A Comparison of Item-Selection Methods for Adaptive Tests with Content Constraints

Wim J. van der Linden
University of Twente

In test assembly, a fundamental difference exists between algorithms that select a test sequentially or simultaneously. Sequential assembly allows us to optimize an objective function at the examinee’s ability estimate, such as the test information function in computerized adaptive testing. But it leads to the non-trivial problem of how to realize a set of content constraints on the test—a problem more naturally solved by a simultaneous item-selection method. Three main item-selection methods in adaptive testing offer solutions to this dilemma. The spiraling method moves item selection across categories of items in the pool proportionally to the numbers needed from them. Item selection by the weighted-deviations method (WDM) and the shadow test approach (STA) is based on projections of the future consequences of selecting an item. These two methods differ in that the former calculates a projection of a weighted sum of the attributes of the eventual test and the latter a projection of the test itself. The pros and cons of these methods are analyzed. An empirical comparison between the WDM and STA was conducted for an adaptive version of the Law School Admission Test (LSAT), which showed equally good item-exposure rates but violations of some of the constraints and larger bias and inaccuracy of the ability estimator for the WDM.

The two primary requirements test forms in a standardized testing program have to meet are similar content for each examinee and reliable estimates of their abilities. To satisfy the first requirement, typically a set of content specifications is established and the forms are assembled according to these specifications. To satisfy the second requirement, the forms are assembled to be maximally informative on the examinees’ abilities.

For testing programs with a paper-and-pencil format, the first requirement does not involve any insurmountable problems. These programs usually assemble their test forms well in advance and have large items pools that can be searched for combinations of items meeting the set of specifications. Due to developments in integer programming for test assembly, the search can be automated and an optimal combination of items is now found in less than a few seconds (van der Linden, 1998, 2005). However, for paper-and-pencil testing, guaranteeing scores with high levels of information on the ability of each individual examinee is more difficult. Because each item has its maximum information at a different ability level, the test has to be assembled compromising between the ability levels of the examinees in the population.

A computerized adaptive testing format is ideal to realize the second requirement. In this format, the examinee’s ability estimate is updated after each new response
and the next item is selected to have maximum information at the updated estimate. However, for adaptive testing the requirement to give each examinee a test with a similar content is more difficult to meet. It can only be realized if the item-selection algorithm is forced to combine the objective of maximum information with a strategy that imposes the same set of content constraints on the selection for each examinee.

At a more formal level, the comparison between these two testing formats reveals a fundamental choice in test assembly, namely, between a simultaneous and sequential method of item selection. If we want the test to optimize an objective function such as the one of maximum information at the examinee’s ability level, \textit{sequential} test assembly can be used. But if the test has to meet a set of content constraints, the items should be selected \textit{simultaneously}.

The following fictitious example, adapted from van der Linden (2005, sect 4.6), provides an illustration of the dilemma. Suppose we have to assemble a test from the pool of five items in Table 1. The test has to be maximally informative at ability level $\theta$. Besides, it has to meet the following constraints:

1. Test length equal to two;
2. At least two items with attribute $A$;
3. At most one item with attribute $B$;
4. No items that are “enemies” of one another.

$V_A$ and $V_B$ are the sets of items in the pool with attributes $A$ and $B$. Enemies are items that cannot be in the same test because one of them has a clue to the solution of the others. $V_E$ is the set of enemies in the pool. Membership of the three sets is indicated by + and −.

Sequential item-selection algorithms pick one item at a time. If the objective is to maximize the information in the test, a sequential algorithm would pick item 1 first. This choice is not against any of the constraints but does have consequences for the choice of the next item. In fact, once item 1 is selected it becomes impossible to select another item without violating some of the constraints. As for the information in the test, the next best choice is item 2, but this choice would violate the fourth constraint. A choice of item 3, 4, or 5 would not only mean violation of one or more of the constraints, but also a larger loss of information. But Table 1 does have two feasible tests, one consisting of item 2 and 3 and the other of item 2 and 5. The former is optimal. This optimum can only be guaranteed to be found by a

---

**TABLE 1**

<table>
<thead>
<tr>
<th>Item</th>
<th>$I(\theta)$</th>
<th>$V_A$</th>
<th>$V_B$</th>
<th>$V_E$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.44</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>2</td>
<td>0.38</td>
<td>+</td>
<td>−</td>
<td>+</td>
</tr>
<tr>
<td>3</td>
<td>0.31</td>
<td>+</td>
<td>+</td>
<td>−</td>
</tr>
<tr>
<td>4</td>
<td>0.17</td>
<td>−</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>5</td>
<td>0.12</td>
<td>+</td>
<td>+</td>
<td>−</td>
</tr>
</tbody>
</table>

---
A Comparison of Item-Selection Methods

Simultaneous algorithms do not select individual items but move from one combination of $n$ items to another until one is found that is both feasible and optimal.

Efficient algorithms for the simultaneous selection of tests do not check every possible combination of items blindly but avoid subsets that would lead to infeasible tests or tests known to be less than optimal. If such a subset is hit, these algorithms move back, and begin their search from a new combination. The critical difference between simultaneous and sequential algorithms is this feature of backtracking. Simultaneous algorithms can undo earlier choices of items if they appear to lead to combinations with unfavorable properties; sequential algorithms have to stick to them.

