domain of variables)} \] (23)

The constraints in (9) fix the values of the decision variables of all \( k \)-1 items already administered to 1. In doing so, the model automatically accounts for the attributes of all these items (Step 6 in the pseudo-algorithm). The constraints in (10)-(11) serve a double goal. On the one hand, they constrain the number of items per stimulus. On the other hand, the variables for the stimuli in the right-hand sides of these constraints keep the selection of the items and stimuli consistent. It is only possible to assign an item from a set to the shadow test if its stimulus is assigned and, reversely, a stimulus cannot be assigned to a shadow test without assigning a permitted number of its items. The other constraints were already explained above.

4.3. Numerical Aspects

Commercial software for solving models as in (6)-(23) is amply available. This software can be used to calculate optimal values for the variables \( x_i, i = 1, \ldots, I, \) and \( z_s, s = 1, \ldots, S. \) Exact solutions to 0-1 LP problems can only be obtained through explicit enumeration, preferably in the form of an implementation of the branch-and-bound (BAB) method. 0-1 LP problems are known to be NP-hard, that is, their solution time is not bounded by a polynomial in the size of the problem. Therefore, special implementations of the method for larger problems are needed. The following ideas have been implemented successfully.

First, note that the constraints in (7)-(23) do not depend on the value of \( \hat{\theta} \). The update of this estimate only affects the objective function in (6). Repeated application of the model for \( k = 1, \ldots, n \) can thus be described as a series of problems in which the space of feasible solutions remains the same but the coefficients in the objective function (i.e., the values for the item information function) change. The changes become small if the ability estimates stabilize. Because solutions can be found much quicker if the algorithm starts from a good initial feasible solution, the obvious choice is to use the \((k - 1)\)th shadow test as the initial solution for the \( k \)th test. This measure has been proven to improve the speed of the solution processes dramatically. The first shadow test need not be calculated in real time. It can be preassembled for the initial estimates of \( \theta \) before the adaptive test becomes operational.

Second, additional improvements include specifying the order in which the variables are branched on. The variables for the items, \( x_i \), determine the selection of individual items whereas those for the stimuli, \( z_s \), determine the selection of larger sets of items. It always pays off to start branching on the variables that have the largest impact.
Therefore, branching on the stimulus variables should precede branching on the item variables. Also, forcing the slack variables to be integer for the constraints in the model has proven to be an efficient measure; that is, the variables used by the LP program to turn the inequalities in the model into equalities. In the branching order, slack variables for constraints with stimuli (items) should have higher priority than the decision variables for these stimuli (items). For technical details on optimal branching in BAB methods for solving 0-1 LP problems in constrained adaptive testing, see Veldkamp (submitted).

It is a common experience with BAB processes that values close to optimality are found long before the end of the process. If a good upper bound to the value of the objective function is available, it therefore makes sense to stop the process as soon as the objective function approaches the bound satisfactorily closely. Good results have been found for tolerances as small as 1-2% of the value of the upper bound.

All applications later in this chapter used CAT software developed at the University of Twente. To calculate the shadow tests, the software made calls to either the ConTEST package for 0-1 LP based test assembly (Timminga, van der Linden & Schweizer, 1997) or the solver in the CPLEX 6.0 package (ILOG, 1998). In most examples below, the CPU time needed for one cycle of ability estimation and item selection was 2-3 seconds. The only exceptions were for two adaptive tests in Application 3, which took 5-6 and 7-8 seconds per items, respectively and one in Application 4, for which 6-8 seconds per items were needed (all results on a Pentium Pro/166 MHz processor). All these tests had item sets with common stimuli; it is a general experience that solving 0-1 LP problems for tests with items sets tend to take more time (van der Linden, 2000a).

These CPU times are suitable for real-life application of the proposed CAT method. In fact, much larger times would not have involved any problem since it is always possible to calculate ahead. When the examinee works on item \( k - 1 \), the computer can already calculate two solutions for the \( k \)th shadow test, one for the update of \( \theta \) after a correct and the other after an incorrect response to item \( k - 1 \).

5. Four Applications to Adaptive Testing Problems

The principle underlying the adaptive testing scheme in this chapter is that every feature an adaptive test is required to have implies a set of constraints to be imposed on the item selection process. The constraints are implemented through the assembly of a shadow test prior to the selection of the item. As shadow tests are full-size linear tests,
in principle any feature possible for linear tests can also be realized for an adaptive test. The only thing needed is the formulation of the constraints as a set of linear (in)equalities for insertion into the model in (6)-(23).

