Journal of Educational Measurement
Fall 1990, Vol. 27, No. 3, pp. 241–253

The Construction of Customized Two-Stage Tests

Jos J. Adema
University of Twente

In this paper mixed integer linear programming models for customizing two-stage tests are given. Model constraints are imposed with respect to test composition, administration time, inter-item dependencies, and other practical considerations. It is not difficult to modify the models to make them useful in constructing multistage tests.

An idea implicit in the practice of customized testing is that each examinee takes a single (conventional) paper-and-pencil test. However, it is well known that adaptive procedures such as two-stage testing are more efficient than conventional tests. It is also known that two-stage tests, consisting of a routing test followed by a second test, can be as easily administered as paper-and-pencil tests (Lord, 1971, 1980; Fischer & Pendl, 1980). Routing tests with known item parameters can be scored immediately, possibly even by the examinee himself or herself. The second test is then selected on the basis of the examinee's score. One may wonder if it would not also be possible to tailor two-stage tests from an item bank to a customer's population of examinees in an efficient way. The purpose of this paper is to provide the test constructor with a computerized method for constructing customized two-stage tests from an item bank calibrated under an item response model. The method allows the test constructor to formulate demands with respect to useful test properties such as test composition. The two-stage test that gives the most information and at the same time fulfills all the specified demands is selected by the proposed method. It is obvious that selecting such a test by hand is practically impossible.

Two-stage tests are most valuable in situations where the group tested has a range of ability too wide to be measured effectively by a peaked conventional test (Lord, 1980, p. 146). Other forms of adaptive testing are often more efficient than two-stage testing. However, as already mentioned, two-stage tests have the advantage that they can be administered by paper and pencil. Wainer and Kiely (1987) gave a number of problems that are associated with most forms of adaptive testing, like context effects, lack of robustness, and item difficulty ordering (the items administered in the beginning are too difficult for the least able students). In general, however, two-stage and multistage tests need not be sensitive to these problems if the problems are dealt with appropriately during the construction of the tests.

The author would like to thank Leendert van Staalduinen, Martijn Berger, Wim van der Linden, Robert Jannarone, two anonymous reviewers, and the editor for their useful comments on a draft of this article.

As Yen (1983) and Theunissen (1985) pointed out, integer linear programming (ILP) models can be used to construct tests (although we use the word *model,* an LP model is in fact an optimization problem). Mixed integer linear programming (MILP) models for the construction of two-stage tests will be given in this paper. MILP models contain continuous and integer decision variables, and their objective functions and constraints are linear in their decision variables. These models take into account practical constraints—that is, demands with respect to the properties of the test. Such practical constraints can control the test composition, the administration time, and the like. The MILP models are based on the maximin model for test construction (van der Linden & Boekkooi-Timminga, 1989), and can be extended to the construction of multistage tests. The MILP approach is used in this paper because the solution procedures for MILP models can easily deal with practical constraints.

Throughout this paper, it is assumed that a bank of items calibrated under an item response model is available. Two popular item response models are the Rasch model (Rasch, 1960) and the logistic three-parameter model (Birnbaum, 1968). For the three-parameter model the probability $P_i(\theta)$ that an examinee with ability level θ answers item i correctly is given by

$$P_i(\theta) = c_i + (1 - c_i)\frac{\exp(a_i(\theta - b_i))}{1 + \exp(a_i(\theta - b_i))},$$

where a_i, b_i, and c_i are the discrimination, difficulty, and guessing parameters of Item i. The information function (see Lord, 1980, chap. 5) of Item i can be written as

$$I_i(\theta) = \frac{a_i^2(1 - c_i)}{(c_i + \exp(a_i(\theta - b_i)))(1 + \exp(-a_i(\theta - b_i)))^2}.$$

Setting $a_i = 1$ and $c_i = 0$ in the formulas above produces the Rasch model. The information function of a test with n items is found by adding the item information functions:

$$I(\theta) = \sum_{i=1}^{n} I_i(\theta).$$

An important feature of the test information function for an unbiased estimator of ability is that it is the reciprocal of the (asymptotic) sampling variance of the estimator.

In the maximin model the test constructor has to provide the relative shape of a target test information function. This is done by giving target values at certain ability points. Furthermore, the constructor must specify the number of items in the test. Thus, specifications might include setting the number of items in the test to 30 and requiring that the proportion of target test information function values at ability levels -1, 0, and 1 be 1:2:1.