In computer adaptive testing (CAT), items are selected in real time, and once an item is administered its choice cannot be undone. Item selection in CAT is thus sequential by nature. Three different approaches to item-selection for CAT have been proposed to deal with its inherent inability to use backtracking. One of the approaches is based on the partitioning of the item pool implied by the key attributes of the items (Kingsbury & Zara, 1991). The algorithm enforces the content specifications by spiraling the selection of the items over the classes in this partition proportionally to the numbers needed from them. The other two methods are the weighted-deviations methods (WDM) (Stocking & Swanson, 1993) and the shadow-test approach (STA) (van der Linden, 2000; 2005, chap. 9; van der Linden & Reese, 1998). These methods compensate for the lack of backtracking by looking forward. They calculate projections of the future consequences of the selection of the items and pick the item with the best projections. As will be shown below, these two methods differ, however, in the nature of the projections they use.

It is the goal of this article to analyze the pros and cons of these three entirely different methods of item selection in CAT. In addition, we evaluate their performances empirically for an adaptive test from a former item pool of the Law School Admissions Test (LSAT). For an empirical case study with an adaptive test in use at Educational Testing Service (ETS), see Robin, van der Linden, Eignor, Steffen, and Stocking (2004). The differences between this case study and the empirical evaluation in this article are explained below.

Before embarking on our analyses and evaluations, we first introduce a menu of the possible test specifications adaptive tests have to meet in practice.

Review of Possible CAT Specifications

A precise way to formulate a test assembly problem is as an optimization model formulated for the decision variables needed to select the items. We present the skeleton of such a model, which has the objective function and the main menu of constraints from which we have to choose if we formulate test specifications for a CAT program. Some programs offer their examinees sets of items organized around common stimuli, for example, reading passages or descriptions of cases. Typically, if a stimulus is selected by the algorithm, the number of items selected along with it has to be between certain bounds but smaller than the set available in the item pool. We formulate the model for the general case of a CAT with a mixed format of
Suppose the item pool has $S - 1$ item sets. We denote the stimuli in these sets as $s = 1, \ldots, S - 1$. In addition, the pool may have a larger set of discrete items, that we identify by a dummy stimulus $s = S$. The items nested under stimulus $s$ are denoted as $i_s = 1, \ldots, I_s$. We use two different sets of 0–1 decision variables for the selection of the stimuli and items. The first set contains the variables $z_s$ that take the value $z_s = 1$ if stimulus $s$ is selected and $z_s = 0$ if it is not. The second contains the variables $x_{is}$ that take the value $x_{is} = 1$ if item $i_s$ is selected and $x_{is} = 0$ if it is not. Of course, the selection of stimuli and items has to be coordinated, and we will do so by adopting special logical constraints in the model. (Logical constraints are constraints needed to implement logical operations in the test assembly, such as “if-then” selection.)

Both the items and stimuli can have quantitative and categorical attributes. Examples of quantitative attributes are item-response theory (IRT) or classical item parameter, word counts, and reading times. We use $q_{is}$ and $r_s$ as generic symbols for the quantitative attributes of item $i_s$ and stimulus $s$. A special quantitative attribute is the value of the information function for the items for a given examinee under the item-response theory IRT model (Lord, 1980) used to calibrate the item pool; for item $i_s$ and ability level $\theta$ we denote this value as $I_{is}(\theta)$. Examples of categorical attributes are content category, cognitive level, and item format. We denote subsets of items in the pool that share a common categorical attribute as $V_c, c = 1, \ldots, C$; subsets of stimuli with a common attribute are denoted as sets $V_d, d = 1, \ldots, D$.

If we knew the ability level of the examinee, the test assembly problem would have the following general form:

$$\begin{align*}
\text{optimize } & \sum_{s=1}^{S} \sum_{i=1}^{I_s} I_{is}(\theta)x_{is}, \quad \text{(maximum information)} \\
\text{subject to } & \\
& \sum_{s=1}^{S} \sum_{i=1}^{I_s} x_{is} \geq n, \quad \text{(number of items)} \\
& \sum_{s=1}^{S-1} z_s \geq m, \quad \text{(number of item sets)} \\
& \sum_{s=1}^{S} \sum_{i_s \in V_c} x_{is} \geq n_c, \quad \text{for all } c, \quad \text{(categorical item attribute)} \\
& \sum_{s=1}^{S} \sum_{i_s = 1}^{I_s} q_{is} x_{is} \geq b_q, \quad \text{(quantitative item attribute)} \\
& \sum_{s=1}^{S} \sum_{i_s \in V_e} x_{is} \leq 1, \quad \text{for all } e, \quad \text{(enemy items)}
\end{align*}$$
A Comparison of Item-Selection Methods

\[ \sum_{i_s=1}^{l_s} x_{i_s} \geq n_s z_s, \quad \text{for all } s, \quad \text{(number of items per set)} \]  

(7)

\[ \sum_{s \in V_d} z_s \geq n_d, \quad \text{for all } d, \quad \text{(categorical stimulus attribute)} \]  

(8)

\[ \sum_{s=1}^{S-1} r_s z_s \geq b_r, \quad \text{(quantitative stimulus attribute)} \]  

(9)

\[ x_{i_s} \in \{0, 1\}, \quad \text{for all } i \text{ and } s, \quad \text{(range of variables)} \]  

(10)

\[ z_s \in \{0, 1\}, \quad \text{for all } s. \quad \text{(range of variables)} \]  

(11)

The objective in (1) is maximization of the information in the test at \( \theta \). Possible constraints on this maximization are formulated in (2)–(11). In these constraints, \( \geq \) represents an appropriate selection of an inequality or the equality sign. It is possible to include both an upper and a lower bound on the same test attribute in the model. The constraints in (2) and (3) are on the numbers of discrete items and the number of item sets in the test, respectively. The constraints in (4)–(5) guarantee that the test has the required distribution of categorical item attributes and the required bounds on the quantitative item attributes. The sets \( V_e, e = 1, \ldots, E \) in the constraints in (6) are the sets of enemy items in the pool. Consequently, these constraints forbid the presence of more than one item in the test from each of these sets. The constraints in (7) impose bounds \( n_s \) on the number of items for stimulus \( s \) in the test. At the same time they introduce an “if-then” operation in the selection of the items and stimuli: It is easy to verify that the bound \( n_s \) can only be active if \( z_s = 1 \), that is, if stimulus \( s \) is selected; otherwise the right-hand side is equal to zero and no items are selected. Analogous to (4)–(5), the constraints in (8)–(9) guarantee that the test has the required distribution of categorical stimulus attributes and the required bounds on quantitative stimulus attributes. Finally, the constraints in (10)–(11) impose the appropriate range of values on the variables.