The principle is illustrated with four applications each addressing a different aspect of adaptive testing. In the first application, the practicality of the shadow test approach is assessed for a CAT program with an extremely large number of content constraints of varying nature. The second application deals with the problem of differential speededness in CAT. Because each examinee gets a different selection of items, some of them may have trouble completing the test. It is shown how the problem can be solved by inserting a response-time constraint in the shadow test. The question of how to deal with item-exposure control in constrained adaptive testing with shadow tests is addressed in the third example. The method of control is based on Stocking and Lewis' (1998) conditional approach but uses the shadow test to define the list from which the items are sampled for administration. Also, sampling is with replacement. The last example is relevant to the case of a testing program that offers its examinees a choice between an adaptive and a linear version of the same test. It is shown how the observed-scores on the two versions of the test can be automatically equated by inserting a few additional constraints into the model for the shadow test. Each of these applications provides one or more empirical examples using a data set from an existing CAT program.

5.1. CAT with Large Numbers of Nonstatistical Constraints

To check the practicability of a shadow test approach for a CAT program with a large number of nonstatistical constraints, a simulation study was conducted for a pool of 753 items from the Law School Admission Test (LSAT). A 50-item adaptive version of the LSAT was simulated. The current linear version of the LSAT is twice as long; all its specification were therefore reduced to half their size. The specifications dealt with such item and stimulus attributes as item and stimulus content, gender and minority orientation, word counts, and answer key distributions. The set of content attributes defined an elaborate classification system for the items and stimuli for which the test had to meet a large number of specifications. In all, the LP model for the shadow test had 804 variables and 433 constraints.

Three conditions were simulated: (1) unconstrained CAT (adaptive version of the LSAT ignoring all current specifications); (2) constrained CAT with the least severely constrained section of the LSAT first; and
(3) constrained CAT with the most severely constrained section first. Mean-squared error (MSE) and bias functions were calculated for each three conditions after $n = 10, 20, ..., 40$ items.

The results are shown in Figure 2 and 3. For all test lengths, the results for the conditions of unconstrained CAT and constrained CAT with the least severely constrained section first were practically indistinguishable. The MSE and bias functions for the condition of constrained CAT with the most severely constrained first were less favorable for the shorter test lengths but matched those of the other two conditions for $n > 20$. The results are discussed in more detail in van der Linden and Reese (1998). The main conclusion from the study is that adding large numbers of constraints to an adaptive test is possible without substantial loss in statistical precision.
Figure 3. Bias functions after \( n = 10, 20, 30, \) and 40 items (Condition 1: dotted; Condition 2: dashed; Condition 3: solid)

5.2. CAT with Response-Time Constraints

Another problem not anticipated before adaptive testing became operational deals with the timing of the test. In the current CAT programs, examinees are typically free to sign up for a time slot in the period of their choice but the length of the slots is constant. Test questions, however, require different amounts of time to complete. Such differences are due to the amount of reading involved in the item or the difficulty of the problem formulated in it. Because each examinee gets an individual selection of items, some examinees may run out of time whereas others are able to finish the test easily in time.

The requirement that adaptive tests should have items that enable each examinee to finish in time is another example of a constraint to be imposed on the assembly of the shadow tests. A constraint to this effect was already proposed in one of the early papers on applying the technique of 0-1 LP to assembling linear tests:
where \( t_i \) is the amount of time needed for item \( i \), estimated, for instance, as the 95th percentile in the distribution of response times recorded in the pretest of the item, and \( t_{tot} \) is the total amount of time available (van der Linden & Timminga, 1989).

A more sophisticated approach is followed in van der Linden, Scrams and Schnipke (1999). Since examinees are also known to show individual variation in response times, they suggest using a lognormal model for the distribution of the response times of an examinee responding to an item with separate parameters for the examinee and the item. They also discuss how the values of the item parameter can be estimated from pretest data and how estimates of the value of the examinee parameter can be updated using the actual response times during the test. A Bayesian procedure is proposed for these updates as well as for obtaining the posterior predictive densities for the response times for the examinee on the remaining items in the pool. These densities are used to establish the constraint that controls the time needed for the test.