The idea is to select the items that maximize the information in the test, subject to the constraint that the test information function still reflects the desired shape. This method circumvents the problem of specifying the exact

height of the target test information function, which can be difficult for the average test constructor because the metric of the information measure may have no clear meaning.

For the maximin model the decision variables x_i are specified such that

$$x_i = \begin{cases} 0 \text{ item } i \text{ not in the test,} \\ 1 \text{ item } i \text{ in the test,} \end{cases} \qquad i = 1, \ldots, I$$

where I is the number of items in the item bank. Let p_h, $h = 1, \ldots, H$ be the relative amount of information required at ability level θ_h as specified by the test constructor and $I_i(\theta_h)$ the amount of information at ability level θ_h for item i. Let $(p_1 y, \ldots, p_H y)$ be a series of lower bounds to the target test information function, where y is an additional decision variable that should be maximized under the constraint that the number of items in the test is equal to n as specified by the test constructor. The maximin model can then be formulated as follows:

$$\text{Max. } y \tag{1}$$

subject to

$$\sum_{i=1}^{I} I_i(\theta_h) x_i - p_h y \geq 0, \qquad h = 1, 2, \ldots, H, \tag{2}$$

$$\sum_{i=1}^{I} x_i = n, \tag{3}$$

$$x_i \in \{0, 1\}, \qquad i = 1, 2, \ldots, I, \tag{4}$$

$$y \geq 0. \tag{5}$$

Constraints (2) assure that the $p_h y$ are indeed a series of lower bounds to the target test information function at the specified ability levels. The coefficients p_h in constraints (2) force the target information function to take on the desired shape. Constraints (2) form the min-part of the model. The max-part of the model is found in the objective function (1). The variable y is maximized so that the lower bounds $p_h y$ are as high as possible. Constraint (3) implies that the number of items in the test is equal to n. A justification and an extensive explanation of the maximin model are found in van der Linden and Boekkooi-Timminga (1989).

In the operations research literature (e.g., Wagner, 1975; Hartley, 1985), the maximin model is known as a MILP model, because it contains continuous (y) as well as integer variables (x_i, $i = 1, \ldots, I$) and because the constraints (2) through (5) as well as the objective function (1) are linear in their variables. These models can be solved by branch-and-bound methods, which maximize y and simultaneously compute the corresponding optimal x_i values (Land & Doig, 1960). The branch-and-bound method is implemented in standard computer codes that are amply available in textbooks (e.g., Land & Powell, 1973; Kuester and Mize, 1973; Syslo, Deo, & Kowalik, 1983); software libraries such as NAG (Numerical Algorithms Group Limited); and software packages such as

IBM MPSX/370 V2 (1988), Lando (Anthonisse, 1979), or Lindo (Schrage, 1987).

In the next section I will formulate several practical constraints and then present models for constructing two-stage tests. I will also show how these models can be extended to solve the problem of constructing multistage tests, and give examples to illustrate the models.

Practical Constraints

The MILP models for the construction of two-stage tests may include a number of practical constraints as formulated by van der Linden and Boekkooi-Timminga (1989). In most item banks we can distinguish subsets of items; for instance, items may be grouped into subsets on the basis of their content or format (e.g., multiple choice, completion).

Formally, three different kinds of subsets can be distinguished:

$E_j (j = 1, 2, \ldots, J_E)$: Exactly n_{E_j} items should be selected from the subsets E_j;
$F_j (j = 1, 2, \ldots, J_F)$: At most n_{F_j} items may be selected from the subsets F_j;
$G_j (j = 1, 2, \ldots, J_G)$: At least n_{G_j} items should be selected from the subsets G_j.

For example, suppose that an item bank for French vocabulary is partitioned with respect to its content (noun, verb, or adjective) and its format (multiple choice or matching). A test constructor may then have the following demands with respect to the composition of the test:

(1) The number of verb items in the test is equal to 6:

$$\sum_{i \in E_1} x_i = 6,$$

where E_1 is the subset of verb items.

(2) The number of noun items in the test is not greater than 5:

$$\sum_{i \in F_1} x_i \leq 5,$$

where F_1 is the subset of noun items.

(3) The number of adjective items in the test is greater than 5:

$$\sum_{i \in G_1} x_i \geq 6,$$

where G_1 is the subset of adjective items.

(4) The number of multiple-choice items in the test is not greater than 12:

$$\sum_{i \in F_2} x_i \leq 12,$$

where F_2 is the subset of multiple-choice items.

