Adema, Jos J.
41p.; For a related document, see TM 015 953.
Bibliothek, Department of Education, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands.
Reports - Evaluative/Feasibility (142)
MF01/PC02 Plus Postage.
*College Entrance Examinations; *Computer Assisted Testing; Equations (Mathematics); Foreign Countries; Higher Education; *Item Banks; Item Response Theory; Mathematical Models; Mathematics Tests; Student Placement; *Test Construction; Test Items
Information Function (Tests); *Mathematical Programming; Maximin Model; *Parallel Test Forms; Placement Tests

Data banks with items calibrated under an item response model can be used for the construction of tests. Mathematical programming models like the Maximin Model are formulated for computerized item selection from a bank. In this paper, mathematical programming models based on the Maximin Model are proposed for the construction of weakly parallel tests. Numerical experiments were conducted to obtain an impression of the practicality of the approach using an item bank of 600 items from college placement mathematics examinations (520 items were from 13 previously administered American College Testing Assessment Program tests, and 80 were from the Collegiate Mathematics Placement Program). Six tests were constructed, and their test information functions were computed. The results demonstrate that tests constructed with the proposed models were near-optimal with respect to the Maximin criteria and were approximately weakly parallel. Five tables present information about the constructed tests. Four graphs illustrate the information functions. (Author/SLD)
The Construction of Weakly Parallel Tests by Mathematical Programming

Jos J. Adema
The Construction of Weakly Parallel Tests

by Mathematical Programming

Jos J. Adema
The construction of weakly parallel tests by mathematical programming, Jos J. Adema - Enschede: University of Twente, Department of Education, September, 1990. - 34 pages
Abstract

Databanks with items calibrated under an item response model can be used for the construction of tests. Mathematical programming models like the Maximin Model are formulated for computerized item selection from a bank. In this paper mathematical programming models based on the Maximin Model are proposed for the construction of weakly parallel tests. Numerical experiments have shown that tests constructed with the proposed models are near-optimal with respect to the maximin criterion and are approximately weakly parallel.

Keywords: Item Banking; Test Construction; Mathematical Programming; Weakly Parallel Tests.
A new development in item response theory is the construction of tests by mathematical programming models (e.g., Adema & van der Linden, 1989; Baker, Cohen & Barmish, 1988; Boekkooi-Timminga, 1989; Theunissen, 1985; van der Linden & Boekkooi-Timminga, 1989). These models are used to select items from a bank calibrated under an item response model, such that the constructed test is in some sense (approximately) optimal. The models also take into account that all the demands of the test constructor with respect to, for instance, test composition and administration time are satisfied. Boekkooi-Timminga (1987, 1990) has proposed a number of mathematical programming models for the construction of weakly parallel tests. Samejima (1977) defines two weakly parallel test forms as "a pair of tests which measure the same ability and whose test information functions are identical".

Two main approaches can be distinguished in the construction of weakly parallel tests by mathematical programming: (1) Sequential and (2) Simultaneous test construction (Boekkooi-Timminga, 1990). In the sequential case the tests are constructed one after the other. Each time the best items are selected, which implies that the psychometric quality of the tests is likely to decrease in the order of construction. For the Rasch model the decrement in quality can be small, but for the 3-parameter model the
decrement in general is large. So the disadvantage of this approach is that the constructed tests are not always fully parallel. In the simultaneous case all the tests are constructed at the same time. Simultaneous constructed tests are approximately weakly parallel (exact weakly parallel tests do not exist in practice because test information functions are never equal). The drawback of the simultaneous approach are the large amount of computer time and storage required.

In this paper mathematical models are proposed that allow weakly parallel tests to be constructed sequentially. The advantage of the new approach is that the constructed tests are approximately weakly parallel and that it is not as time and storage consuming as simultaneous test construction.