In an application of the model to a real-life test, the size of the constraint set can be large. In the empirical example below, the number of constraints needed to model the content specifications of the LSAT was equal to 105. For a more complete treatment of the technique of modeling test assembly specifications using integer programming, see van der Linden (2005).

If \( \theta \) would be known and the test be administered in paper-and-pencil format, the model could be solved simultaneously for an optimal test, that is, an optimal set of values for the decision variables \( z_s \) and \( x_{i_s} \) = the string of 0s and 1s that optimizes (1), by using a commercial integer solver, for example, the one in the linear-programming software package CPLEX 9.0 (ILOG, Inc., 2003).

In adaptive testing, \( \theta \) is estimated during the test. It is important to note that estimation of \( \theta \) during the test leads only to changes in the objective function in (1). The constraint set in (2)–(11) remains unchanged. A simultaneous algorithm is necessary to find a combination of items that satisfies this set, but the adaptive format entails sequential item selection. This is the dilemma discussed earlier. In the next
section we will see how three main item-selection methods for CAT deal with this dilemma.

Item-Selection Methods

Spiraling Item Selection

This method, which is due to Kingsbury and Zara (1991), is based on the observation that the content attributes of the items partition the item pool. This partitioning is immediately obvious for the sets of categorical item attributes $V_c, c = 1, \ldots, C$, in the constraints in (3). (How the method deals with the other constraints is addressed further on in this article.) The numbers of items from these sets, $n_c$, are the frequencies with which they should be visited.

For notational convenience, we ignore the possible item-set structure of the pool for a while and denote the items in the pool as $i = 1, \ldots, I$. The items in the test are denoted as $k = 1, \ldots, K$. It thus holds that $i_k$ is the index of the item in the pool selected as the $k$th item in the test. If item $k$ is to be selected, the set of items already administered is $T_{k-1} = \{i_1, \ldots, i_{k-1}\}$. Therefore, item $k$ has to be selected from the set $R_k$ defined as the set of items that are in $\{1, \ldots, I\}$ but not in $T_{k-1}$. During the test the number of times set $V_c$ has been visited is recorded. Let $n_c^{(k-1)}$ denoted the number of visits to $V_c$ prior to the selection of item $k$.

The algorithm proposed by Kingsbury and Zara is based on an application of the minimax principle over the sets $V_c$; the algorithm picks the set for which the largest number of items is still needed in the test. The first step of the algorithm is therefore to identify the set that has the maximum value in $\{n_1 - n_1^{(k-1)}, \ldots, n_C - n_C^{(k-1)}\}$. We denote this set as $V^{(k-1)}_c$. Item $k$ is then selected as the one in $V^{(k-1)}_c$ with maximum information at the current estimate $\hat{\theta}_{k-1}$, that is, the item that maximizes

$$I_{ik}(\hat{\theta}_{k-1}).$$

If enough items have been selected from a set, it is removed from the pool. The same strategy can be followed to deal with the sets of enemies, $V_e$, in (9). The algorithm then also records whether or not an item has been selected from these sets. As soon as an item is selected from a set, it is removed from the pool. To control the exposure rates of the items in the pool, Kingsbury and Zara suggest a method introduced by McBride and Martin (1983), which does use the criterion in (12) to select the best item in $V^{(k-1)}_c$ but involves random selection of an item from the best $m$ items in this set, where $m$ typically is decreased during the test.

This method thus imposes the content constraints by “brute force.” It does not look at future consequences when selecting an item, but simply picks the best item from a set when its turn has arrived according to the minimax principle. CAT studies in which this method has been used have been reported in Chen, Ankenmann, and Spray (1999, April) and Leung, Chang, Hau, and Wen (2001, April; 2003, April).

Weighted-Deviations Method

The WDM was introduced as a method for the assembly of linear tests in Swanson and Stocking (1993). The method selects the items sequentially. An application to
A Comparison of Item-Selection Methods

CAT, with additional updates of the interim estimate of \( \theta \) before each next item, is therefore a natural extension of the method (Stocking & Swanson, 1993).

The WDM was developed out of concern for possible poor quality of item pools in large-scale test assembly. Such pools may not have a feasible solution, i.e., one that meets all of the constraints. Because in such cases it is still desirable to produce a test, Swanson and Stocking proposed to change the nature of the constraints and interpret their bounds as goal values. This interpretation implies the following changes for the test assembly model in (1)–(11):

1. For each of the attributes in the test the bounds in the constraints in (2)–(9) are interpreted as goal values. In addition, goal values are formulated for the test information function in (1).
2. The difference between the goal values and the actual value of each test attribute is defined as an objective to be minimized. These objectives are only active as long as their goal values are not realized.
3. Because we now have as many objectives as relevant test attributes, the problem is replaced by one in which the function to be minimized is a weighted sum of these objectives.
4. The items are selected sequentially minimizing this weighted sum.