(5) The number of matching items in the test is greater than 6:

$$\sum_{i \in G_2} x_i \geq 7,$$

where G_2 is the subset of matching items.

In general, constraints related to the subsets E_j ($j = 1, \ldots, J_E$); F_j ($j = 1, \ldots, J_F$); and G_j ($j = 1, \ldots, J_G$) can be formulated as follows:

$$\sum_{i \in E_j} x_i = n_{E_j}, \qquad j = 1, 2, \ldots, J_E, \tag{6}$$

$$\sum_{i \in F_j} x_i \leq n_{F_j}, \qquad j = 1, 2, \ldots, J_F, \tag{7}$$

$$\sum_{i \in G_j} x_i \geq n_{G_j}, \qquad j = 1, 2, \ldots, J_G. \tag{8}$$

If we also want to restrict the administration time of the test, we can do this by including the following constraint:

$$\sum_{i=1}^{I} t_i x_i \leq T, \tag{9}$$

where t_i is an estimate of the time an examinee from the population needs for answering item i, and T is the upper bound on the administration time for the test.

Another possible kind of constraint reflects possible dependencies among items. The item bank may contain subsets of items V_j ($j = 1, 2, \ldots, J_V$) from which it is not allowed to select more than one item, because every item in such a subset contains information about the answers to the other items in that subset. This demand can be formulated as the following linear constraint:

$$\sum_{i \in V_j} x_i \leq 1, \qquad j = 1, 2, \ldots, J_V. \tag{10}$$

On the other hand, it may also be desirable to select either all or none of the items from a subset W_j ($j = 1, 2, \ldots, J_W$):

$$\sum_{i \in W_j} x_i = n_{W_j} x_{i_j}, \qquad j = 1, 2, \ldots, J_W, \tag{11}$$

where n_{W_j} is the number of items in W_j and x_{i_j} is the decision variable for an arbitrary item in W_j. For instance, let the first four items in the item bank form subset W_1 ($n_{W_1} = 4$) such that all four items should either all be included in the test or all be excluded. The decision variables corresponding to the items are x_1 through x_4. The constraint related to subset W_1 can be formulated as follows:

$$x_1 + x_2 + x_3 + x_4 = 4x_1,$$

where x_1 is arbitrarily chosen. Imposing this constraint makes feasible only solutions with $x_1 = x_2 = x_3 = x_4 = 0$ or $x_1 = x_2 = x_3 = x_4 = 1$.

Two-Stage Test Construction

If a two-stage test is administered as a paper-and-pencil test, the constructor must make a decision about the ability levels at which the second tests should aim. Theunissen (1986) showed how these ability levels can be computed if the tests are constructed sequentially.

If the test is administered by computer, it is possible to adapt the second test to

the ability of the individual examinee, because the ability of the examinees can be estimated directly from the score on the routing test. In this discussion it is assumed that a paper-and-pencil procedure is followed, but obviously, the MILP models are also applicable if the test is administered by computer.

A test constructor can impose constraints on item selection either at the subtest level (routing or second test) or at the test level. When specifying the practical constraints separately for the routing and second tests is desirable, the test constructor imposes the constraints at the subtest level. When constraints on the entire two-stage test are desired and separate specifications for the routing and second tests are unnecessary, the constructor imposes the constraints at the test level. Each option will be considered in the following sections. In each section the presentation of the models is followed by an explanation.

Constraints at the Subtest Level

In this case, separate constraints for the routing test and the second tests are imposed. A general MILP model that selects items for the routing test r can be formulated as follows:

$$\text{Max. } y_r \tag{12}$$

subject to

$$\sum_{i=1}^{I} I_i(\theta_h) x_{ir} - y_r \geq 0, \qquad h = 1, \ldots, H, \tag{13}$$

$$\sum_{i=1}^{I} x_{ir} = n_r, \tag{14}$$

$$\sum_{i \in E_{rj}} x_{ir} = n_{E_{rj}}, \qquad j = 1, \ldots, J_{E_r}, \tag{15}$$

$$\sum_{i \in F_{rj}} x_{ir} \leq n_{F_{rj}}, \qquad j = 1, \ldots, J_{F_r}, \tag{16}$$

$$\sum_{i \in G_{rj}} x_{ir} \geq n_{G_{rj}}, \qquad j = 1, \ldots, J_{G_r}, \tag{17}$$