In most of the test construction models available the test information function plays an important role. The test information function for an unbiased estimator of ability is defined as the reciprocal of the (asymptotic) sampling variance of the estimator (Lord, 1980) which makes it a measure of the quality of a test. Also the feature that the test information function can be computed by addition of the item information function is very useful:

\[ I(\theta) = \sum_{i=1}^{n} I_i(\theta), \]

where \( \theta \) is the ability parameter, \( n \) the number of items in the test, and \( I_i(\theta) \) the information function of item \( i \).
The mathematical programming models as formulated in this paper are based on the Maximin Model (van der Linden & Boekkooi-Timminga, 1989), which is applicable for the construction of one test at a time. A brief review of the Maximin Model is given here. Associated with each item is a decision variable $x_i$ such that

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

where $I$ is the number of items in the item bank. The information function of the test to be constructed is only considered at a number of ability levels $\theta_k$, $k = 1, \ldots, K$. The test constructor can choose both the number and spacing of these levels. Let $I_i(\theta_k)$ denote the information function value of item $i$ at ability level $\theta_k$. The test constructor has to specify the relative shape of the target test information function by choosing constants $r_k$, $k = 1, \ldots, K$. Let $y$ be decision variable such that $(r_1y, \ldots, r_Ky)$ is a series of lower bounds to the test information function at the ability levels $\theta_k$. If $n$ is the prescribed number of items in the test then the Maximin Model is formulated as follows:

(1) \quad \text{Maximize } y,$$

subject to

...
Weakly Parallel Tests

(2) \[ \sum_{i=1}^{I} I_i(\theta_k)x_i - r_ky \geq 0, \quad k = 1, \ldots, K, \]

(3) \[ \sum_{i=1}^{I} x_i = r, \]

(4) \[ \sum_{i=1}^{I} a_{ij}x_i = b_j, \quad j = 1, \ldots, J, \]

(5) \[ x_i \in \{0,1\}, \quad i = 1, \ldots, I, \]

(6) \[ y \geq 0. \]

Constraints (4) are a general provision for practical constraints such as constraints on the administration time, test composition etc. By maximizing \( y \) in the objective function (1) the lower bounds \((r_1y, \ldots, r_Ky)\) are forced to be as high as possible (max-part). By imposing the constraints (2) \((r_1y, \ldots, r_Ky)\) is a series of lower bounds to the test information function (min-part). The number of items in the test is controlled by constrains (3).

The Maximin Model is known in the operations research literature (e.g. Wagner, 1975; Hartley, 1985) as a mixed integer linear programming (MILP) model, because the objective function and constraints are linear in the decision variables and there are continuous \((y)\) as well as integer \((x_i)\) variables in the model. A well-known method for solving MILP models is the branch-and-bound method (Land & Doig,
1960). In its standard form this method is time-consuming. Adema, Boekkooi-Timminga, and van der Linden (in press) have proposed a heuristic based on the branch-and-bound method for solving test construction models, that solves the Maximum Model for large item banks in favorable time. The heuristic will be used in forthcoming numerical experiments.

In the next section a mathematical programming model for simultaneous test construction is formulated. The model is not recommended for practical application, but will be used to evaluate the quality of forthcoming models in the discussion section. Then, mathematical programming models are presented to construct tests sequentially. Next, these models are used in numerical experiments to get an impression about the practicability (CPU-time and accuracy) of the approach presented in this paper. The results of the experiments are evaluated in the discussion section.

A Simultaneous Test Construction Model

In this section a mixed integer linear programming model is presented for constructing tests simultaneously.

Suppose $T$ tests have to be constructed and define the decision variables $x_{it}$ as:

$$x_{it} = \begin{cases} 
0 & \text{item } i \text{ not in test } t \\
1 & \text{item } i \text{ in test } t.
\end{cases}$$

$i = 1, \ldots, I$; $t = 1, \ldots, T$
The Maximin Model can easily be extended to the case of simultaneous test construction by taking \((r_1y, \ldots, r_Ky)\) as a series of lower bounds for the test information function of all \(T\) tests:

\[
\text{(7) Maximize } y, \\
\text{subject to}
\]

\[
\sum_{i=1}^{I} I_i(\theta_k)x_{it} - r_ky \geq 0, \quad k = 1, \ldots, K; \\
t = 1, \ldots, T,
\]

\[
\sum_{i=1}^{I} x_{it} = n, \quad t = 1, \ldots, T,
\]

\[
\sum_{i=1}^{I} a_{ij}x_{it} = b_j, \quad t = 1, \ldots, T; \\
j = 1, \ldots, J,
\]

\[
\sum_{t=1}^{T} x_{it} \leq 1, \quad i = 1, \ldots, I,
\]

\[
x_{it} \in \{0, 1\}, \quad i = 1, \ldots, I; \\
t = 1, \ldots, T,
\]

\[
y \geq 0.
\]

Constraints (8) imply that \((r_1y, \ldots, r_Ky)\) is a series of lower bounds for all \(T\) test information functions. The number
of items in the tests are equal to \( n \) by constraints (9). Constraints (10) are a general notation for possible practical constraints. To preclude that items are selected for more than one test constraints (11) are imposed.

