A Cluster-Based Method for Test Construction

Ellen Boekkooi-Timminga
University of Twente

Several methods for optimal test construction from item banks have recently been proposed using information functions. The main problem with these methods is the large amount of time required to identify an optimal test. In this paper, a new method is presented for the Rasch model that considers groups of interchangeable items, instead of individual items. The process of item clustering is described, the cluster-based test construction model is outlined, and the computational procedure and results are given. Results indicate that this method produces accurate results in small amounts of time. *Index terms: information functions, item banking, item response theory, linear programming, test construction.*

In 1968, Birnbaum suggested a procedure based on item response theory for test construction using a target information function. This procedure, which assumes the availability of a calibrated, unidimensional pool of items, was also mentioned by Lord (1977, 1980). Automated methods for item selection based on the use of a target information function have been proposed (Adema, 1990; Boekkooi-Timminga, 1987, 1989, 1990; Theunissen, 1985, 1986; van der Linden & Boekkooi-Timminga, 1989). These methods approach test construction from a mathematical programming perspective, generally using 0-1 linear programming methods. The primary problem with these methods is the large amount of computer time needed to select the best test items, but this is a problem inherent to 0-1 programming problems (e.g., Lenstra & Rinnooy Kan, 1979).

Fast test construction methods are important, especially when tests are constructed interactively—a preferable feature of test construction systems. Here, a new test construction method based on integer programming is proposed for the Rasch model (e.g., Lord, 1980; Rasch, 1960) that selects tests in small amounts of computer time. It was designed to be implemented on microcomputers, such as for classroom use. This method is attractive for test construction problems that consider a limited number of item characteristics, and the larger the item bank, the more computing time will be saved.

This new method, termed here the cluster-based method, assumes that the items in the bank have been grouped according to their item information functions such that items within a group (cluster) can be considered equivalent. Introducing this assumption may reduce the number of decision variables in the model drastically. However, the approximate nature of the equivalence assumption reduces the accuracy of this method; yet this reduction is small.

Because of the above modification of the test construction procedure, some of the usual constraints on item selection cannot be met. For instance, inter-item dependencies (Theunissen, 1986) are difficult to handle. Requirements such as this can be met, however, in an interactive test construction system by adapting constraints and fixing decision variables by the test constructor during the test construction process. For example, if a constructed test includes two items that are not allowed to enter the same test, the test construction process can be restarted, excluding one item from being selected. A description of an interactive test construction system is given in Boekkooi-Timminga (1989).

Item Clustering

Items fitting the Rasch model can be clustered using a simple procedure. The ability (θ) scale

APPLIED PSYCHOLOGICAL MEASUREMENT
Vol. 14, No. 4, December 1990, pp. 341-354
© Copyright 1990 Applied Psychological Measurement Inc.
0146-6216/90/040341-14$1.95

is partitioned into C intervals ($c = 1, ..., C$). All items with difficulties in the same interval are considered to belong to the same cluster. It is assumed that the θ scale is partitioned within a certain range, and items not falling within this range are included in the outermost clusters.

The information function representing all items in cluster c is computed using the mean item difficulty b_c of the items in the cluster. Computational results showed that this information function differed very little from the mean item information function of the items in a cluster for cluster widths of .4 logits or less. For banks of 1,000 items with difficulty parameters drawn from the distributions $b \sim$ N(0,1), $b \sim$ N (0,2), and $b \sim$ U(−3,3), this deviation was less than 1% at all θ levels. The advantage of using the mean item difficulty b_c is that less computational effort is required.

Width of Intervals

The item bank should consist of as few clusters as possible, containing as many items as possible, in order to profit the most from the cluster-based test construction method; however, all items in a cluster should remain interchangeable. To determine the appropriate width of the clusters, the maximum differences between item information values of items located within the same cluster were computed. For interval widths of .5, .4, .3, .25, .2, .1, and .05 logits, the maximum differences were 6%, 3.88%, 2.2%, 1.56%, 1%, .24%, and .08%, respectively. For example, accepting a maximum difference of 4%, interval widths should not exceed .4 logits.

The appropriate widths also depend on the number of items in the bank, and the dispersion of the difficulty parameters. If it is desired, for example, to have at least 20 items in each cluster, a width of .4 is needed for an item bank of 300 items with $b \sim$ U(−3,3). The clusters used for this study were of the same width, but it was also possible to partition the θ scale into clusters of different widths. Wider clusters may be preferred when small numbers of items are located in certain parts of the θ scale, and more narrow clusters may be preferred when there are many items in ranges of the θ scale. How the clusters are actually selected is arbitrary and at the discretion of the user, who should decide on the inaccuracies that are acceptable.

A Cluster-Based Test Construction Method

The cluster-based test construction method consists of three stages. In the first stage, the numbers of items to be selected from the clusters are determined on the basis of the target test information function. Then additional practical constraints (e.g., a maximum test administration time) are considered. Finally, the test items are selected from the clusters. Figure 1 provides a flow chart of the method.