$$\sum_{i=1}^{I} t_{ir} x_{ir} \leq T_r, \tag{18}$$

$$\sum_{i \in V_{rj}} x_{ir} \leq 1, \qquad j = 1, \ldots, J_{V_r}, \tag{19}$$

$$\sum_{i \in W_{rj}} x_{ir} = n_{W_{rj}} x_{i_j r}, \qquad j = 1, \ldots, J_{W_r}, \tag{20}$$

$$x_{ir} \in \{0, 1\}, \qquad i = 1, \ldots, I, \tag{21}$$

$$y_r \geq 0, \tag{22}$$

where $x_{ir} = 1$ if item i is selected for the routing test, and 0 otherwise. The practical constraints in this MILP model are the constraints (15) through (20). Constraints (15) through (17) govern the composition of the routing test. The administration time is restricted by (18), whereas (19) and (20) constrain

dependencies between items. The number of items in the routing test is equal to n_r. Because an adaptive test is supposed to measure ability accurately over the entire range, it is assumed that the test constructor wants the same amount of information for each ability level, which implies that the values of p_h are set to one (see the constraints in Inequality 2). The advantage of this approach is that the test constructor does not have to specify the relative amount of information that is required for different θ values.

If a routing test is constructed by solving (12) through (22), the test constructor can continue the procedure by specifying the ability levels at which the second tests should aim. The second tests are constructed to give maximal information at the specified ability levels. The following is a zero-one linear programming (ZOLP) model—that is, a linear programming model with all variables of the 0–1 type—for constructing a second test s at a specified ability level θ^*:

$$\text{Max.} \sum_{i=1}^{I} I_i(\theta^*)\, x_{is} \tag{23}$$

subject to

$$\sum_{i \in U} x_{is} = 0, \tag{24}$$

(14)–(17) with subindex r replaced by s

$$\sum_{i=1}^{I} t_{is}(\theta^*)\, x_{is} \leq T_s, \tag{25}$$

(19)–(21) with subindex r replaced by s

where U is the set of items selected for the routing test and $x_{is} = 1$ if item i is selected for the second test, and 0 otherwise. Constraint (24) implies that all decision variables related to items in U are equal to zero. This ZOLP model should be solved for all specified values of θ^*. In constraint (25) it is assumed that the time needed for answering an item depends on ability level θ^*. In practice, this implies that a large number of response times must be estimated. If these estimates are unavailble, it is impossible to include constraint (25) in the model. It should be noted that, say, G_{r1} and G_{s1} can be two totally different sets of items. For instance, suppose an item bank contains 200 items, of which the first 100 items are multiple choice and the other 100 items are the matching type. In this case a test constructor might want 20 items in the routing test, with at least half the items multiple choice. Then G_{r1} is the set of multiple-choice items and F_{r1} the set of matching items. The second tests should also consist of 20 items, but now at least half the items should be the matching type. In that case G_{s1} is the set of matching items and F_{s1} the set of multiple-choice items.

Although the model for the second tests is not as hard to solve as the model for the routing test, it is still not a simple task for every possible set of practical constraints. Therefore, advanced algorithms would also be needed for solving the ZOLP models for the second tests. Adema (1988) gave a heuristic based on the branch-and-bound method that can be used for solving the present test construc-

tion problems. This heuristic gives acceptable results, as the following examples show.

Constraints at the Test Level

In models for constructing two-stage tests with demands at the test level, the constraint on the administration time is omitted. An advantage of specifying the demands at the test level is that the MILP model for the routing test is less restricted, which implies that the routing test will give more information at the chosen ability levels. The MILP model for the construction of the routing test r can be formulated as follows:

$$\text{Max. } y_r \tag{26}$$

subject to

(13)–(14),

$$\sum_{i \in E_j} x_{ir} \leq n_{E_j}, \qquad j = 1, \ldots, J_E, \tag{27}$$

$$\sum_{i \in F_j} x_{ir} \leq n_{F_j}, \qquad j = 1, \ldots, J_F, \tag{28}$$

$$\sum_{i \in V_j} x_{ir} \leq 1, \qquad j = 1, \ldots, J_V, \tag{29}$$

$$\sum_{i \in W_j} x_{ir} = n_{W_j} x_{j,r}, \qquad j = 1, \ldots, J_W, \tag{30}$$

along with (21) and (22).