The main disadvantage of model (7)-(13) is that the number of variables and constraints increases rapidly if the number of tests increases. Thus solving model (7)-(13) can be time consuming and cost a lot of computer storage. In the next section MILP models with results comparable to model (7)-(13) are formulated. The new MILP models are, however, less time and computer storage consuming than model (7)-(13).

The results of model (7)-(13) will not always yield (approximately) weakly parallel tests, because the test information functions are only bounded from below. However, model (7)-(13) can be generalized such that bounds from above are included (Boekkooi-Timminga, 1990). As model (7)-(13) is a generalized version of all other simultaneous models including bounds from above, its objective function will be highest. This objective function value will be used for evaluating the quality of the forthcoming models in the discussion section.

New Models for the Construction of Weakly Parallel Tests

In this section MILP models for constructing weakly parallel tests sequentially are presented. These models contain extra constraints that, at previous stages, allow for tests to be constructed at later stages.
Next, the MILP model for constructing the first test is formulated. The basic idea is as follows: Suppose T tests have to be constructed. The model for the first test is used for constructing two tests, namely, the first test and a dummy test that is T-1 times the size of this first test. In the latter test the decision variables are allowed to take non-integer values. The large test actually represents the T-1 tests that have to be constructed after the first test. The decision variables corresponding to the first test are

\[ x_i = \begin{cases} 
0 & \text{item } i \text{ not in the first test} \\
1 & \text{item } i \text{ in the first test}
\end{cases} \]

and for the other test the decision variables \( z_i \) are introduced, where \( z_i \) is the fraction of item \( i \) in the large test. The model is as follows:

\[
\begin{align*}
(14) \quad & \text{Maximize } y, \\
(15) \quad & \sum_{i=1}^{I} I_i(\theta_k)x_i - r_ky \geq 0, \quad k = 1, \ldots, K, \\
(16) \quad & \sum_{i=1}^{I} x_i = n,
\end{align*}
\]
Weakly Parallel Tests

$$\sum_{i=1}^{I} a_{ij} x_i = b_j, \quad j = 1, \ldots, J,$$

$$\sum_{i=1}^{I} I_i (\theta_k) z_i - (T-1) r_k y \geq 0, \quad k = 1, \ldots, K,$$

$$\sum_{i=1}^{I} z_i = (T-1) n, \quad i = 1, \ldots, I,$$

$$\sum_{i=1}^{I} a_{ij} z_i = (T-1) b_j, \quad j = 1, \ldots, J,$$

$$x_i + z_i \leq 1, \quad i = 1, \ldots, I,$$

$$x_i \in \{0, 1\}, \quad i = 1, \ldots, I,$$

$$z_i \geq 0, \quad i = 1, \ldots, I,$$

$$y \geq 0.$$

The decision variables $x_i$ in model (14)-(24) denote which items are selected for the first test. The objective function (14) and constraints (15)-(17), (22) and (24) give the basic Maximin Model. By inclusion of constraints (18)-(21), and (23) the best items are prevented from being selected only in the first test. Constraints (21) imply $z_i \leq 1$ and prevent items from being selected for the first and the dummy test.
Weakly Parallel Tests

For the construction of the second test model, (14)-(24) is applied again with $T-2$ instead of $T-1$ and with the items selected for the first test deleted from the bank. The decision variables now denote which items will be selected for the second test.

In an analogous way the remaining tests are constructed. If in constraints (18)-(20) $T-1$ is replaced by $T-t$, the complete test construction procedure is as given by Algorithm A:

Algorithm A

Step 1: $t := 1$;
Step 2: Solve model (14)-(24); The selected items from the item bank represent test $t$.
Step 3: If $t = T$ then STOP else delete the selected items from the item bank, $t := t+1$ and go to Step 2.