Stage 1: The Basic Model

The model is a generalized version of the maximin model proposed by van der Linden and Boekkooi-Timminga (1989). Using this objective function, only the relative heights of the target function at some freely selected θ levels need to be specified. Test information is maximized under the condition that this relative distribution is fulfilled (i.e., z is maximized, subject to Equation 1). Formally, the target is characterized by a series of lower bounds ($r_1 z, ..., r_K z$), in which z is an additional decision variable to be maximized, and r_k is the relative information value desired at θ level k. In the Basic Model, z is maximized, subject to the following conditions:

$$\sum_{c=1}^{C} x_c I_c(\theta_k) - r_k z \geq 0 \quad , \qquad k = 1, \ldots, K \quad , \tag{1}$$

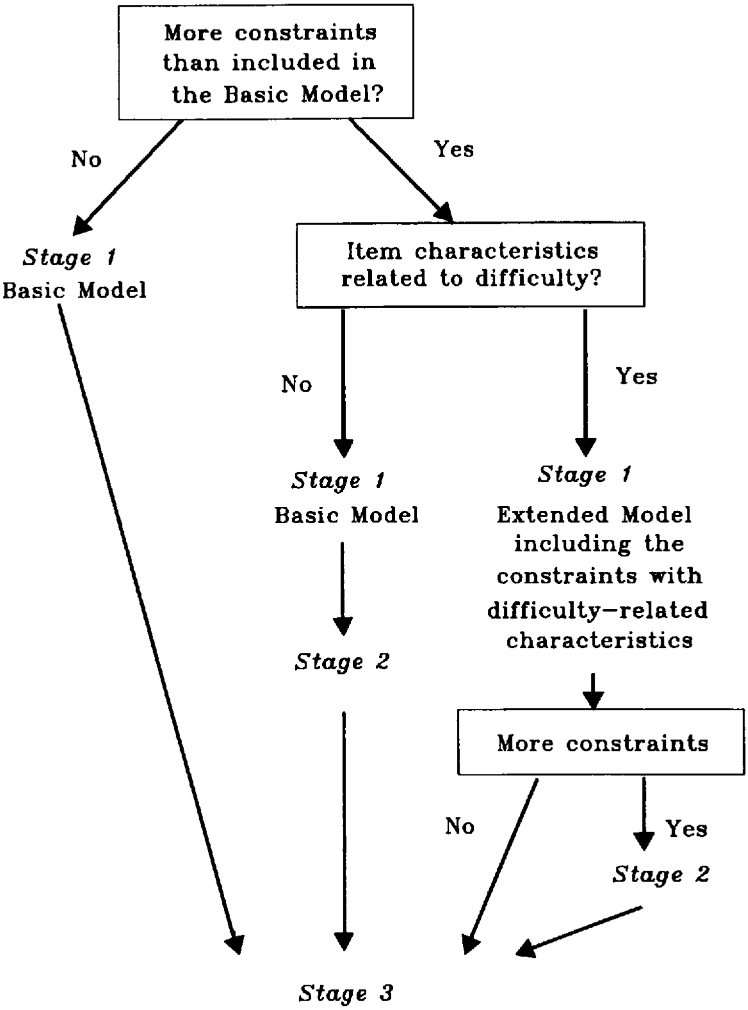
$$\sum_{c=1}^{C} x_c = N \quad , \tag{2}$$

$$l_c \leq x_c \leq u_c, \text{ and integer,} \qquad c = 1, \ldots, C \quad , \tag{3}$$

$$z \geq 0 \quad . \tag{4}$$

The decision variable x_c gives the number of items to be included in the test from cluster c. $I_c(\theta_k)$ is the value of the information function

Figure 1
Flow Chart of the Cluster-Based Test Construction Method


More constraints than included in the Basic Model?
No
Yes
Stage 1
Basic Model
Item characteristics related to difficulty?
No
Yes
Stage 1
Basic Model
Stage 2
Stage 1
Extended Model including the constraints with difficulty–related characteristics
More constraints
No
Yes
Stage 2
Stage 3

for cluster c at θ level k. Equation 2 sets the number of items to be selected at N. The upper and lower bounds for x_c, u_c ($\leq \text{max}_c$), and l_c (≥ 0), respectively, are specified in Equation 3; max_c gives the number of items in cluster c. The constraint in Equation 4 defines the lower bound for decision variable z.

Subject Matter Aspects: The Extended Model

A unidimensional item bank may consist of items on several subject matter aspects (e.g., learning objectives). In most practical test construction situations, test specifications regarding these aspects are required. Because the item difficulty parameters of items reflecting a certain aspect may not be equally spread over the θ continuum, it is necessary to include constraints on these aspects in the Basic Model.

Assume that within each cluster J, nonoverlapping subject matter aspects can be distinguished. Transforming the variables x_c, u_c, and l_c in Equations 1 through 4 into x_{cj}, u_{cj}, and l_{cj}, where x_{cj} defines the number of items on subject matter aspect j ($j = 1, ..., J$) to be selected from cluster c, an extended model can be formulated. In the Extended Model, z is maximized, subject to

$$\sum_{c=1}^{C} \sum_{j=1}^{J} x_{cj} I_c(\theta_k) - r_k z \geq 0 \quad , \qquad k = 1, \ldots, K \quad , \tag{5}$$

$$\sum_{j=1}^{J} \sum_{c=1}^{C} x_{cj} = N \quad , \tag{6}$$

$$\sum_{c=1}^{C} x_{cj} \leq n_j \quad , \qquad j = 1, \ldots, J \quad , \tag{7}$$

$$l_{cj} \leq x_{cj} \leq u_{cj}, \text{ and integer,} \qquad c = 1, \ldots, C \quad , \qquad j = 1, \ldots, J \quad , \tag{8}$$

$$z \geq 0 \quad . \tag{9}$$

The exact, minimum, or maximum numbers of items to be selected from each subject matter aspect can be constrained in this model. For example, Equation 7 defines the maximum numbers of items n_j to be selected from subject matter aspects j, where n_j is not allowed to be greater than the total number of items covering subject matter aspect j ($\max_{cj}$). In this case, Equation 6 can become redundant.