In this model it is supposed that the number of items in the routing and the second tests are specified separately. Restrictions on the subsets G_j ($j = 1, \ldots, J_G$), from which at least n_{G_j} items should be selected, are not needed, because a two-stage test with 0 items from G_j in the routing test and at least n_{G_j} items in the second test is feasible. Thus zero or more items from G_j would be selected for the routing test, which would result in a redundant constraint. The way the constraints in (30) are formulated precludes some of the items of subset W_j ($j = 1, \ldots, J_W$) from being in the routing test while others are in the second tests, because the constraints in (30) imply that all or none of the items in the subsets W_j are selected for the routing test.

The model for the construction of a second test at ability level θ^* is:

$$\text{Max.} \sum_{i=1}^{I} I_i(\theta^*)\, x_{is} \tag{31}$$

subject to

$$\sum_{i=1}^{I} x_{is} = n_s, \tag{32}$$

$$\sum_{i \in U} x_{is} = 0, \tag{33}$$

$$\sum_{i \in E_j} x_{is} = n_{E_j} - \sum_{i \in E_j} x_{ir}, \qquad j = 1, \ldots, J_E, \tag{34}$$

$$\sum_{i \in F_j} x_{is} \leq n_{F_j} - \sum_{i \in F_j} x_{ir}, \qquad j = 1, \ldots, J_F, \tag{35}$$

$$\sum_{i \in G_j} x_{is} \geq n_{G_j} - \sum_{i \in G_j} x_{ir}, \qquad j = 1, \ldots, J_G, \tag{36}$$

$$\sum_{i \in V_j} x_{is} \leq 1 - \sum_{i \in V_j} x_{ir}, \qquad j = 1, \ldots, J_V, \tag{37}$$

$$\sum_{i \in W_j} x_{is} = n_{W_j} x_{i_j s}, \qquad j = 1, \ldots, J_W, \tag{38}$$

$$x_{is} \in \{0, 1\}, \qquad i = 1, \ldots, I, \tag{39}$$

where U is the set of items selected for the routing test. In this model x_{ir} is no longer a variable, because it is fixed at the value 0 or 1 after the model for the routing test has been solved. According to constraints (34) the number of items from F_j, $j = 1, \ldots, J_F$, in the second tests should not be greater than the maximum number of items from F_j in the whole two-stage test, as specified by the test constructor, minus the number of items of F_j selected for the routing test. The constraints (35) through (38) can be interpreted in the same way. Objective function (31) and constraints (32), (33), (38), and (39) can also be found in the ZOLP model for the construction of second tests with demands at subtest level. For the sake of clarity these constraints were given again.

Extension of the Model With Constraints at the Test Level

It is possible that no feasible solution for the LP model of the second test exists if the constraints are imposed at the test level. It is, for instance, possible that at least 7 addition items and exactly 4 subtraction items must be selected for the second test, whereas the number of required items in the second test is 10. This problem can be approached as follows: Let the integer variables z_{E_j} $(j = 1, \ldots, J_E)$ and z_{Gj} $(j = 1, \ldots, J_G)$ denote the minimum number of items from sets E_j $(j = 1, \ldots, J_E)$ and G_j $(j = 1, \ldots, J_G)$ to be selected for the second test. Several criteria such as content and item format can be found for partitioning an item bank in subsets of items. Let M denote the number of criteria. The coefficients δ_{mE_j} and δ_{mG_j} are now defined as follows:

$$\delta_{mE_j} = \begin{cases} 0 \text{ if the items are not in } E_j \text{ because of criterion } m \\ 1 \text{ if the items are in } E_j \text{ because of criterion } m \end{cases}$$

$$\delta_{mG_j} = \begin{cases} 0 \text{ if the items are not in } G_j \text{ because of criterion } m \\ 1 \text{ if the items are in } G_j \text{ because of criterion } m. \end{cases}$$

These variables and coefficients are included in the constraints in (27) of the model for the routing test as well as in two new kinds of constraints for this model:

$$\sum_{i \in E_j} x_{ir} + z_{E_j} = n_{E_j}, \qquad j = 1, \ldots, J_E, \tag{40}$$

$$\sum_{i \in G_j} x_{ir} + z_{G_j} \geq n_{G_j}, \qquad j = 1, \ldots, J_G, \tag{41}$$

$$\sum_{j=1}^{J_E} \delta_{mE_j} z_{E_j} + \sum_{j=1}^{J_G} \delta_{mG_j} z_{G_j} \leq n_s, \qquad m = 1, \ldots, M. \tag{42}$$

For each criterion, constraints (40) through (42) prevent the minimum number of items that should be selected for the second test from exceeding the specified value n_s.