The tests constructed according to the above approach are not necessarily weakly parallel tests, because the test information functions are not bounded from above. The problem occurs when for one or more $\theta_k$'s the chosen $r_k$'s are such low that the corresponding constraints in (15) and (18) in the relaxed model (14)-(24) are not active, i.e., that for the optimal solution to model (14)-(24) with $0 \leq x_i \leq 1$ instead of $x_i \in \{0,1\}$ the test information function value is above $r_k\gamma$ for some $\theta_k$'s. To resolve this problem Algorithm B is proposed. In this algorithm the values of $r_k$ that are too low
are increased such that the constraints in (15) and (18) are more restrictive:

Algorithm B

Step 1: \( t := 1; \)
Step 2: Solve the relaxed model (14)-(24);
Step 3: Adjust \( r_k, \ k = 1, \ldots, K; \)

\[
r_k = \left( \sum_{i=1}^{I} I_i(\theta_k)x_i + \sum_{i=1}^{I} I_i(\theta_k)z_i \right) (Ty)^{-1},
\]

where \( x_i \ (i = 1, \ldots, I), \ z_i \ (i = 1, \ldots, I), \) and \( y \) are computed in Step 2;
Step 4: Solve model (14)-(24). The selected items from the item bank represent test \( t; \)
Step 5: If \( t = T \) then STOP else delete the selected items from the item bank, \( t := t+1 \) and go to Step 4.

In Step 3 the value of \( r_k \) is adjusted only at ability levels \( \theta_k \) with \( \Sigma I_i(\theta_k)x_i > r_ky \) and/or \( \Sigma I_i(\theta_k)z_i > (T-1)r_ky \) such that large differences between test information functions are less likely, because of the more restrictive constraints (15) and (18).

To make the approach even better - but also more time consuming because the model will be more restricted - it is possible to put an upper bound on
How to impose this upper bound is explained next. Let $e_k$, $k = 1, \ldots, K$ be decision variables that are equal to the test information function value at $\theta_k$ minus $r_ky$. Then it is required that:

\begin{equation}
\sum_{i=1}^{I} I_i(\theta_k)x_i - r_ky = e_k = 0, \quad k = 1, \ldots, K.
\end{equation}

If constraints (15) are replaced by constraints (25), the test information function can be bounded from above by imposing upper bounds on the variables $e_k$:

\begin{equation}
0 \leq e_k \leq E_k, \quad k = 1, \ldots, K,
\end{equation}

where $E_k$ is a prespecified upper bound that guarantees a required precision.

Numerical Experience

It is hard to solve mixed integer linear programming models in general. Therefore, the heuristic as proposed by Adema (1988) will be used in this section. The heuristic is based on the branch-and-bound (BAB) method (Land & Doig, 1960). A full explanation of the heuristic is beyond the scope of this paper. However, two important parameters ($H_1$ and $H_2$) are
Weakly Parallel Tests

15

described here. The branch-and-bound method as well as the heuristic start with solving the relaxed model \(0 \leq x_i \leq 1\) instead of \(x_i \in \{0,1\}\) by standard linear programming. In the heuristic the model is, then, reduced by fixing variables according to the following rules (\(H_1 < 1\) is prespecified):

1. Fix \(x_i\) to 0, if in the relaxed solution \(x_i = 0\) and \((1 - H_1)z_{LP} < d_i\), where \(z_{LP}\) is the objective function value for the optimal solution to the relaxed model and \(d_i\) the reduced cost of variable \(x_i\) (See e.g. Murtagh, 1981, p.25);

2. Fix \(x_i\) to 1, if in the relaxed solution \(x_i = 1\) and \((1 - H_1)z_{LP} < -d_i\).

Another feature of the heuristic is that the difference between \(z_{LP}\) and the objective function value of the optimal 0-1 solution is required to be smaller than \((1 - H_2)*100\%\) of \(z_{LP}\), where \(H_2 < 1\) is also prespecified. The above modifications speed up the search process after the relaxed model is solved. A last modification is to stop as soon as an 0-1 solution - not necessarily the optimal 0-1 solution - has been found. The objective function value of the 0-1 solution found will be between \(H_2*z_{LP}\) and \(z_{LP}\). So if \(H_2\) is close to 1 the heuristic will give a nearly optimal 0-1 solution.

MPSX/370 V2 is an IBM licensed program for handling linear and mixed integer linear programming problems (IBM MPSX/370 V2 Program Reference Manual, 1988). It provides the user with algorithmic tools which enable him/her to build his/her own heuristics and algorithms by writing a control program in ECL, a computer programming language based on
Weakly Parallel Tests

16

PL\l. The heuristic together with Algorithms A and B were implemented in ECL programs. The CPU-times in the forthcoming tables are the execution times of the ECL programs on an IBM9370 computer.