We denote the individual attributes in the constraints of the test assembly problem by \( h = 1, \ldots, H \). The contribution of item \( i \) to attribute \( h \) is represented by a generic symbol \( a_{ih} \). If attribute \( h \) is categorical, \( a_{ih} = 1 \) when item \( i \) has the attribute and \( a_{ih} = 0 \) otherwise. If attribute \( h \) is quantitative, \( a_{ih} = q_i \); for example, for the value of the item information function at \( \hat{\theta} \) we have \( a_{ih} = I_i(\hat{\theta}) \). If item \( k \) is to be selected, the algorithm calculates for each candidate item in \( R_k \), a projection of the value of attribute \( h \) for the eventual test. The projection consists of (1) the sum of the attribute values of the \( k - 1 \) items already administered, (2) the attribute value of the candidate item, and (3) \( n - k \) times the average attribute value for the other items in \( R_k \). In sum, the projection of the value of attribute \( h \) for the selection of candidate item \( i_k \) is

\[
\pi_{ikh} = \sum_{i \in S_{k-1}} a_{ih} + a_{ikh} + (n - k) \frac{\sum_{i \in R_k \setminus \{i_k\}} a_{ih}}{(l - k)} .
\]  

(13)

Let \( \gamma_h \) denote the goal value for attribute \( h \). The difference between this goal value and the projection in (13) upon selection of item \( i \) is \( \gamma_h - \pi_{ikh} \). The item selected is the one in \( R_k \) that minimizes the weighted sum of the absolute differences

\[
\sum_{h=1}^{H} w_h |\pi_{ikh} - \gamma_h| .
\]  

(14)

The weights \( w_h \) for the attributes are to be chosen by the testing organization.

Observe that the first term in (13) is constant across all candidate items, for a given \( k \). This term therefore plays no role in the relative ranking of the items in \( R_k \) with respect to attribute \( h \). Also, for an item pool of reasonable size, the last term in (13) is
robust. The reason why these two terms have been added to (13) is to get a projection of the attribute value for the eventual test on the actual scale of the attribute. To select an item, the WDM balances the extent to which the items improve these projections with the importance of their attributes as specified by the weights.

The WDM is an example of an approach to solving multiple-objective problems in linear programming that is more generally known as goal programming. For a review of this and other approaches to multiple-objective test assembly, see van der Linden (2005, sect. 3.3.4) or Veldkamp (1999). Several empirical examples of the use of the WDM in CAT are given in Stocking and Swanson (1993). An application of the same method to a linear test assembly problem is discussed in Stocking, Swanson, and Pearlman (1993), whereas Stocking and Swanson (1998) used the method for selecting a set of rotating CAT pools from a master pool.

Shadow-Test Approach

The STA (van der Linden, 2000; van der Linden & Reese, 1998) also performs projection-based item selection. It differs from the WDM in that these projections are not of a weighted combination of the attributes of the eventual test, but of a realization of the full test. These projections are known as shadow tests. Shadow tests are linear tests assembled prior to the selection of each item, which:

1. have maximum information at the current ability estimate,
2. meet all the constraints,
3. and contain all items already administered to the examinee.

The STA picks the item that has maximum information at the ability estimate among the free items in the shadow test for administration. Let $A_k$ denote the set of free items in the shadow test prior to the selection of item $k$. The item selected is the one in $A_k$ that maximizes

$$I_{ik}(\hat{\theta}_{k-1}). \quad (15)$$

Technically, the only thing the CAT algorithm has to do prior to the selection of item $k$ is to update the model for the CAT in (1)–(11) and make a call to an integer solver used to calculate the shadow tests. The updates consist of (1) substituting the last estimate $\hat{\theta}_{k-1}$ in the objective function of the model and (2) adjusting the following constraint added to the model to make sure that the items already administered are in the shadow test:

$$\sum_{i \in S_{k-1}} x_i = k - 1. \quad (16)$$

In the empirical example with the LSAT further on in this article, we used the integer solver in the software package CPLEX 9.0 (ILOG, Inc., 2003) to calculate the shadow tests. The CPU time needed to calculate a shadow test by this solver and select the next item was less than a second.
If the CAT proceeds, the part of the shadow test fixed by (16) increases and the number of free items decreases. This process is entirely analogous to the increase of the first term and decrease of the third term in the projections of the attributes in the WDM in (13). Both processes lead to a stabilization of the projections during the adaptive test. The critical difference between the two methods is only in the nature of their projections.

The STA solves the earlier dilemma between sequential and simultaneous selection of the $n$ items in the CAT by approaching the problem as a sequence of $n$ simultaneous problems. Because at each step the search is for an optimal feasible combination of items instead of a single item, the method avoids the violations of the constraints demonstrated in Table 1. Also, the method can be used to give a CAT any feature that traditionally has been given to a linear test, simply by inserting the appropriate constraints into the model for the shadow test.

An empirical example of the use of the STA is given in Li (2002, April). In addition, the method has been used in a series of studies to evaluate new types of constraints on shadow tests that give the adaptive test such features as alpha-stratified item selection (van der Linden & Chang, 2003), the removal of differential speededness (van der Linden, Scrams, & Schnipke, 1999), number-correct-score equating to a linear reference test (van der Linden, 2001), the possibility to deal with multidimensional abilities (Veldkamp & van der Linden, 2002), and constraining the item-exposure rates to be below target values (van der Linden & Veldkamp, 2004).

**Comparison Between Methods**

The most important differences between the three methods are in their underlying philosophies. Each method represents a different view of the nature of the test specifications, of how the relative importance of the attributes in these specifications should be expressed, and of the principle on which item selection should be based.

Both the spiraling method and the STA view test specifications as constraints that have to be imposed on the item selection. The only exception is with respect to the information function of the test, which is treated as an objective function that has to be maximized. If the test is assembled, the bounds in the constraints are realized first, and the remaining space is used to optimize the objective function. As for the value of the objective function, it is not bounded in any way; it will be the optimal value admitted by the item pool. According to the philosophy underlying the WDM, all test specifications should be treated as objectives. Therefore, bounds in specifications formulated as constraints should actually be treated as goal values. The test should be selected to be optimal with respect to these values. As long as a test is optimal, it is not a problem if it violates some of the original bounds that were redefined as goal values.

The two philosophies lead to different strategies for controlling the relative importance of the test attributes. In the spiraling method and the STA, the test attributes are dealt with by constraints, and their importance can be controlled by tightening or relaxing the bounds in these constraints. In the WDM, they are dealt with as an objective function, and their importance is controlled by the weights in the sum of objective functions in (14).
As for the nature of item selection, a critical difference exists between the spiraling method and both the WDM and STA. Item selection by the former is purely sequential. This method focuses on the selection of one item at a time, and blindly follows the minimax principle to identify the next set \( V_c \) from which the item has to be picked. On the other hand, the WDM and STA look forward; they are able to suspend the selection of a locally optimal item when it leads to an unfavorable projection for the test. The WDM does so in the framework of sequential optimization and the STA by solving a sequence of simultaneous optimization problems.

The choice between these philosophies is not inconsequential. It leads to different performances on the criteria we discuss in the next sections.

**Test-Information-Function Value**

Both the spiraling method and the STA maximize the value of the information function for the examinee subject to the constraints. The spiraling method picks the items with maximum information from the attribute sets. The minimax principle makes the algorithm move between these sets in an order that is basically determined by the numbers of items needed from each of them; the larger the number, the earlier and more frequently the set is visited. But sets from which larger numbers are needed are not necessarily composed of the most informative items for the majority of the examinees at the current \( \hat{\theta} \).

The STA picks the items with maximum information from the shadow tests. Unlike the attribute sets in the spiraling method, these tests are not fixed but each time reassembled to be optimal at the new ability estimate. Except for the unlikely case of an item pool with its informative items in the attribute set from which most items are needed, we therefore expect the test-information-function values realized by the STA to be more favorable than those by the spiraling method.

The values of the information function realized by the WDM depend mainly on goal values set for this function and the relative weight \( w_h \) in (14) given to it. It is therefore impossible to rank the WDM on this criterion relative to the spiraling method and STA. If the weights for all other constraints approach zero, the WDM can be expected to outperform the other two methods; it then effectively maximizes the value of the information function in an unconstrained fashion. If the weight for the information function approaches zero, the WDM will not optimize the value of this function at all. If all weights approach equality, the results depend entirely on the composition of the item pool and become unpredictable.

**Constraint Realization**

The three methods differ both in the type of constraints they can handle and in the degree to which they admit violations of them. In this section, we do not yet address constraints on item sets; these are discussed in the next section.

The spiraling method was basically designed to deal with the categorical item attributes in (3). Without further heuristic adjustments, the method is unable to deal with the other types of attributes in (2)–(9). Even for categorical attributes, the method can handle equality constraints but has difficulty with constraints that take the form of inequalities; the minimax principle no longer offers guidance, and other
rules on how frequently and in what order the attribute sets \( V_c \) should be visited become necessary. If some of the constraints are on quantitative attributes, the attributes need to be categorized before the method can be applied. Logical constraints, such as the “if-then” constraints in (7) or the special CAT constraints referred to above, require further adjustments of the spiraling method. The method will never violate equality constraints on categorical attributes, but for the other types of constraints violations are possible. For example, if the method is applied with categorized quantitative attributes, it may capitalize on items exclusively near the lower or upper category bounds and fail to satisfy inequality constraints.

The STA represents the other extreme on this criterion. It can deal with any type of constraint that can be formulated as a linear (in)equality in the decision variables. For a general classification of types and level of linear constraints that can occur in test assembly, see van der Linden (2005, chap. 3). Also, the STA produces a feasible solution for each examinee if the item pool contains at least one combination of items that meets the constraint set (van der Linden & Reese, 1998). This seems a reasonable requirement for a professionally constructed item pool, so in practice the method will never stop because it cannot meet a constraint.

The WDM can be used to implement any type of categorical or quantitative attribute for which a goal value is meaningful. But the method needs adjustments to deal with logical constraints; examples of such adjustments for pools with item sets are discussed in the next section.

The Achilles heel of the WDM is its susceptibility to constraint violation. The reason is the weighted-sum criterion in (14). This sum admits trade-offs between constraints, allowing the algorithm to capitalize on some of them while ignoring others. This susceptibility in fact illustrates a self-fulfilling prophecy: As indicated earlier, the WDM was introduced out of concern for infeasibility due to poor item pool quality. However, because it replaces the search for a feasible solution by the criterion in (14), item selection is likely to produce violations even when the pool contains feasible solutions.

**Item Sets**

The relative abilities of the three methods to deal with item sets reflect those for the previous criterion. The spiraling method has difficulties dealing with them, especially when the numbers of items selected per stimulus are not fixed but have to be between upper and lower bounds. But the STA can deal with any type of item-set problem; the only thing that has to be done to take them into account is to insert an appropriate selection from the constraint in (6)–(9) into the model for the shadow tests.