Multistage Testing

Multistage testing differs from two-stage testing in that more than one subtest is administered after the routing test. The choice of each subtest depends on the scores on the preceding test. The ZOLP models for the construction of second-stage tests with constraints at the subtest level can be used, with one modification for the construction of multistage tests. In this case, the set U in the constraint in (24) must be redefined as the set of items selected for the preceding subtests. The models with constraints specified at the test level can also be modified in a straightforward way, so that they are useful for the construction of multistage test procedures as well.

Examples

In this section two examples will be given. In the first example a two-stage test with constraints specified at subtest level will be constructed sequentially. In the second example the constraints will be specified at the test level. A simulated item bank for French vocabulary with 300 items that fit the 3-parameter model ($a_i \sim U(0.5, 1.5)$; $b_i \sim U(-3, 3)$; $c_i = 0.2$) will be used for both examples. The item bank is partitioned with respect to its content into noun (Items 1–100), verb (Items 101–200), and adjective parts (Items 201–300). The first 50 items of those subsets are of the multiple-choice type and the other items are of the matching type.

The linear programming models with 0–1 variables that are used in the examples can be solved by a branch-and-bound method (Land & Doig, 1960). The models were solved on a DEC-2060 computer with a modified version of the program Lando (Centre for Mathematics and Computer Science). The modifications in the branch-and-bound part of the algorithm have been described by Adema (1988). The CPU times in the examples do not include the time needed for reading the input file, initializing, and writing to the output file. The CPU times are shown to give an impression of the practicability of the approaches.

Example 1

Suppose a test constructor has the following demands with respect to the routing test: (a) The number of items in the routing test equals 20; (b) the ability levels at which the target information function is specified for the routing test are $\theta_1 = -2$, $\theta_2 = 0$, and $\theta_3 = 2$; (c) the examinee should answer fewer than eight noun items, exactly six verb items, and more than seven adjective items; and (d)

the number of multiple-choice items the examinee should answer is smaller than 13, whereas the number of matching items is greater than 6.

The test constructor wants three second tests, which should be peaked at ability levels -2, 0, and 2. The demands with respect to the composition of these tests are the same as for the routing test.

A routing test is constructed using the MILP model (12) through (22). The test information is 2.752 at θ_1, 2.757 at θ_2, and 2.736 at θ_3. Next, the second tests are constructed. The information values of the second tests are 4.712 for the second test at $\theta^* = -2$; 5.450 for the second test at $\theta^* = 0$; and 5.294 for the second test at $\theta^* = 2$. The total CPU time needed for constructing the routing and second tests was 11.2 seconds.

Example 2

In this example we consider the case of test construction with demands specified at test level.

The demands of the test constructor are as follows: (a) The ability levels at which the target information function is specified for the routing test are $\theta_1 = -2$, $\theta_2 = 0$, and $\theta_3 = 2$; (b) the second tests are peaked at the ability levels -2, 0, and 2; (c) the number of items in the routing test and second tests equals 20; (d) in all, the examinee should answer fewer than 15 noun items, exactly 12 verb items, and more than 15 adjective items; and (e) the number of multiple-choice items the examinee should answer is smaller than 25, whereas the number of matching items is greater than 13.

A MILP model for constructing a routing test that fulfills the demands is formulated. The routing test constructed with this model comprises 6 noun items, 6 verb items, 8 adjective items, 6 multiple-choice items, and 14 matching items. The test information value is 2.750 at θ_1, 2.754 at θ_2, and 2.733 at θ_3.

Given the composition of the routing test, the restrictions on the composition of the second tests are (a) the second tests should contain fewer than 9 noun items, exactly 6 verb items, and more than 7 adjective items; and (b) the second tests should contain fewer than 19 multiple-choice items. A restriction on the number of matching items is not needed, because this number should be 0 or more.

The information values of the second tests are 4.747 for the second test at $\theta^* = -2$; 5.450 for the second test at $\theta^* = 0$; and 5.168 for the second test at $\theta^* = 2$. The total CPU time for constructing the routing and second tests was 8.274 seconds.

Discussion

In this paper I proposed mixed integer linear programming models for constructing two-stage testing procedures. The model with constraints at the subtest level is easy to apply, because the routing and the second test can be constructed separately. When the test constructor specifies the constraints at the test level, some problems arise because he or she must account for the construction of the second-stage test in the model for the routing test, and vice versa. These problems can be solved by introducing the constraints in (40) through (42).