A short description of the item bank used in the experiments is given below. Ackerman (1989) gives a more detailed description of the item bank. It consisted of 600 items; 520 items were from 13 previously administered ACT Assessment Program (AAP) tests and 80 were from the Collegiate Mathematics Placement Program (CMMP). The items were calibrated under the 3-parameter logistic model (Birnbaum, 1968). Thus, the probability of an examinee with ability level $\theta$ to answer an item $i$ correctly was given by:

$$P_i(\theta) = c_i + (1-c_i) \left(1 + \exp(-D a_i(\theta-b_i))\right)^{-1},$$

where $D$ is a constant equal to 1.7 and $a_i$, $b_i$, and $c_i$ are the discrimination, difficulty, and guessing parameter of item $i$. The information function is expressed by:

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

The bank was partitioned in six content areas:

1. Arithmetic and Algebraic Operations (AAO);
2. Arithmetic and Algebraic Reasoning (AAR);
3. Geometry (G);
4. Intermediate Algebra (IA);
(5) Number and Numeration Concepts (NNS);
(6) Advanced Topics (AT).

From the bank items were selected to create weakly parallel tests with 40 items (4 AAO items, 14 AAR items, 8 G items, 8 IA items, 4 NNS items, and 2 AT items). The six ability levels and relative information function values in the test construction models \((\theta_k, r_k)\) were: \((-1.6, 2.0), (-.8, 5.4), (.0, 12.1), (.8, 21.3), (1.6, 10.8),\) and \((2.4, 3.1)\).

Tests were constructed using model (14)-(24) (Algorithm A) according to the above test specifications. In Table 1 the differences between the information function values of the tests giving the most and least information at \(\theta_k\), \(k = 1, \ldots, 6\) are given for \(T\) ranging from 2 to 6. The \(y_{\text{min}}\) value in the table is the lowest objective function value found for the \(T\) constructed tests, i.e., \(r_k y_{\text{min}}\) is a lower bound for the test information function \(a^\ast \theta_k\) for all \(T\) tests.

Insert Table 1 here

For the sake of illustration, the test information functions for \(T = 6\) are shown in Figure 1.

Insert Figure 1 here
The large difference found for $\theta_5 = 1.6$ in Table 1 is caused by the steepness of the information functions at the ability level, as can be seen in Figure 1.

Table 2 is similar to Table 1. However, in this case Algorithm B was used. The adjusted $r_k$ values ($r_k'$) are given in Table 3.

Figure 2 depicts the test information functions for $T = 6$.

The test giving most information at $\theta_6 = 2.4$ contained an item with item information function value 1.810 at this ability level. This item is responsible for the large differences at $\theta_6$ for $T = 5$ and 6.

In Table 4 the results are displayed for the case of the $r_k$ values adjusted and upper bounds (See constraints (25) and (26)) imposed. The upper bounds $E_k$, $k = 1, \ldots, K$, were computed after Step 2 of Algorithm B was executed as:

$$E_k = 0.025z_{LPR_4}.$$
In specifying the upper bounds $r_4$ was used, because at $\theta_4$ most information was wanted. Other ways of specifying the upper bounds are also feasible and may give good results.

Table 4 shows that imposing upper bounds is very effective in making the tests more weakly parallel.

Discussion

In this section three points are discussed: a) How good are the constructed tests with respect to the Maximin criterion?; b) How weakly parallel are the tests?; c) How much CPU-time is needed to solve the models?

Let $y'$ be the objective function value for the optimal solution of the relaxed model (7)-(13) and $y_{0-1}$ be this value for the optimal 0-1 solution to model (7)-(13). Then $y'$ is an upper bound for $y_{0-1}$ and $y_{0-1}$ is an upper bound for $y_{\text{min}}$.
because if the T solutions found by Algorithm A or B are combined they give a 0-1 solution to model (7)-(13), but not necessarily the best one. Hence, \( \gamma_{\text{min}} \), the worst of these subsolutions, is certainly smaller than \( \gamma_{0-1} \):

\[
y' \geq \gamma_{0-1} \geq \gamma_{\text{min}}.
\]

On the other hand the objective function value \( \gamma'' \), for the optimal solution to the relaxed model (14)-(24) for \( t = 1 \), is an upper bound for \( y' \), because every solution to model (7)-(13) is a solution to model (14)-(24). Implying:

\[
y'' \geq y' \geq \gamma_{0-1} \geq \gamma_{\text{min}}.
\]

The values of \( \gamma'' \) were computed during the test construction process. They are shown in Table 5 together with the differences between \( \gamma'' \) and \( \gamma_{\text{min}} \) in percentages of the first.