For the WDM, two alternative ways of implementing adaptive testing with item sets are possible. Stocking and Swanson (1993; see also Swanson & Stocking, 1993) suggest forming all possible subsets of items from each set and treating these, along with possible discrete items in the pool, as the elementary units of selection. These subsets are assigned summaries of the attributes of their items, for instance, a subset-information function instead of an item-information function. If a subset is selected, all items in it are administered. To implement this method, the projections in (13) need to be adjusted as follows:
1. The attributes of the items in the candidate subset in the second term of (13) are aggregated at the level of the subsets. If the attribute is categorical, the aggregate is just the number of items that share this attribute; if it is quantitative, it is equal to the sum of its values.

2. The attributes of the stimuli for the item sets are treated in (13) exactly as the attributes of the items in the original version of the WDM.

3. If the subset has \(m\) items, the last term in the projection in (13) becomes \(n - k - m + 1\) times the average over the remaining units in the pool weighted by the numbers of items in them.

An alternative implementation continues the selection of individual items. If an item from a set is selected, the method stays in the set until the lower bound on its size is passed. The items are then selected from the entire pool, but the set is removed as soon as an item outside it has been selected or its upper bound is passed. This alternative involves only a slight adjustment of the projections in (13). First, stimulus attributes are included in these projections as soon as an item from their set is selected. Second, the calculation of these projections is restricted to the remaining items in the set as long as we are in it and its lower bound is not passed.

In the comparative study later in this article, the item sets were relatively large and the bounds on them loose. As a result, the number of possible subsets from them was extremely large. We therefore used the second method. The same choice has been made by ETS in its adaptive testing programs.

**Item-Exposure Control**

Any of the known methods for item-exposure control can be used in combination with the three methods of item selection. Though Kingsbury and Zara recommended the McBride-Martin method of random item selection for the spiraling method, it can be used with other exposure control methods as well, for example, the Sympson-Hetter (1985, October) method or the conditional version thereof (Stocking & Lewis, 1998). The WDM is typically used in combination with the conditional version of the Sympson-Hetter method. The shadow test method has been used with the same conditional Sympson-Hetter method, as well as with an extension of it in which a double-length shadow test was assembled at each step to create a longer list of items from which the actual item for administration was randomly selected (for details, see Veldkamp & van der Linden, submitted). This extension was used in the study reported by Robin et al. (2004). In the empirical study with the adaptive version of the LSAT below, both for the WDM and STA a new item-exposure control method based on probabilistic item ineligibility constraints was used. In this method, presented in van der Linden and Veldkamp (2004), the target for the item-exposures rates is realized by constraining the items in and out of the pool with probabilities that are updated adaptively during the history of the pool. An advantage of this method is that it can be implemented on the fly and does not require the extensive computer simulations necessary to adjust the control parameters prior to the test in the Sympson-Hetter method.
**Cost Effectiveness**

Commercial software for the three methods of constrained adaptive testing is not available. Use of them entails the necessity to develop one’s own software. The costs of software development are not expected to differ much between the methods. The basic computer code needed for the spiraling method is to implement the minimax principle in (12), for the WDM to implement the criterion in (13)–(14), and for the STA to implement the maximum-information principle. In addition, for the STA a commercial integer solver has to be bought to select the shadow tests.

The returns on these investments differ largely though. The spiraling method has restricted applicability. If the test specifications contain other constraints than equality constraint on categorical attributes, the method needs heuristics adjustment. The adjustments can only be realized through modification of the computer code for the algorithm. The WDM has much larger applicability. Transition to a new set of test specifications or a change of item pool does not necessitate modification of the code but selection of a new set of weights for the attributes $w_h$ in (14). Though experience helps, optimal weights can only be found through a process of trial and error in which trial values for the weights are assumed and the effects assessed by computer simulation. For the STA, a new set of test specifications or a change of item pool does not entail the necessity of any new code or experimentation with weights. Because the model in (1)–(11) and the integer solver are separate entities, the only thing needed to implement a change is a modification of the constraint set in the input file for the integer solver.

**Empirical Comparison**

The purpose of this empirical study was to compare the methods for a new adaptive test with a realistic set of test specifications. We simulated an adaptive version of the current LSAT from a former item pool from this program. Because the set of specifications for the LSAT involved a rather diverse selection from the types of constraints possible in adaptive testing, we were unable to implement the spiraling method. A description of these constraints is given below. The study, therefore, had to be restricted to the WDM and STA.

In a previous study (Robin et al., 2004), the WDM and STA were compared for three different CAT programs at ETS. The item pools were former operational pools constructed by ETS so that the WDM was able to create CATs that satisfied the specifications of these programs. Of interest in this study was whether the STA would produce tests of comparable or better psychometric quality as those produced by WDM and at the same time make better use of the items available in pools. In the current study, the two methods were compared for a new CAT program. In addition, the constraint set of the LSAT was more diverse than those for the ETS tests, which consisted predominantly of inequality constraints on categorical attributes.

We compared the two methods for their bias and mean-squared error (MSE) in the ability estimator, the frequency of different types of constraint violations, and the exposure rates of the items. To assess the impact of test length, the results on these criteria were recorded after $n = 10, 20, \ldots, 50$ items.
Method

Item pool and CAT specifications. The LSAT pool consisted of 753 items calibrated under the three-parameter logistic model. The LSAT consists of three different sections. Two of these sections are set-based; the other section has discrete items. The length of the CAT was \( n = 50 \) items, with 26 items in the sections with the item sets and 24 items in the section with the discrete items. The length of the CAT was half the length of the current paper-and-pencil version of the LSAT. The LSAT specifications deal with such matters as an elaborate multiple-level content classifications of the items, a content classification of the stimuli, the size of the item sets, possible gender and minority orientation of the stimuli, enemy sets, word counts, and the answer key distribution. We used the full set of specifications for the CAT version; the only change was the halving of the bounds in the constraints to account for the reduction in test length. The total number of variables and constraints in the integer programming model for the LSAT was 804 variables and 105 constraints.

For both item-selection methods, 2,000 CAT administrations were simulated for \( \theta = -2.0, -1.5, \ldots, 2.0 \). The initial ability estimate, \( \hat{\theta}_0 \), was set equal to zero. During the test, \( \theta \) was estimated using the expected a priori (EAP) estimator with a uniform prior over \([-4.0, 4.0]\). We simulated both methods without and with item-exposure control. The method of exposure control was a conditional version of the method with probabilistic item ineligibility constraints in van der Linden and Veldkamp (2003). The exposure rates were controlled at the same values of \( \theta = -2.0, -1.5, \ldots, 2.0 \), and the upper bound for the rates was equal to .20. We used 1,000 administrations to adjust the eligibility probabilities in this exposure-control method.

Implementation of the two methods. For the STA, we used the integer solver in the linear-programming software package CPLEX referred to earlier to assemble the shadow tests. The items were selected from the shadow tests using the maximum-information principle in (1).

To implement the WDM, we asked test specialists from the LSAC to select the weights \( w_h \) in (14). All weights were specified on a scale from 0–100%. The majority of the weights were in the 60–100% range but for a few attributes the weights were as low as 2% and 10%. The test specialists found it generally easy to weigh the constraints but indicated for a few of them that they would have preferred using what amounts to a non-linear weighing procedure, that is, one with relatively larger penalties than those in (14) for large violations but smaller penalties for small ones.

The different attributes in the constraints were all on different scales. For example, the total word count for a section of the LSAT typically runs around 1,500, but the numbers of items in certain content categories is lower than 10. To make the formulation of relative weights for constraints meaningful, all attributes have to be on the same scale. For this reason, we removed the scale differences between the bounds in the constraints using the following procedure: If an attribute was constrained from one side only, we divided all coefficients and the bound in the constraint by the value of this single bound. If it was bounded from above and below, we divided the coefficients and bounds by the average value of the two bounds.
The STA maximizes the values of the information function of the test during item selection, while the WDM forces it to approach a set of goal values at a grid of $\theta$s. We ran the simulations with the STA first and used the average values for the information functions obtained for the three sections as goal values (lower bounds) for the WDM. By making this choice, a possible bias in the comparison between the performances of the two methods on the criterion of test information was removed.

The weight put on the goal values for the test information functions in the WDM by the test specialists from the LSAC was 98%.

**Results**

Estimates of the bias and MSE functions for the ability estimates by WDM and STA were calculated as the mean difference and mean-squared difference between the ability estimate and the true ability at the levels used in the simulation study. The results for the conditions without and with item-exposure control were virtually identical. Figure 1 gives the bias and MSE functions for the conditions with item-exposure control.

As expected, these functions decreased with the number of items administered. The results for the WDM showed a larger bias and a less uniform MSE than for the

![Bias and MSE functions for the WDM and STA for n = 10, 20, …, 50 with exposure control (curves closer to horizontal axis are for higher n).](image)
STA. The differences between the bias and MSE functions were particularly large for the smaller test lengths at the more extreme values of $\theta$, where they differed approximately by a factor equal to two.

Tables 2 and 3 present results on the violations of the constraints by the WDM. Table 2 is for the constraints on the content attributes of the test. For the condition without item-exposure control, the number of constraints violated at least once was between 4 and 7 across the $\theta$ values in this study. The average percentage of examinees for which at least one of these constraints was violated was between .8% and 4.6%. The constraints that were violated had average weights between 21.5 and 49.3.

For the condition with item-exposure control, the numbers were substantially larger. Table 3 shows the same types of statistics for the constraints on the information functions for the three sections of the LSAT. Both for the condition with and without item-exposure control, the constraint for each of the three sections was violated. The average percentage of examinees for which this happened was in the 30–50% range, with an outlier at $\theta = 1.5$ for the condition without item-exposure control.

### TABLE 2

**Violations of Content Constraints (WDM)**

<table>
<thead>
<tr>
<th>$\theta$</th>
<th>−2.0</th>
<th>−1.5</th>
<th>−1.0</th>
<th>−0.5</th>
<th>0.0</th>
<th>0.5</th>
<th>1.0</th>
<th>1.5</th>
<th>2.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Exposure Control</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No. of Constraints Violated</td>
<td>6</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>5</td>
<td>9</td>
<td>7</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Average Percentage of Violations</td>
<td>1.6</td>
<td>1.3</td>
<td>2.7</td>
<td>4.2</td>
<td>1.8</td>
<td>1.2</td>
<td>.8</td>
<td>.6</td>
<td>5.6</td>
</tr>
<tr>
<td>Average Weight</td>
<td>34.5</td>
<td>28.7</td>
<td>31.8</td>
<td>22.2</td>
<td>39.8</td>
<td>34.6</td>
<td>55.0</td>
<td>25.3</td>
<td>21.5</td>
</tr>
<tr>
<td>Exposure Control</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No. of Constraints Violated</td>
<td>15</td>
<td>15</td>
<td>17</td>
<td>18</td>
<td>18</td>
<td>14</td>
<td>13</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>Average Percentage of Violations</td>
<td>5.1</td>
<td>5.2</td>
<td>5.0</td>
<td>4.7</td>
<td>4.6</td>
<td>4.5</td>
<td>4.2</td>
<td>3.5</td>
<td>3.3</td>
</tr>
<tr>
<td>Average Weight</td>
<td>45.5</td>
<td>45.5</td>
<td>46.1</td>
<td>46.8</td>
<td>43.8</td>
<td>36.6</td>
<td>31.9</td>
<td>38.9</td>
<td>41.6</td>
</tr>
</tbody>
</table>

### TABLE 3

**Violations of Information Constraints (WDM)**

<table>
<thead>
<tr>
<th>$\theta$</th>
<th>−2.0</th>
<th>−1.5</th>
<th>−1.0</th>
<th>−0.5</th>
<th>0.0</th>
<th>0.5</th>
<th>1.0</th>
<th>1.5</th>
<th>2.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Exposure Control</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No. of Constraints Violated</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Average % of Violations</td>
<td>54.0</td>
<td>33.7</td>
<td>50.0</td>
<td>42.0</td>
<td>54.2</td>
<td>48.5</td>
<td>43.6</td>
<td>8.2</td>
<td>31.0</td>
</tr>
<tr>
<td>Weight</td>
<td>98.0</td>
<td>98.0</td>
<td>98.0</td>
<td>98.0</td>
<td>98.0</td>
<td>98.0</td>
<td>98.0</td>
<td>98.0</td>
<td>98.0</td>
</tr>
<tr>
<td>Exposure Control</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No. of Constraints Violated</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Average % of Violations</td>
<td>47.7</td>
<td>47.4</td>
<td>36.2</td>
<td>27.8</td>
<td>28.6</td>
<td>26.7</td>
<td>28.6</td>
<td>29.2</td>
<td>32.2</td>
</tr>
<tr>
<td>Weight</td>
<td>98.0</td>
<td>98.0</td>
<td>98.0</td>
<td>98.0</td>
<td>98.0</td>
<td>98.0</td>
<td>98.0</td>
<td>98.0</td>
<td>98.0</td>
</tr>
</tbody>
</table>
FIGURE 2. Conditional item-exposure rates at \( \theta = -2.0, -1.5, \ldots, 2.0 \) for the WDM and STA after \( n = 50 \) items without and with exposure control.

We have no explanation for this outlier, which simply may be the result of the local composition of the item pool at this value. Though the goal values or bounds in these constraints were set equal to the actual results for the STA (see earlier), Table 3 shows that these values were often not realized for the WDM.

We also compared the conditional exposure rates for the two methods. In the condition without exposure control, both methods capitalized on the presence of a small set of items in the pool. But in the condition with exposure control, all exposure rates were below target. Also, as shown in Figure 2, the distributions of the conditional exposure rates given the \( \theta \) values were virtually indistinguishable for each method. These results were due to the effectiveness of the new exposure-control method that imposes the target value as a “hard constraint” on the selection of the items (van der Linden & Veldkamp, 2004).

**FinalComments**

A feature of the WDM is the possibility of trade-offs between the test specifications, and hence of violations of constraints that have lower weights. This feature
was confirmed in our empirical study that showed serious violations of the content constraints by the WDM as well as much less informative ability estimates away from the center of the ability distribution.

It is not our claim that such results are always necessary. In fact, the case study by Robin et al. (2004) showed only a small number of constraints violations for the WDM and no difference between test information with the STA at all. The critical difference between the Robin et al. and the present study is that in the former the content specifications, weights, and item pools were the results of a long history of experience with the trade-offs between these quantities in the ETS program. On the other hand, in the present study the weights were used as “content parameters”; their values were set by a test specialist using content considerations only.

In a new testing program, it is certainly possible to use the weights in the WDM as “control parameters” that are to be adjusted until the best trade-off is found. But we expect this process, which has to be executed through a combination of computer simulations and/or actual test administrations, to be long and tedious. Moreover, it has to be repeated every time the test specifications or the item pool changes. The parallel with the control parameters in the Sympon-Hetter method, that have to be adjusted until the best trade-off between the item-exposure rates of all items in the pool is found, is obvious.

The challenge involved in this adjustment process is revealed by a comparison between the results in Tables 2 and 3. Because the item-exposure rates were controlled by hard constraints, the only trade-off possible for the WDM in the condition with exposure control was between the test information and the content constraints. The weights for the test information set by the test specialists were high (98%) relative to those for the majority of the content constraints. As a result, the effect of the introduction of exposure control was not only a larger number of violations of the information constraints (Table 3) but also more frequent violations of the content constraints (Table 2). If the weights had been treated as control parameters, it would have been necessary to adjust them until a combination was found that improved both the results for the information constraints and the majority of the content constraints.

Note

This study received funding from the Law School Admissions Council (LSAC). The opinions and conclusions contained in this article are those of the author and do not necessarily reflect the policy and position of LSAC. The study in this article was conducted while the author was a Fellow of the Center for Advanced Study in the Behavioral Sciences, Stanford, CA. He is indebted to the Spencer Foundation for a grant awarded to the Center to support his fellowship. Assistance by Stephen Luebke and David Kary during the preparations of this study and by Wim M. M. Tielen in the simulation study is gratefully acknowledged.

References


vander Linden, W. J., & Veldkamp, B. P. (2004). Constraining item-exposure rates in comput-
erized adaptive testing with shadow tests. *Journal of Educational and Behavioral Statistics,
Measurement, 36*, 253–266.
Veldkamp, B. P., & van der Linden, W. J. (2002). Multidimensional adaptive testing with
Veldkamp, B. P., & van der Linden, W. J. (submitted). Implementing Sympsoon-Hetter item-
exposure control in adaptive testing with shadow tests.

**Author**

WIM J. VAN DER LINDEN is a Professor of Measurement and Data Analysis, University of
Twente, PO Box 217, 7500 AE Enschede, The Netherlands; w.j.vanderlinden@utwente.nl.
His primary interests include test theory, applied statistics, and research methods.