Stage 2: Additional Test Specifications

In Stage 2, additional test requirements can be taken into account (e.g., requirements addressing test administration time, frequency of previous usage of the items, and subject matter if not related to item difficulty). The numbers of items to be selected for the test from each cluster are known from Stage 1. Those clusters from which items need to be selected are further partitioned on the basis of the aspects to be considered in Stage 2. If more than one aspect needs to be considered, the newly formed partitions are partitioned again.

Let each cluster c be partitioned into Q ($q = 1, \ldots, Q$) final partitions called item-aspect groups. Adding many different item aspects results in a large number of decision variables, which slows down the cluster-based method. The number of items to be included in the test from each item-aspect group is determined using integer programming, such that the additional test specifications are fulfilled best.

The following constraints are always required in the Stage 2 model. They guarantee that the numbers of items to be taken from each cluster (x_c), as determined in Stage 1, are actually selected. This is formulated as

$$\sum_{q=1}^{Q} y_{cq} = x_c \quad , \qquad c \in V_c \quad , \tag{10}$$

where V_c is the collection of clusters c and items have to be selected from ($x_c > 0$). The decision variables y_{cq} denote the number of items to be selected from item-aspect group q in cluster c.

A solution in Stage 2 exactly fitting all constraints does not necessarily exist when constraints are added to Equation 10. In order to anticipate this problem, a special type of objective function to be minimized is used, one that considers a weighted sum of variables d_l ($d_l \geq 0$). These variables indicate the degree to which the test specifications are satisfied. As a result, the solution most fitting the requirements will be obtained. The general expression for the objective function is

$$\sum_{l=1}^{L} w_l (b_s^{-1}) d_l \quad , \tag{11}$$

where L is the total number of unknown decision variables d_l ($d_l \geq 0$) included in the model. Weights w_l can be used to indicate the seriousness of not satisfying the corresponding constraint: they are given by the test constructor. In order to compensate for differences in right-hand-side coefficient values b_s (e.g., N in Equations 2 and 6, and n_j in Equation 7), these constants are introduced in Equation 11.

For each constraint, q, two variables can be included, d_{q1} and d_{q2} (d_{q1}, $d_{q2} \geq 0$), whose values are minimized through the objective function. If such variables are required, they can also be included in Equation 10. The next set of possible constraints explicitly deals with the newly introduced item aspects. These constraints state that the number of items to be selected from item aspect q should be as close to m_q (the right-hand-side coefficient value) as possible:

$$\sum_{c \in V_c} y_{cq} = m_q + d_{q1} - d_{q2} \quad , \qquad (12)$$
$$q = 1, \ldots, Q \quad .$$

Several other kinds of constraints can be also included in the model; examples of these can be found in van der Linden and Boekkooi-Timminga (1989). Finally, constraints on the upper and lower bounds of the decision variables y_{cq} are required to take into account the numbers of items available in each item-aspect group.

Stage 3: Individual Item Selection

After the numbers of items to be selected from the clusters have been determined, the individual items need to be selected, which can be done by random or optimal item selection. Random item selection is preferred because it uses less computer time; however, optimal item selection is usually more accurate.

With optimal item selection, a 0-1 programming model is formulated for the problem. For example, minimizing the maximum distance y between the actual test information values and the objective function value z^* obtained from the basic model is formulated as minimizing y, subject to

$$\sum_{i=1}^{I} I_i(\theta_k)x_i - y \leq r_k z^* \quad , \qquad (13)$$
$$k = 1, \ldots, K \quad ,$$

$$\sum_{i \in V_{ic}} x_i = x_c \quad , \qquad (14)$$
$$c = 1, \ldots, C \quad ,$$

$$x_i \in \{0, 1\} \quad , \qquad (15)$$
$$i = 1, \ldots, I \quad ,$$

$$y \geq 0 \quad , \qquad (16)$$

where $i = 1, \ldots, I$ are the individual items in the item bank, V_{ic} is the set of items i within cluster c, and x_c is the number of items to be included in the test from cluster c.

Computational Procedure

Integer programming problems can optimally be solved (e.g., Garfinkel & Nemhauser, 1972; Taha, 1975) by (1) computing the relaxed version (no integer constraints in Equations 9, 12, or 15) of the integer programming problem (Simplex algorithm); and (2) applying a branch-and-bound strategy to find the best integer solution.

Two approximation algorithms were examined in this study—the heuristic procedure of Adema, and the optimal rounding strategy of van der Linden and Boekkooi-Timminga (1989). The heuristic of Adema is described in detail by Adema (1989), and Adema, Boekkooi-Timminga, and van der Linden (in press). Both procedures adapt the original branch-and-bound procedure as described by Land and Doig (1960).

It is well known that the objective function value z_{LP} of the relaxed integer programming problem solution is an upper bound for the objective function value of the integer programming problem. Adema (1989) considered two adaptations. The first adaptation of the algorithm concerns fixing decision variables with large and small reduced costs at their lower (l_c) and upper bounds (u_c), respectively, after the relaxed integer programming solution is obtained. The reduced costs r_c indicate the decrease of the optimal objective function value when the value of a nonbasic variable is increased by one unit, provided the basis is not changed (Murtagh, 1981; Williams, 1978).

Two rules are used for fixing the decision variables:

if $z_{LP} - h_1 z_{LP} < r_c$, then $x_c = l_c$, and

if $z_{LP} - h_1 z_{LP} < -r_c$, then $x_c = u_c$,

where r_c is the reduced cost for cluster c, and h_1 ($h_1 < 1$) is a help variable whose value is chosen to be close to 1 by the test constructor. However, setting h_1 can easily be implemented in the computer program.

Adema's other adaptation uses the fact that the objective function values obtained from optimal integer programming and relaxed integer programming for test construction problems using the maximin model are very close. He exploits this fact by initializing z_+, the true lower bound of the optimal integer programming objective function value, by $z_+ = h_2 z_{LP}$ ($0 << h_2 < 1$; $h_1 > h_2$) instead of $z_+ = -\infty$. Then, the first integer solution having an objective function value z between $h_2 z_{LP}$ and z_{LP} is accepted. This algorithm should not be used when z_{LP} is equal to 0, as no solution will be obtained. It is possible that no solution is found if h_1 or h_2 is too large—if so, these values should be lowered.

Optimally rounding the relaxed integer programming solution is handled here slightly differently than in van der Linden and Boekkooi-Timminga (1989). For this study, all decision variables with reduced costs larger than 0 were fixed at their lower bounds, and those with reduced costs smaller than 0 were fixed at their upper bounds (this is actually the same as taking $h_1 = 1$ and $h_2 = 0$ in the Adema heuristic). Decision variables with fractional values always have reduced costs of 0. In addition, the branch-and-bound procedure is continued until the best solution is obtained. It is possible that no solution will be obtained, because the constraints cannot be met.

Computational Experiments

The cluster-based method tries to approximate methods based on 0-1 programming. The manner in which these experiments influenced the speed and accuracy of the cluster-based test construction method was studied; Stage 2 was not applied. The Adema heuristic with $h_1 = .999$ and $h_2 = .99$ was used, as well as the optimal rounding strategy taking $h_1 = 1$ and $h_2 = 0$.

Three experiments were conducted with the basic cluster-based method: (1) the objective function values and computing times were compared for the relaxed 0-1 programming approach and the cluster-based method; (2) the differences were examined between the actual objective function values computed after the individual items had been selected at random and those obtained from the cluster-based method, considering both the cluster information functions and the relaxed 0-1 programming approach; and (3) the effect of the upper bound u_c (Equation 3) on computing time was investigated.

Six test construction problems were analyzed. The test specifications for each problem are shown in Table 1. An item bank consisting of 1,000 items with item difficulty values drawn from $b \sim N(0,2)$ was used. The interval widths considered were .4, .3, .25, and .2. All computations were performed on an Olivetti M24 computer (8 MHz) with a math coprocessor, except for the relaxed 0-1 programming solutions, which were computed on a DEC-2060 computer because data storage limitations made it impossible to implement them on the Olivetti M24. The computing times for the Olivetti M24 included the time needed for the actual optimization and writing of the output file. For the DEC-2060, these times also included the time needed to read the input file.

A solution was accepted (indicated by an asterisk in the tables) when all constraints were met and an objective function value was obtained that did not differ more than 1% from its upper bounds z_{LP}. If the LP solution was accepted, it was the optimal solution, because it included only integer decision variable values. If a solution was found using the Adema heuristic, it was always accepted.

0-1 Programming Versus the Cluster-Based Method

For maximization problems, the objective function value for the relaxed integer programming problem is an upper bound to the objective function value of the optimal integer programming problem. Results for the Rasch model have shown that the objective function values of maximin test construction problems using 0-1 programming are very close to the objective function values of the corresponding relaxed problems (e.g., Boekkooi-Timminga, 1989). The same phenomenon was noted for the cluster-based method (compare z_{LP} and z^* in Table 3). Thus, a comparison of the upper bounds for the

Table 1
Objective Function Values (z_{LP}) and Differences in Percent Between z and z_{LP} of the 0-1 Solution (d_{01}) for the Relaxed Integer Solution (LP), Adema Solution (AD), and Optimally Rounded Solution (RD) of the Cluster-Based Approach, and the Solution to the Relaxed 0-1 Programming Problem ($I = 1{,}000$; $b \sim N(0,2)$; $u_c = \max_c$).