The differences are rather small given the fact that \( \gamma'' \) is an upper bound and \( \gamma_{\text{min}} \) is the minimum taken over the T constructed tests.

Ackerman (1989) has constructed 6 parallel tests by a simple heuristic with the item bank used here. He does not apply the Maximin criterion but has a fixed target test.
information function, therefore comparisons are not possible. However, a comparison with respect to the second question in this section can be made.

Insert Figure 4

Figure 4 is a reproduction of the figure of the 6 information functions in the paper of Ackerman. The results in Figure 1 are not as good as the results by Ackerman. Figures 2 and 3, however, are an improvement on the Ackerman results. Especially when upper bounds (constraints (25) and (26)) are imposed, the tests in this paper are practically strictly weakly parallel.

The automated test construction procedure presented here is time consuming, but the construction of weakly parallel tests by hand would be more time consuming, if not impossible. Also, in practice some of the CPU-time needed for constructing the tests can be regained, because tests can be printed and inspected while other tests are still being constructed. Another point to be made is that the paper concentrates on the formulation of the models, where the heuristic applied was not especially developed for these kinds of models. Hence, probably CPU-time can be gained by the development of more specific heuristics especially designed to solve the proposed models.
References


Table 1

Differences between test information functions for weakly parallel tests constructed with model (14)-(24)

<table>
<thead>
<tr>
<th>T</th>
<th>Y_{min}</th>
<th>θ_1</th>
<th>θ_2</th>
<th>θ_3</th>
<th>θ_4</th>
<th>θ_5</th>
<th>θ_6</th>
<th>CPU time (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1.476</td>
<td>0.031</td>
<td>0.983</td>
<td>0.877</td>
<td>0.382</td>
<td>0.083</td>
<td>0.523</td>
<td>2.64</td>
</tr>
<tr>
<td>3</td>
<td>1.384</td>
<td>0.028</td>
<td>0.594</td>
<td>0.914</td>
<td>1.206</td>
<td>0.269</td>
<td>0.311</td>
<td>6.66</td>
</tr>
<tr>
<td>4</td>
<td>1.311</td>
<td>0.059</td>
<td>0.587</td>
<td>0.680</td>
<td>0.432</td>
<td>1.587</td>
<td>0.576</td>
<td>10.68</td>
</tr>
<tr>
<td>5</td>
<td>1.226</td>
<td>0.138</td>
<td>0.767</td>
<td>0.900</td>
<td>0.819</td>
<td>2.563</td>
<td>0.884</td>
<td>14.34</td>
</tr>
<tr>
<td>6</td>
<td>1.161</td>
<td>0.133</td>
<td>1.179</td>
<td>1.443</td>
<td>0.733</td>
<td>3.423</td>
<td>1.269</td>
<td>17.82</td>
</tr>
</tbody>
</table>

Note. H_{1} = 0.999; H_{2} = 0.975.
Table 2

Differences between test information functions for weakly parallel tests constructed with model (14)-(24) for adjusted $r_k$ values

<table>
<thead>
<tr>
<th>T</th>
<th>$Y_{\text{min}}$</th>
<th>$\theta_1$</th>
<th>$\theta_2$</th>
<th>$\theta_3$</th>
<th>$\theta_4$</th>
<th>$\theta_5$</th>
<th>$\theta_6$</th>
<th>CPU time (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1.481</td>
<td>0.008</td>
<td>0.376</td>
<td>0.240</td>
<td>0.584</td>
<td>0.522</td>
<td>0.306</td>
<td>4.56</td>
</tr>
<tr>
<td>3</td>
<td>1.391</td>
<td>0.024</td>
<td>0.440</td>
<td>0.288</td>
<td>0.464</td>
<td>0.172</td>
<td>0.188</td>
<td>8.23</td>
</tr>
<tr>
<td>4</td>
<td>1.299</td>
<td>0.152</td>
<td>0.501</td>
<td>0.305</td>
<td>0.673</td>
<td>0.144</td>
<td>0.516</td>
<td>12.19</td>
</tr>
<tr>
<td>5</td>
<td>1.207</td>
<td>0.197</td>
<td>0.562</td>
<td>0.849</td>
<td>1.280</td>
<td>1.990</td>
<td>1.079</td>
<td>23.25</td>
</tr>
<tr>
<td>6</td>
<td>1.158</td>
<td>0.105</td>
<td>0.503</td>
<td>0.645</td>
<td>0.901</td>
<td>0.670</td>
<td>1.335</td>
<td>33.79</td>
</tr>
</tbody>
</table>