Suppose \( k - 1 \) items have been administered. At this point, both the actual response times for the examinee on the items in \( S_{k-1} \) and the predictive densities for the remaining items in \( R_k \) are known. Let \( t_{ij}^{\alpha_k} \) be the \( \alpha_k \)th percentile in the posterior predictive density for item \( i \in R_k \). The percentile is formulated to be dependent on \( k \) because it makes sense to choose more liberal values of \( \alpha \) in the beginning of the test but more conservative ones towards the end.

The following constraint controls the time needed for the test:

\[
\sum_{i \in S_{k-1}} t_{ij} x_i + \sum_{i \in R_k} t_{ij}^{\alpha_k} x_i \leq t_{tot}. \tag{25}
\]

Inserting the constraint in the model in (6)-(23) forces the new items in the shadow test to be selected such that the sum of their predicted response times does not exceed the remaining amount of time available.

The procedure was applied to an item pool for the CAT version of the Arithmetic Reasoning Test in the Armed Services Vocational Aptitude Battery (ASVAB). The pool consisted of 186 items calibrated under the model in (1). Response times had been recorded for 38,357 examinees who had taken the test previously. The test had a length of 15 items for which \( t_{tot} = 39 \) minutes. Percentile \( \alpha_k \) was chosen to be the 50th percentile for \( k = 1 \) and moved up in equal steps to the 95th percentile for the last items.
Figure 4. Time left after completion of the test (a) without and (b) with response time constraints

To evaluate the effects of the constraint in (27) on the time needed by the examinees, two different conditions were simulated: (1) the current version of the ASVAB test and (2) a version of the test with the constraint. The results are given in Figure 4. The first panel shows the average time left after completion of the test as a function of the slowness of the examinee, $\tau$, for the condition without the constraint. Different curves are displayed for the different values of $\theta$ used in the study. The faster examinees had still left some time after the test but the slower examinees ran out of time. The second panel shows the same information for the condition with the constraint. The left-hand side of the curves remain the same but for $\tau \geq -0.30$ the constraints became active. The lower curves run now more horizontally and none of them crosses the dotted line representing the time limit. The effects of the constraints on the bias and MSE in the estimator of $\theta$ were also estimated. These effects were of the same small order as for the LSAT in Application 1. For these and other results from this simulation study, see van der Linden, Scrams, and Schnipke (1999).

An interesting phenomenon was discovered during the simulations. In the first panel in Figure 4, the curves are ordered uniformly by the values of the ability parameter examinees, $\theta$. However, the order was counter to the intuitions of the authors who had expected to see the examinees with a low ability and not those with a high ability suffer from the time limit. Additional analyses of the data set revealed that there was no correlation between ability and speed (.04). Thus, among the more able examinees there were as many slow as fast examinees. On the other hand, as expected, a strong positive correlation (.65) was found between the item difficulty parameter and the parameter for the
amount of time required by the item. Because the test was adaptive, the more able examinees received the more difficult items, particularly towards the end of the test. Consequently, the more able examinees responding more slowly had trouble completing the test.

5.3. CAT with Item-Exposure Control

Because in CAT items are selected to maximize the information measure in (1), variations in the information in the items can have a large impact on their probability of selection. Items with a maximum contribution at some values of $\theta$ are chosen each time the ability estimate is updated to these values. Items that have no maximum contribution for any value of $\theta$ are never chosen at all.

This process has two undesired consequences. First, items with a high probability of being selected are frequently exposed and run the danger, therefore, of becoming known to the examinees. Second, items with a low probability of selection represent a loss of resources. They have gone through a long process of reviewing and pretesting and, even if only slightly less than optimal, do not return much of these investments.

To prevent from security problems, CAT programs usually constrain their item selection process to yield item exposure rates not larger than certain target values. As a result, the exposure rates of popular items go down and those of some of the less popular ones go up. A review of the existing methods of item-exposure control is given in Stocking and Lewis (this volume).

Sympson and Hetter introduced the idea of having an additional probability experiment determine if items selected are actually administered or removed from the pool for the examinee. Stocking and Lewis (1998) proposed a conditional version of this experiment. Let $P_i(S \mid \theta)$ denote the probability of selecting item $i$ conditional on the ability $\theta$ and $A$ the event of administering the item. It holds that:

$$P_i(A \mid \theta) = P_i(A, S \mid \theta) = P_i(A \mid S, \theta)P_i(S \mid \theta).$$

(26)

The exposure rates $P_i(A \mid \theta)$ are controlled by setting the values $P_i(A \mid S, \theta)$ at appropriate levels. However, the probabilities of selecting the items, $P_i(S \mid \theta)$, can only be determined by simulating adaptive test administrations from the item pool. Moreover, the event of selecting item $i$ is dependent on the presence of the other items in the pool. Therefore, the values of the control parameters can be set only through an iterative process with cycles of: (1) simulating the test; (2) estimating the probabilities of selection; and (3) adjusting the values for the control parameters.
In the original version of the Sympon-Hetter method, the probability experiment with the control parameters is repeated until an item is administered. In Stocking and Lewis' modification of the experiment, only one experiment is needed. In both methods, the experiments are based on sampling without replacement; once an item is passed, it is not replaced in the pool for the examinee.

The method of item-exposure control for the adaptive testing scheme in this chapter is the following modification of the Stocking-Lewis experiment: First, for the \( k \)th shadow test, the experiment is always run over the list of eligible items, \( A_k \). The list is ordered by the values of the items for their information functions at \( \hat{\theta} \). The list does not have a fixed length but decreases as the number of items administered increases. Second, the experiment is based on sampling with replacement. All items on the list passed are of better quality than the one actually administered. Not returning such items to the pool would result in a lower optimal value of the objective function for the later shadow tests, or, in the worst case, even in the impossibility to assemble them. On the other hand, the only consequence of replacing the items in the pool is higher probabilities of selection for the best items and lower values for the control parameters to compensate for the increase.

The effects of the proposed method of exposure control on the bias and MSE functions of the ability estimator were assessed for adaptive tests from item pools from three different adaptive tests programs: the Graduate Management Admission Test (GMAT), the Graduate Record Examination (GRE), and the PRAXIS test. Two conditions were simulated: (1) constrained CAT without exposure control and (2) constrained CAT with exposure control. Quantitative details on the models for these tests as well as the target values for the exposure rates used, the number of steps required to find the optimal values for the control parameter, the maximum exposure rate in the item pool realized, and the number of items with exposure rates violating the target are given in Table 1.

<table>
<thead>
<tr>
<th>Test</th>
<th># of Variables</th>
<th># of Constraints</th>
<th>Target Value</th>
<th># of Steps</th>
<th>Maximum Value</th>
<th># of Violations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>397</td>
<td>78</td>
<td>.29</td>
<td>4</td>
<td>.29</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>534</td>
<td>140</td>
<td>.22</td>
<td>5</td>
<td>.26</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>900</td>
<td>139</td>
<td>.20</td>
<td>6</td>
<td>.21</td>
<td>2</td>
</tr>
</tbody>
</table>
Figure 5. Differences between MSE and bias functions for CAT and without exposure control (Test 1: solid; Test 2: dotted; Test 3: dashed)

For two of these tests a few items in the pool had exposure rates larger than the target; however, both the number and sizes of the violations were low. The number of steps required to find the values for the control parameters were typical (see Stocking & Lewis (1998). In Figure 5, the differences between the bias and MSE functions for the ability estimators in the three tests are plotted. The general conclusion is that these differences are unsystematic and negligibly small. Adding the above method of exposure control to the procedure for constrained adaptive testing with shadow tests did not entail any loss of statistical precision in these examples.

5.4. CAT with Equated Number-Correct Scores

Testing programs making the transition to an adaptive testing format often want their examinees to have the choice between the former linear version and the new adaptive version of the test. However, this choice is only justified if the scores on both tests are comparable. To achieve comparable scores, the method of equipercentile equating has been applied to equate ability estimates on the adaptive version of the test to the number-right scores on the paper-and-for version (Segall, 1997).