Problem and Width	LP		AD		RD	
	z_{LP}	d_{01}	z_{AD}	d_{01}	z_{RD}	d_{01}
Problem 1: $r_k = 1$ for $\theta_k = -3,-2,-1,0,1,2,3$; $N = 40$						
.4	4.1901	.3%	4.1599*	1.0%	4.1681*	.8%
.2	4.1985	.3%	4.1566*	1.1%	4.1580*	1.0%
0-1 Solution	4.2008					
Problem 2: $r_k = 1$ for $\theta_k = -3,-1,1,3$; $N = 40$						
.4	4.3481	.2%	4.3298*	.6%	4.3298*	.6%
.2	4.3563	0.0%	4.3161*	1.0%	4.3442*	.3%
0-1 Solution	4.3574					
Problem 3: $r_k = 1$ for $\theta_k = -2,0,2$; $N = 40$						
.4	5.3316	.3%	5.2824*	1.2%	5.3213*	.5%
.2	5.3428	.1%	5.2946*	1.0%	5.3185*	.6%
0-1 Solution	5.3484					
Problem 4: $r_k = 1$ for $\theta_k = -1,0,1$; $N = 40$						
.4	7.8596	0.0%	7.8089*	.7%	7.8453*	.2%
.2	7.8640	0.0%	7.8590*	0.0%	7.8590*	0.0%
0-1 Solution	7.8620					
Problem 5: $r_k = 10$ for $\theta_k = -2,2$; $r_k = 1$ for $\theta_k = 0$; $N = 40$						
.4	.5364	.3%	.5359*	.4%	.5360*	.3%
.2	.5378	0.0%	.5375*	.1%	.5370*	.2%
0-1 Solution	.5378					
Problem 6: $r_k = 1$ for $\theta_k = 0$; $N = 40$						
.4	9.9993*	.1%	9.9993*	.1%	9.9993*	.1%
.2	10.0000*	.1%	10.0000*	.1%	10.0000*	.1%
0-1 Solution	9.9913*					

*Accepted integer solution.

relaxed 0-1 programming method and the cluster-based method gives an indication how well the cluster-based method approximates the optimal 0-1 programming method.

Table 1 gives the objective function values z obtained for the six test construction problems. The relaxed integer (LP), Adema (AD), and optimal rounding (RD) solutions were obtained for the cluster-based method. Only the results for interval widths of .4 and .2 are included in the table. In addition, the differences between the objective function values (LP, AD, RD) and the objective function value for the relaxed 0-1 programming problem are given in percentages of the relaxed 0-1 programming problem. These differences were small for all problems; differences slightly larger than 1% were found only for Problems 1 and 3. Note that in the case of Problem 6, the optimal objective function values for the cluster-based method were higher than for the 0-1 programming problem; this was caused by the fact that the cluster-based method uses the cluster information function during the optimization process.

Table 2 summarizes the computing times for the problems in Table 1. For the cluster-based method, computing time was very low. The largest amount of time was needed for Problem 1 with interval width .2. In this case, however, the optimal rounded solution was also accepted.

Table 2
Computing Times (In Seconds) for the LP, AD, and RD Solutions of the Problems from Table 1 on an Olivetti M24 Computer, and for the Corresponding Relaxed 0-1 Solutions on a DEC-2060 Computer

Problem and Width	LP	AD	RD
Problem 1			
.4	2.90	15.50*	3.80*
.2	4.20	135.00*	5.10*
0-1 Solution	103.87		
Problem 2			
.4	1.70	4.60*	5.30*
.2	4.20	22.60*	6.50*
0-1 Solution	114.36		
Problem 3			
.4	1.30	3.30*	7.10*
.2	2.40	19.60*	3.20*
0-1 Solution	102.53		
Problem 4			
.4	1.30	2.50*	1.60*
.2	2.20	5.20*	2.80*
0-1 Solution	113.25		
Problem 5			
.4	.90	2.10*	1.20*
.2	1.90	6.50*	2.50*
0-1 Solution	69.28		
Problem 6			
.4	.80*	.80*	.80*
.2	1.20*	1.20*	1.20*
0-1 Solution	42.45*		

*Accepted integer solution.

The Actual Objective Function Values

After the numbers of items to be selected from each of the clusters had been determined in Stage 1, the individual items were selected in Stage 3. For Problems 1 to 6, comparisons were made between the actual objective function values after the individual items had been selected. Also compared were the objective function values z for the 0-1 programming problem, and the accepted integer solutions of the cluster-based method. The optimally-rounded solution was taken as the accepted solution when its objective function value did not differ more than 1% from the relaxed objective function value; otherwise, the Adema solution was taken. In the case of Problem 6, the relaxed solutions were integer solutions.

The widths of the cluster intervals were .4, .3, .25, and .2. The actual tests were selected in two different ways: at random, and such that the items selected from the clusters reflected the cluster information function poorly. The latter was determined as follows: (1) two tests were selected to include the items located at the upper or lower ends of the cluster intervals, respectively; (2) the test with the worst objective function value was considered. Table 3 summarizes the objective function values for the relaxed (z_{LP}), accepted (z^*), random (z_r), and worst (z_w) solutions. The differences are also included between the objective function values obtained both for the worst and the randomly selected tests, and the relaxed 0-1 programming problem (d_{01}).

It was found that random selection almost always resulted in fairly accurate solutions. Except for four cases, d_{01} was always smaller than 1%, and no difference was larger than 2%. For the worst tests, these differences were much larger (they varied between 0 and 6.5%). As could be expected for worst item selection, the best results were obtained for the smallest interval widths. For random item selection, this trend was less convincing.

The Effect of u_c on Computing Time

The effect of a change in the upper bound u_c on computing time was examined. An interval width of .3 was selected. The following five cases were examined:

1. u_c = max$_c$.
2. If max$_c \geq 10$, then $u_c = 10$,
   else u_c = max$_c$.
3. If max$_c \geq 20$, then $u_c = 10$,
   else u_c = trunc(max$_c$/2).
4. If max$_c \geq 50$, then $u_c = 10$,
   else u_c = trunc(max$_c$/5).
5. If max$_c \geq 5$, then $u_c = 5$,
   else u_c = max$_c$.

Going from Case 1 to 5, most item banks will show a decrease of u_c. For the six test construction problems, the computing times for finding the relaxed integer (LP), Adema (AD), and optimal rounded (RD) solutions are summarized

Table 3
Objective Function Values Computed After the Individual Items Were Selected for the Accepted Integer Solution (z^*) for the Worst Test (z_w) and Random Test (z_r), Results From the Corresponding Relaxed 0-1 Programming Problem (z_{LP}, and Differences (d) Between z_w or Z_r and z_{LP} of the 0-1 Problem in Percentages

Width	z_{LP}	z^*	z_w	$d_{w,>01}$	z_r	$d_{r,>01}$
Problem 1						
.4	4.1901	4.1681	4.0302	4.1%	4.1781	.5%
.3	4.1958	4.1743	4.0563	3.4%	4.1623	.9%
.25	4.2008	4.1960	4.1425	1.4%	4.1910	.2%
.2	4.1985	4.1580	4.0966	2.5%	4.1502	1.2%
0-1 Solution	4.2008					
Problem 2						
.4	4.3481	4.3298	4.1417	5.0%	4.3029	1.3%
.3	4.3539	4.3492	4.2120	3.3%	4.3505	.2%
.25	4.5348	4.3477	4.2320	2.9%	4.3493	.2%
.2	4.3563	4.3442	4.3047	1.2%	4.3452	.3%
0-1 Solution	4.3574					
Problem 3						
.4	5.3316	5.3213	5.1078	4.5%	5.3044	.8%
.3	5.3324	5.3010	5.1815	3.1%	5.2949	1.0%
.25	5.3398	5.2873	5.1930	2.9%	5.2400	2.0%
.2	5.3428	5.3185	5.2571	1.7%	5.3225	.5%
0-1 Solution	5.3484					
Problem 4						
.4	7.8596	7.8453	7.3487	6.5%	7.7952	.9%
.3	7.8642	7.8596	7.5536	3.9%	7.8317	.4%
.25	7.8624	7.8618	7.6442	2.8%	7.8124	.6%
.2	7.8640	7.8590	7.7756	1.1%	7.8532	.1%
0-1 Solution	7.8620					
Problem 5						
.4	.5364	.5360	.5307	1.3%	.5357	.4%
.3	.5378	.5378	.5200	3.3%	.5367	.2%
.25	.5371	.5361	.5292	1.6%	.5364	.3%
.2	.5378	.5370	.5339	.7%	.5370	.2%
0-1 Solution	.5378					
Problem 6						
.4	9.9993	9.9993	9.9461	.5%	9.9570	.3%
.3	10.0000	10.0000	9.9773	.1%	9.9789	.1%
.25	9.9996	9.9996	9.9857	.1%	9.9911	0.0%
.2	10.0000	10.0000	9.9905	0.0%	9.9913	0.0%
0-1 Solution	9.9913					

in Table 4. The table shows that a decrease of u_c generally resulted in a small increase of computing time. This occurred because fewer items were allowed to be selected from the clusters than desired. For example, the solution to Case 1 for Problems 4 and 6 showed that 40 items were desired to be selected from the middle cluster, but it was never allowed to select more than 10 items from the same cluster for the other cases.

Conclusions

These results indicate that the basic model for the cluster-based test construction method works well, in terms of both computing time and accuracy. A small increase in computing time was noted when the maximum number of items to

Table 4
Effect of Varying u_c on Computing Time in Seconds for the Relaxed Integer Solution (LP), Adema Solution (AD), and Optimally Rounded Solution (RD)

Case	LP	AD	RD
Problem 1			
1	3.30	22.10*	4.80*
2	3.70	15.30*	5.70
3	3.80	16.10*	5.00
4	3.90	25.10*	4.80
5	5.30	25.80*	6.70
Problem 2			
1	2.70	11.90*	3.50*
2	3.40	20.70*	4.50*
3	2.70	19.70*	3.80*
4	3.00	14.80*	4.50*
5	4.80	16.40*	5.70*
Problem 3			
1	1.70	33.90*	5.00*
2	2.10	5.20*	3.80*
3	2.40	5.50*	4.10*
4	2.40	5.30*	3.60*
5	2.50	3.80*	3.30*
Problem 4			
1	1.70	3.30*	2.00*
2	2.90	4.30*	3.10*
3	3.00	4.40*	3.20*
4	2.90	4.40*	3.20*
5	3.10	3.80*	3.50*
Problem 5			
1	1.30	2.40*	1.80*
2	1.90	3.60*	2.30*
3	1.90	3.70*	2.50*
4	2.00	5.50*	2.40*
5	2.00	3.30*	2.40*
Problem 6			
1	.90*	.90*	.90*
2	1.20*	1.20*	1.20*
3	1.50*	1.50*	1.50*
4	1.80*	1.80*	1.80*
5	1.90*	1.90*	1.90*

*Accepted integer solution.

be included in the test from each cluster was decreased. When the cluster intervals were narrowed, the amount of computing time increased because of the larger number of decision variables. This also caused an increase in accuracy; however, for interval widths up to .4, the maximum difference between the "worst" test selected from the clusters and the relaxed solution for the 0-1 programming problem was smaller than 5% most of the time. In cases in which the tests were randomly selected from the clusters, the differences were even smaller (< 2%).

For most problems, the optimal rounded solution was accepted. When these optimal rounded solutions were compared to the rounded relaxed solutions (which was not described here), they were the same for most problems. For Problem 6, the relaxed solution always gave an integer result. This occurred because only one θ level was considered. Thus, the $N = 40$ items with difficulty levels closest to this θ level were selected. The same experiments were carried out for item banks with difficulties taken from the distributions $b \sim \mathrm{N}(0,1)$ and $b \sim \mathrm{U}(-3,3)$. However, no major differences could be noted, either in computing time or in accuracy.

Example Applications

Construction of a Selection and a Diagnostic Test

For the following example, as well as the one below, the same item bank was considered—a Rasch item bank of 1,000 items with difficulty parameters drawn from $b \sim \mathrm{N}(0,2)$ and a cluster width of .25. The item bank covered 25 subject matter aspects. The distribution of the items over subject matter aspects and clusters is given in Table 5.

Two tests were constructed: one for selection (Test A), and the other for diagnostic purposes (Test B). For Test A, maximum information was required at $\theta = 1$. In addition, each of the subject matter aspects 15 to 20 had to be represented by five items in the test. For Test B, the relative information values were required to be the same at all levels $\theta = -1, 0, 1$. Hence, $r_1 = r_2 = r_3 = 1$. From each of the 10 subject matter aspects 8 to 10, 12 to 15, and 17 to 19, three items had to be included in the test. Finally, it was required that the total test administration time, T, for Tests A and B be as close as possible to 150 minutes. The item administration times (in minutes) were generated from a uniform distribution $t \sim \mathrm{U}(2,12)$.

Table 5
Maximum Number of Items to be Selected from Subject Matter Aspect j Within Cluster c, Maximum Number of Items to be Selected from Cluster c ($\max_c$), and Maximum Number of Items to be Selected from Subject Matter Aspect j ($\max_j$) [I = 1,000; $b \sim N(0,2)$; Width = .25]

Cluster Lower Bound	Subject Matter Aspect																									
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	19	20	21	22	23	24	25		$\max_c$
-3.125	10	6	3																							19
-2.875	2	6	3	1																						12
-2.625		2	7	3	1																					13
-2.375		2	6	8	5	3																				24
-2.125			1	5	10	8	3																			27
-1.875				6	10	12	10	3																		41
-1.625					3	10	12	10	3																	38
-1.375						4	15	16	15	5																55
-1.125							3	15	20	15	3															56
-.875								5	10	15	10	5														45
-.625									5	15	19	15	5													59
-.375										10	15	29	15	10												79
-.125											6	10	15	15	10	6										62
.125												10	15	19	15	8	2									69
.375													10	15	26	15	10									76
.625														5	10	22	10	5								52
.875															5	15	23	15	5							63
1.125																5	10	18	10	5						48
1.375																	6	6	15	8	5					40
1.625																		3	6	18	8	3				38
1.875																			2	5	12	5	3			27
2.125																					4	10	5			19
2.375																						2	8	3		13
2.625																							1	4	1	6
2.875																							4	5	10	19
$\max_j$	12	16	20	23	29	37	43	49	53	60	53	69	60	64	66	71	61	47	38	36	29	20	21	12	11	

In Stage 2, the clusters from which items had to be selected were partitioned according to their item administration times. Each cluster consisted of five partitions with mean item administration times of 3, 5, 7, 9, and 11. The objective function $z = d_1 + d_2$ was used because the constraint for the test administration time was

$$\sum_c \sum_j \sum_i t_i y_{cji} = T + d_1 - d_2 \quad . \tag{17}$$

Table 6 summarizes the characteristics of both tests constructed. Computation times for the selection test are smallest because only one θ level had to considered. The test administration time for the diagnostic test was 150 minutes, whereas the selection test took 174 minutes. In order to find a selection test with a shorter test administration time, the u_cs in the test construction model for Stage 1 should be lowered, such that the items will be selected from different clusters. There will then be a greater chance of finding a test with a shorter test administration time in Stage 2. Note the small difference between z_r and z^* (see also Table 3).

The Construction of Four Weakly-Parallel Tests

Tests are considered to be weakly parallel if their information functions are identical (Samejima, 1977). Four weakly-parallel tests were determined, having the same target information function as Test B in the previous example. The tests were constructed simultaneously and

Table 6
Computing Time (in Seconds), y for the Accepted Integer Solution (z^*), y for a Randomly Selected Test (z_r), and Total Test Administration Time in Minutes for the Diagnostic and Selection Test

	Computing Time				Total
Test	Stage 1	Stage 2	z^*	z_r	Time
Selection	4.70*	7.30*	7.481	7.447	174
Diagnostic	17.70*	34.80*	5.747	5.585	150

*Optimally rounded solution fit all requirements and was accepted.

sequentially.

For simultaneous test construction, the model was adapted slightly—u_{cj} in Equation 8 was divided by the number of tests to be constructed. After determining the number of items to be selected from each cluster, the tests were randomly selected from these clusters. When the tests were constructed sequentially, the same test construction model was used four times, adapting u_{cj} after each run. Table 7 summarizes the characteristics of the selected tests.

Table 7
Computing Time (in Seconds), y for the Accepted Integer Solution (z^*), and y Computed for a Randomly Selected Test (z_r), for Four Parallel Tests Constructed Simultaneously and Sequentially

Test	Computing Time	z^*	z_r
Simultaneously			
1	26.80	5.655*	5.562
2		5.655*	5.560
3		5.655*	5.564
4		5.655*	5.561
Sequentially			
1	17.70	5.747*	5.690
2	19.20	5.709*	5.615
3	17.70	5.638*	5.546
4	18.90	5.624*	5.522

*Optimally rounded solution fit all requirements and was accepted.

The tests were better when weakly parallel, and less computer time was needed when they were constructed simultaneously. For simultaneous test construction, test content was also more parallel, because the same numbers of items for each test were taken from the same item aspect groups of the same clusters. In the simultaneous case, however, there is a larger chance of not finding a solution because the problem is more constrained.

Conclusions

A new test construction procedure for the Rasch model that uses integer programming was described. With this new test construction procedure, tests fitting the requirements can be selected quickly from large item banks using a microcomputer. The main advantages of the method are the minimal amounts of computer time and data storage needed in comparison to the traditional 0-1 programming approach.

Two heuristic procedures were used in the computational experiments. The optimal rounding strategy produced acceptable results most of the time. However, rounding the fractional decision variable values of the relaxed integer solution to the nearest integers often gave acceptable results, as well. It is thus advisable to examine the rounded relaxed integer solution first, and if it is not acceptable, then the optimally rounded solution should be obtained. If this solution is not acceptable, then the Adema solution should be computed.

A cautionary remark is in order. If only the first stage of the test construction process is used, computing time will be short. However, if several additional test specifications need to be considered in the second stage, computing time will increase rapidly because of the rapid increase in the number of decision variables. The method will be most valuable for test construction problems that consider a limited number of item characteristics. Because it is expected that the actual number of item characteristics considered for test construction for unidimensional item banks will be small, this feature is not too problematic.

Although the method described here is not able to consider all types of practical constraints, it has some useful properties. First, it is easy to construct parallel tests by adapting the model slightly (see Example 2). Computational results for linear programming methods show that there is a trend toward selecting large numbers of items with approximately the same difficulty (Baker, Cohen, & Barmish, 1988; de Gruijter, 1990), which is usually not desirable in practice. Using the present method, this problem can easily be precluded by setting low u_c values.

The cluster-based method can also be applied to two- or three-parameter logistic models. However, it is expected that many clusters will be needed in order to be able to consider the items in a cluster as equivalent. As a result, the problem might degenerate to the corresponding 0-1 programming problem. The primary advantages of the present method are its speed and its low computer storage and memory requirements, in comparison to standard (0,1) linear programming methods.

References

Adema, J. J. (1989). *Implementations of the branch-and-bound method for test construction problems* (Research Report 89-6). Enschede, University of Twente, The Netherlands, Department of Education.

Adema, J. J. (1990). The construction of customized two-stage tests. *Journal of Educational Measurement, 27,* 241–253.

Adema, J. J., Boekkooi-Timminga, E., & van der Linden, W. J. (in press). Achievement test construction using 0-1 linear programming. *European Journal of Operations Research.*

Baker, F. B., Cohen, A. S., & Barmish, B. R. (1988). Item characteristics of tests constructed by linear programming. *Applied Psychological Measurement, 12,* 189–199.

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

Boekkooi-Timminga, E. (1987). Simultaneous test construction by zero-one programming. *Methodika, 1,* 101–112.

Boekkooi-Timminga, E. (1989). *Models for computerized test construction.* De Lier, The Netherlands: Academisch Boeken Centrum.

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

de Gruijter, D. N. M. (1990). Test construction by means of linear programming. *Applied Psychological Measurement, 14,* 175–181.

Garfinkel, R. S., & Nemhauser, G. L. (1972). *Integer programming.* New York: Wiley

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

Lenstra, J. K., & Rinnooy Kan, A. H. G. (1979). Computational complexity of discrete optimization problems. In P. L. Hammer, E. L. Johnson, & B. H. Korte (Eds.), *Discrete optimization I.* New York: North-Holland.

Lord, F. M. (1977). Practical applications of item characteristic curve theory. *Journal of Educational Measurement, 14,* 117–138.

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

Murtagh, B. A. (1981). *Advanced linear programming: Computation and practice.* New York: McGraw-Hill.

Rasch, G. (1960). *Probabilistic models for some intelligence and attainment tests.* Copenhagen: Danish Institute for Educational Research.

Samejima, F. (1977). Weakly parallel tests in latent trait theory with some criticisms of classical test theory. *Psychometrika, 42,* 193–198.

Taha, H. A. (1975). *Integer programming.* New York: Academic Press.

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, 54,* 237–247.

Williams, H. P. (1978). *Model building in mathematical programming.* New York: Wiley.

Acknowledgments

The author gratefully acknowledges the Netherlands Organization for Scientific Research (N.W.O.) for funding this project. This research was conducted while the author was supported by a PSYCHON *grant from this organization (560-267-001), awarded to W. J. van der Linden. Thanks are also due to Wim J. van der Linden and Jos J. Adema, Department of Education, University of Twente, for their helpful comments during the preparation of this manuscript.*

Author's Address

Send requests for reprints or further information to Ellen Boekkooi-Timminga, University of Twente, Department of Education, P.O. Box 217, 7500 AE Enschede, The Netherlands.