Note. $H_1 = 0.999$; $H_2 = 0.975$. 
### Adjusted \( r_k \) values

<table>
<thead>
<tr>
<th>( T )</th>
<th>( r_1' )</th>
<th>( r_2' )</th>
<th>( r_3' )</th>
<th>( r_4' )</th>
<th>( r_5' )</th>
<th>( r_6' )</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2.000</td>
<td>5.818</td>
<td>12.100</td>
<td>21.300</td>
<td>10.800</td>
<td>3.100</td>
</tr>
<tr>
<td>3</td>
<td>2.000</td>
<td>5.571</td>
<td>12.100</td>
<td>21.300</td>
<td>10.800</td>
<td>3.100</td>
</tr>
<tr>
<td>4</td>
<td>2.000</td>
<td>5.514</td>
<td>12.100</td>
<td>21.300</td>
<td>11.119</td>
<td>3.100</td>
</tr>
<tr>
<td>5</td>
<td>2.000</td>
<td>5.652</td>
<td>12.433</td>
<td>21.300</td>
<td>11.584</td>
<td>3.100</td>
</tr>
<tr>
<td>6</td>
<td>2.000</td>
<td>5.943</td>
<td>13.214</td>
<td>21.300</td>
<td>11.884</td>
<td>3.100</td>
</tr>
</tbody>
</table>

**Note.** The non-adjusted values are: \( r_1 = 2 \), \( r_2 = 5.4 \), \( r_3 = 12.1 \), \( r_4 = 21.3 \), \( r_5 = 10.8 \), and \( r_6 = 3.1 \).
### Table 4

**Differences between test information functions for weakly parallel tests constructed with model (14), (16)-(26) for adjusted \( r_k \) values**

<table>
<thead>
<tr>
<th>( t )</th>
<th>( y_{min} )</th>
<th>( \theta_1 )</th>
<th>( \theta_2 )</th>
<th>( \theta_3 )</th>
<th>( \theta_4 )</th>
<th>( \theta_5 )</th>
<th>( \theta_6 )</th>
<th>CPU time (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1.486</td>
<td>0.051</td>
<td>0.784</td>
<td>0.063</td>
<td>0.158</td>
<td>0.139</td>
<td>0.645</td>
<td>5.80</td>
</tr>
<tr>
<td>3</td>
<td>1.401</td>
<td>0.024</td>
<td>0.394</td>
<td>0.196</td>
<td>0.323</td>
<td>0.208</td>
<td>0.111</td>
<td>7.54</td>
</tr>
<tr>
<td>4</td>
<td>1.292</td>
<td>0.130</td>
<td>0.430</td>
<td>0.518</td>
<td>0.743</td>
<td>0.749</td>
<td>0.542</td>
<td>18.22</td>
</tr>
<tr>
<td>5</td>
<td>1.229</td>
<td>0.044</td>
<td>0.207</td>
<td>0.574</td>
<td>0.535</td>
<td>0.496</td>
<td>0.116</td>
<td>26.10</td>
</tr>
<tr>
<td>6</td>
<td>1.155</td>
<td>0.209</td>
<td>0.558</td>
<td>0.208</td>
<td>0.579</td>
<td>0.530</td>
<td>0.194</td>
<td>53.10</td>
</tr>
</tbody>
</table>

**Note.** \( H_1 = 0.999; \ H_2 = 0.975 \). The value of \( H_1 \) had to be adjusted to 0.975 for \( t = 5 \) and 6 during the construction of the last test.
Table 5

Upper bounds for $y_{\text{min}}$

<table>
<thead>
<tr>
<th>$T$</th>
<th>$y''$</th>
<th>$T_{\text{able1}}$</th>
<th>$T_{\text{able2}}$</th>
<th>$T_{\text{able4}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1.513</td>
<td>2.44%</td>
<td>2.12%</td>
<td>1.78%</td>
</tr>
<tr>
<td>3</td>
<td>1.417</td>
<td>2.33%</td>
<td>1.83%</td>
<td>1.13%</td>
</tr>
<tr>
<td>4</td>
<td>1.331</td>
<td>1.50%</td>
<td>2.40%</td>
<td>2.93%</td>
</tr>
<tr>
<td>5</td>
<td>1.256</td>
<td>2.39%</td>
<td>3.90%</td>
<td>2.15%</td>
</tr>
<tr>
<td>6</td>
<td>1.189</td>
<td>2.35%</td>
<td>2.61%</td>
<td>2.86%</td>
</tr>
</tbody>
</table>
Acknowledgements