The logic of constrained adaptive testing with shadow tests proposed in this chapter suggests the search for constraints that yield an adaptive test with observed scores automatically equated to those on the linear test. Such constraints are possible using the condition for two tests to have the same conditional number-correct score distributions given \( \theta \) presented in van der Linden and Luecht (1998). They show that, for any value of \( \theta \), the conditional distributions of observed number-
correct scores on two test forms with items \( i = 1, \ldots, n \) and \( j = 1, \ldots, n \) are identical if and only if

\[
\sum_{i=1}^{n} P_i^r(\theta) = \sum_{j=1}^{n} P_j^r(\theta), \quad r = 1, \ldots, n. \tag{27}
\]

They also show that for \( r \to n \) the conditions become quickly negligible and report nearly perfect empirical results for \( r = 2 \) or 3. Note that the conditions in (27) are linear in the items. Thus, the formulation of constraints for insertion in a 0-1 LP model for the assembly of shadow tests is possible.

Let \( j = 1, \ldots, n \) be the reference test which the adaptive test has to be equated to. The observed number-correct scores on the adaptive test can be equated to those on the reference test by inserting the following set of constraints in the model in (6)-(23):

\[
\sum_{i=1}^{n} P_i^r(\hat{\theta}_{k-1}) x_i - \sum_{j=1}^{n} P_j^r(\hat{\theta}_{k-1}) \leq c, \quad r = 1, \ldots, R, \quad \leq n \tag{28}
\]

\[
\sum_{i=1}^{n} P_i^r(\hat{\theta}_{k-1}) x_i - \sum_{j=1}^{n} P_j^r(\hat{\theta}_{k-1}) \geq -c \quad r = 1, \ldots, R \leq n \tag{29}
\]

where \( c \) is a tolerance parameter with an arbitrarily small value and \( R \) need not be larger than 3 or 4, say. Note that these constraints thus require the difference between the sums of powers of the response functions at \( \hat{\theta}_{k-1} \) to be in an arbitrarily small interval about zero, \((-c, c)\). They do not require the two sets of response functions to be identical across the whole range of \( \theta \). They also require only that sums of powers of their values at \( \hat{\theta}_{k-1} \) be identical, not the powers of the response functions of the individual items. Thus, the algorithm does not build adaptive tests that are required to be item-by-item parallel to the alternative test.

To assess the effects of the constraints in (28)-(29) on the observed number-scores in the adaptive test as well as on the statistical properties of its ability estimator, a simulation study was conducted for the same item pool from the LSAT as in Application 1. The conditions compared were: (1) unconstrained CAT and (2) constrained CAT with the above conditions for \( R = 1, 2 \). In either condition, the true values of the examinees were sampled from \( N(0, 1) \). The observed-number correct scores were recorded after \( n = 20, \ldots, 50 \) items. As a reference test, a previous form of the LSAT was used.
Figure 6. Observed-score distributions for CAT with and without constraints for number-correct score equating (target distribution: dotted; CAT without constraints: solid; CAT with 1 constraint: dashed-dotted; CAT with 2 constraints: dashed)

The results are given in Figure 6. As expected, the observed number-correct distribution for the unconstrained CAT was peaked with a mode slightly larger than n/2. After 20 items the observed number-correct distributions for the constrained condition had already moved away from this distribution towards the target distribution on the reference
test. After 30 items, the observed number–correct distributions for the constrained CAT and the reference test were indistinguishable for all practical purposes. The choice of value for $R$ did not seem to matter much. The ability estimators in both conditions did not show any differences in the bias function but a small loss in MSE for the constrained CAT condition. The loss was comparable to the one for one of the constrained CAT conditions in Figure 2 (Application 1). A more detailed discussion of the results and other applications of the constraints in (29)-(30) is given in van der Linden (2000b).

6. Concluding Remark

The empirical examples above illustrate the application of several types of constraints. These examples do not exhaust all possibilities. A more recent example is the formulation of Chang and Ying’s (1999) $\alpha$-stratified multistage adaptive testing scheme in the current framework to allow for large numbers of content constraints on the adaptive test (van der Linden & Chang, submitted). Also, because the adaptive test is realized through a series of shadow tests, specifications can be imposed in ways that do not exist for the assembly of a single linear test. These new implementations include the possibility of alternating systematically between objective functions for successive shadow tests to deal with cases of multiple-objective test assembly or using models with stochastic constraints (that is, constraints randomly sampled from a set of alternatives or with randomly sampled values for their coefficients). However, applications of such implementations to constrained adaptive testing still have to be explored.

References


van der Linden, W. J. & Chang, H.-H. (submitted) Alpha-stratified multistage adaptive testing with large numbers of content constraints.