Compared to manual two-stage test construction, the use of mixed integer linear programming has two main advantages. Firstly, manual item selection, such as by spreading out item cards on the floor (Bejar, 1985), is time consuming, especially when several kinds of criteria should be considered. As the examples show, test construction by MILP models does not have this drawback. Secondly, item selection by MILP models guarantees the test constructor good psychometric test properties by constructing a two-stage test that is almost optimal with respect to the maximin criterion.

There are other forms of adaptive testing that may be more efficient than two-stage testing. The construction of such adaptive tests using mathematical programming models is not always practical and can be difficult, if not impossible. An example of an impractical case is tailored testing. In tailored testing, one item is selected at a time, and the selection criterion involves simply selecting the item that gives the most information at the current ability estimate of the examinee, subject to the practical constraints. After the selection of an item, the practical constraints must be adjusted, just like we adjusted constraints (31) through (39) to the items selected for the routing test. Thus, a complex method like mathematical programming is not needed for tailored test item selection.

References

Adema, J. J. (1988). *A note on solving large-scale zero-one programming problems* (Research Report No. 88-4). Enschede, The Netherlands: University of Twente, Department of Education.

Anthonisse, J. M. (1979). *Lando* [Computer program]. Amsterdam, The Netherlands: Centre for Mathematics and Computer Science.

Bejar, I. I. (1985). Speculation on the future of test design. In S. E. Embretson (Ed.), *Test design: Developments in psychology and psychometrics*. New York: Academic Press.

Birnbaum, A. (1968). Some latent trait models. In F. M. Lord & M. R. Novick, *Statistical theories of mental test scores*. Reading, MA: Addison-Wesley.

Fischer, G. H., & Pendl, P. (1980). Individualized testing on the basis of the dichotomous Rasch model. In L. J. T. van der Kamp, W. F. Langerak, & D. N. M. de Gruijter (Eds.), *Psychometrics for educational debates*. New York: John Wiley & Sons.

Hartley, R. (1985). *Linear and nonlinear programming*. Chichester: Ellis Horwood.

IBM Mathematical Programming System Extended/370 Version 2 (1988). *Program reference manual* [Computer program manual]. Rome, Italy: Laboratorio di Sciluppo Software, IBM Italia. (Form No. SH 19-6553-0).

Kuester, J. L., & Mize, J. H. (1973). *Optimization techniques with Fortran*. New York: McGraw-Hill.

Land, A. H., & Doig, A. G. (1960). An automated method of solving discrete programming problems. *Econometrica, 28*, 497–520.

Land, A. H., & Powell, S. (1973). *Fortran codes for mathematical programming: Linear, quadratic and discrete*. London: John Wiley & Sons.

Lindo Systems Inc. (1989). *Lindo* [Computer program]. San Francisco: Scientific Press.

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

Lord, F. M. (1980). *Applications of item response theory to practical testing problems*. Hillsdale, NJ: Erlbaum.

Rasch, G. (1960). *Probabilistic models for some intelligence and attainment tests.* Copenhagen: Nielsen and Lydicke.

Schrage, L. (1987). *User's manual for Lindo.* Redwood City: The Scientific Press.

Syslo, M. M., Deo, N., & Kowalik, J. S. (1983). *Discrete optimization algorithms: With Pascal programs.* Englewood Cliffs, NJ: Prentice-Hall.

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

Theunissen, T. J. J. M. (1986). Some applications of optimization algorithms in test design and adaptive testing. *Applied Psychological Measurement, 10,* 381–389.

van der Linden, W. J., & Boekkooi-Timminga, E. (1989). A maximin model for test design with practical constraints. *Psychometrika, 53,* 237–247.

Wagner, H. M. (1975). *Principles of operations research.* London: Prentice-Hall.

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

Yen, W. M. (1983). Use of the three-parameter model in the development of a standardized achievement test. In R. K. Hambleton (Ed.), *Applications of item response theory.* Vancouver: Educational Research Institute of British Columbia.

Author

JOS J. ADEMA is Research Associate, University of Twente, Department of Education, Division of Educational Measurement and Data Analysis, P.O. Box 217, 7500 AE Enschede, The Netherlands. *Degrees:* BS, MS, University of Twente. *Specialization:* operations research.