The author would like to thank ACT and Terry Ackerman for making the item bank available to him. The author also appreciates the help of Jos Rikers and Lorette Bosch in the preparation of the figures.
Figure Captions

**Figure 1.** Information functions of tests constructed with model (14)-(24) for $T = 6$.

**Figure 2.** Information functions of tests constructed with model (14)-(24) where the $r_k$ values were modified for $T = 6$.

**Figure 3.** Information functions of tests constructed with model (14),(16)-(26) where the $r_k$ values were modified for $T = 6$.

**Figure 4.** Information functions of tests constructed by Ackerman. (Note. From "An alternative methodology for creating parallel test forms using the IRT information function" by T.A. Ackerman, 1989. Paper presented at NCME annual meeting. San Francisco. Reprintend by permission.)
Titles of recent Research Reports from the Division of Educational Measurement and Data Analysis, University of Twente, Enschede, The Netherlands.

RR-90-6  J.J. Adema, The Construction of Weakly Parallel Tests by Mathematical Programming
RR-90-5  J.J. Adema, A Revised Simplex Method for Test Construction Problems
RR-90-4  J.J. Adema, Methods and Models for the Construction of Weakly Parallel Tests
RR-90-3  H.J. Vos, Simultaneous Optimization of Classification Decisions Followed by an End-of-Treatment Test
RR-90-2  H. Tobi, Item Response Theory at subject- and group-level
RR-90-1  P. Westers & H. Kelderman, Differential item functioning in multiple choice items
RR-89-6  J.J. Adema, Implementations of the Branch-and-Bound method for test construction problems
RR-89-5  H.J. Vos, A simultaneous approach to optimizing treatment assignments with mastery scores
RR-89-4  M.P.F. Berger, On the efficiency of IRT models when applied to different sampling designs
RR-89-3  D.L. Knol, Stepwise item selection procedures for Rasch scales using quasi-loglinear models
RR-89-2  E. Boekkooi-Timminga, The construction of parallel tests from IRT-based item banks
RR-89-1  R.J.H. Engelen & R.J. Jannarone, A connection between item/subtest regression and the Rasch model
RR-88-18  H.J. Vos, Applications of decision theory to computer based adaptive instructional systems
RR-88-17  H. Kelderman, Loglinear multidimensional IRT models for polytomously scored items
RR-88-16  H. Kelderman, An IRT model for item responses that are subject to omission and/or intrusion errors
RR-88-15  H.J. Vos, Simultaneous optimization of decisions using a linear utility function
RR-88-14  J.J. Adema, The construction of two-stage tests
RR-88-13  J. Kogut, Asymptotic distribution of an IRT person fit index
RR-88-12  E. van der Burg & G. Dijksterhuis, Nonlinear canonical correlation analysis of multiway data
RR-88-11  D.L. Knol & M.P.F. Berger, Empirical comparison between factor analysis and item response models
RR-88-10  H. Kelderman & G. Macready, Loglinear-latent-class models for detecting item bias
RR-88-9   W.J. van der Linden & T.J.H.M. Eggen, The Rasch model as a model for paired comparisons with an individual tie parameter
RR-88-8   R.J.H. Engelen, W.J. van der Linden, & S.J. Oosterloo, Item information in the Rasch model
RR-88-7   J.H.A.N. Rikers, Towards an authoring system for item construction
RR-88-6   H.J. Vos, The use of decision theory in the Minnesota Adaptive Instructional System
RR-88-5   W.J. van der Linden, Optimizing incomplete sample designs for item response model parameters
RR-88-4   J.J. Adema, A note on solving large-scale zero-one programming problems
RR-88-3   E. Boekkooi-Timminga, A cluster-based method for test construction
RR-88-2   W.J. van der Linden & J.J. Adema, Algorithmic test design using classical item parameters
RR-88-1   E. van der Burg & J. de Leeuw, Nonlinear redundancy analysis

Research Reports can be obtained at costs from Bibliotheek, Department of Education, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands.