DOCUMENT RESUME

ED 329 583 TM 016 217

AUTHOR	Boekkooi-Timminga, Ellen
TITLE	A Method for Designing IRT-Based Item Banks. Research Report 90-7.
INSTITUTION	Twente Univ., Enschede (Netherlands). Dept. of Education.
PUB DATE	Dec 90
NOTE	39p.
AVAILABLE FROM	Bibliotheek, Department of Education, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands.
PUB TYPE	Reports - Evaluative/Feasibility (142)
EDRS PRICE	MF01/PC02 Plus Postage.
DESCRIPTORS	Equations (Mathematics); Foreign Countries; *Item Banks; Item Response Theory; *Linear Programing; *Mathematical Models; Psychometrics; *Test Construction; Test Format
IDENTIFIERS	Rasch Model

ABSTRACT

Since 1985 several procedures for computerized test construction using linear programing techniques have been described in the literature. To apply these procedures successfully, suitable item banks are needed. The problem of designing item banks based on item response theory (IRT) is addressed. A procedure is presented that determines whether an existing item bank meets the test construction requirements. If not, the method indicates which items have to be added to the banks so that it will meet the requirements. The comparison of desired and present item bank characteristics, writing, and calibrating items continues until the characteristics of the item bank are acceptable. Four categories of characteristics are: (1) general characteristics (such as format); (2) subject matter characteristics (such as learning objective); (3) psychometric characteristics (such as IRT-parameters); and (4) user statistics. One figure illustrates the procedure. A 19-item list of references is included. (Author/SLD)

A Method for Designing IRT-based Item Banks

Research Report

90-7

Ellen Boekkooi-Timminga

ment of
EDUCATION

Division of Educational Measurement
and Data Analysis

University of Twente

Project Psychometric Aspects of Item Banking No. 51

Colofon:
Typing: L.A.M. Bosch-Padberg
Cover design: Audiovisuele Sectie TOLAB
Toegepaste Onderwijskunde
Printed by: Centrale Reproductie-afdeling
Oplage: 125

A Method for Designing IRT-based Item Banks

Ellen Boekkooi-Timminga

A method for designing IRT-based item banks , Ellen Boekkooi-Timminga - Enschede : University of Twente, Department of Education, December, 1990. - 31 pages

Abstract

Since 1985 several procedures for computerized test construction using linear programming techniques have been described in the literature. To apply these procedures successfully in practice, suitable item banks are needed.

In this paper the problem of designing IRT-based item banks is addressed. A procedure is presented that, for an existing item bank, determines whether it meets the tests construction requirements. If not, the method indicates which items have to be added to the bank, so that it will meet the requirements.

Key words: Item response theory, Rasch model linear programming, item banking, test construction, item bank design, cluster-based test construction, information functions.

A Method for Designing IRT-based Item Banks

An item bank is a large set of items stored with their item characteristics. The use of item banks for test construction is becoming more and more popular. Especially the application of (integer) linear programming methods to test construction problems turns out to be promising (Adema, 1990a, 1990b, 1990c; Adema, Boekkooi-Timminga, & van der Linden, in press; Adema & van der Linden, 1989; Baker, Cohen & Barmish, 1988; Boekkooi-Timminga, 1987, 1989, 1990a, 1990b; de Gruijter, in press; Theunissen, 1985, 1986; van der Linden & Boekkooi-Timminga, 1989). In contrast to the large amount of attention paid to procedures for test construction, the problem of designing item banks was rather ignored. Present guide-lines for designing item banks regard the quality of the individual items, and, for the total collection of items, for instance, the spread of items over objectives, skills, and tasks. No guide-lines are provided for designing suitable IRT-based item banks to which the modern test constiuction methods can be applied successfully. The importance of well-designed item banks is clear. For instance, an item bank may become exhausted after some time, meaning that no longer satisfactory tests can be constructed from it, because a large number of items is excluded from selection on basis of their previous usage. Also, the psychometric quality of tests constructed later may decrease.

Here the problem of assessing the practical usefulness of item banks is addressed. A method is proposed that compares the desired features of the item bank with the present features. To determine the desired features of an item bank, it is assumed that tests will be selected from the item bank using (integer) linear programming techniques. Furthermore, it is assumed that the Rasch model holds. In the discussion of this paper, it is argued why other IRT-models are not considered.

In this paper, first, an outline of the item bank design method is given, and the notation used is summarized. Then, the Basic Method is described. Next, it is described how the Basic Method can be adapted if additional practical constraints have to be considered for the tests. The paper ends with a discussion.

Outline of the Item Bank Design Method

It is assumed that the item banks to be examined fit the Rasch model. The question is, whether an existing item bank will be capable to handle a series of test construction requests satisfactory.

In short the method works as follows: First, the specifications of all tests to be selected from the item bank have to be given. Next, the numbers and characteristics of the items needed in the bank, to be able to construct these tests, are determined and compared with the characteristics of the present item bank. If the item bank does not fit the needs, the method determines the characteristics of items that have to be added to the item bank, such that it will become suitable. After these items have been constructed and calibrated, the present and the desired item bank are compared again. This process of comparing the desired and present item bank, writing and calibrating items continues until the characteristics of the item bank are acceptable.

Four categories of item characteristics, that can be stored in an item bank, are distinguished (van Thiel & Zwarts, 1986): (1) general characteristics, e.g. item format, (2) subject matter characteristics, e.g. learning objective, (3) psychometric characteristics, e.g. IRT-parameters, and (4) user statistics, e.g. date of last use. Stodola (1974) gives a thorough overview of the individual item characteristics of interest for item banking. It is not necessary to consider constraints on all of these item characteristics in the test specifications to be given in the first step of the item bank design method. One reason is that a practitioner often can not formulate such detailed requirements, when the actual construction of the tests is not under discussion. The second reason is that, because uni-dimensional item banks are considered, items differ only with respect to a small number of characteristics. The most important differences are psychometric. Therefore the main emphasis in this paper will be on the psychometric characteristics of the items. The information function is used to represent these psychometric characteristics. A definition of the information function can be found, for instance, in Birnbaum (1968) or Lord (1980).

Notation

The symbols used are given in alphabetical order.

Basic Method

b_c difficulty parameter of items located within ability interval c.

$c = 1, \ldots, C$ ability intervals.

$g = 1, \ldots, G$ groups of tests not allowed to have any items in common.

H_t set of ability intervals with $s_{tc} > 0$ for test t.

$I_c(\theta_k)$ information of an item located within ability interval c at ability level θ_k.

$j = 1, \ldots, J$ item characteristics.

$k = 1, \ldots, K$ ability points.

K_t set of ability intervals with $v_{tc} < 0$ for test t.

l_{tj} the shortage of items reflecting item characteristic j in the item bank for test t.

m_c, m_{cj} help variable used for computing the number of items (reflecting item characteristic j) to be added to ability interval c of the item bank.

N test length.

N_j number of items in test reflecting item characteristic j.

n_c, n_{cj} final number of items (reflecting item characteristic j) to be added to ability interval c of the item bank.

r_k relative height of the test information function at ability level k.

s_{gc}, s_{gcj} number of items (re*lecting item characteristic j) needed in interval c for Group g of non-overlapping tests.

s_{tc} number of items needed in ability interval c for test t.

s_{tj} number of items reflecting item characteristic j needed for test t.

$t = 1, \ldots, T$ tests to be constructed.

t_{tcj} number of items reflecting item characteristic j needed in interval c for test t.

V_g set of tests t belonging to Group g.

$|v_{gc}|$, $|v_{gcj}|$ number of items (reflecting item characteristic j) to be added to ability interval c so that the tests in Group g can be constructed best (if v_{gc}, $v_{gcj} < 0$).

$|v_{tc}|$ number of items to be added to ability interval c of the item bank so that test t can be constructed best (if $v_{tc} < 0$).

w_{tj} number of items reflecting item characteristic j available in the item bank in the ability intervals in H_t.

x_c, x_{cj} number of items (reflecting item characteristic j) in ability interval c needed for the test.

y additional decision variable used to maximize the relative test information function.

z_c, z_{cj} number of items (reflecting item characteristic j) available in ability interval c of the item bank.

Basic Method

It is assumed that the ability continuum is partitioned into C (c = 1, 2, ..., C) intervals. $I_c(\theta_k)$ represents the item information function value at ability level θ_k of all items, having difficulty parameter values within ability interval c. Taking the midpoint of each ability interval c as the item difficulty value b_c, $I_c(\theta_k)$ is computed as follows:

(1) $I_c(\theta_k) = [\exp(\theta_k - b_c)(1 + \exp(-\theta_k + b_c)]^{-1}$.

The five steps of the Basic Method are outlined below.

Step 1. *Determine the characteristics of the desired item bank.*

Make an overview of the desired types and numbers of tests to be constructed from the item bank. Groups ($g = 1, 2, ..., G$) of tests that are not allowed to have any overlapping items, are identified. Tests may be part of several groups. If tests are allowed to have some items in common, the practitioner should decide whether the tests have to be treated as tests that may have all or no items in common.

For each test the number of items, x_c, that should be available in each ability interval c of the item bank, is determined, so that a test fitting the specifications best can be selected from the bank. Below it is described how this is done. Next, for each Group g the number of items s_{gc} needed in interval c for this group, are computed by summing the x_c's of all tests in this group.

The tests are "constructed" according to the test specifications as follows. It is assumed that the item bank contains an infinite number of items in each ability interval c. It is free to choose any linear programming model for the "construction" of a test, as long as the following decision variables are used: x_c, indicating the number of items from interval c to be included in the test. Two examples of test construction models are the model of minimum test length (Theunissen, 1985) and the Maximin Model (van der Linden and Boekkooi-Timminga, 1989). As the ability continuum is assumed to be partitioned into C intervals, actually, the cluster based method proposed by Boekkooi-Timminga (1990b) is applied 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 cluster can be considered equivalent. In case of the Maximin Model the test "construction" problem is formulated as follows:

$$(2) \quad \text{maximize} \quad y,$$

subject to

$$(3) \quad \sum_{c=1}^{C} I_c(\theta_k)x_c - r_k y \geq 0, \qquad k = 1, 2, \ldots, K,$$

$$(4) \quad \sum_{c=1}^{C} x_c = N,$$

$$(5) \quad x_c \geq 0, \qquad c = 1, 2, \ldots, C,$$

$$(6) \quad y \geq 0.$$

The decision variable x_c gives the number of items needed in ability interval c of the item bank to be able to construct the test. The r_k's in (3) give the relative heights of the test information function at the ability levels k (= 1, 2, ..., K), specified by the practitioner. By maximizing decision variable y in (2), the lower bounds $r_k y$ to the test information at the K ability levels considered are maximized. Constraint (4) fixes the test length at N. The bounds on the decision variables are set in (5) and (6).

In this model the maximum number of items to be selected from each ability interval c is unbounded ($x_c \geq 0$ in Expression [5]), thus the item bank is infinite. If it occurs that there are intervals for which no items can be written, the corresponding x_c's should be set to zero in (5). Note that the decision variables x_c are not restricted to take integer values. Thus, standard linear programming problems that can be solved quickly are the result.

Model (2) - (6) is a basic model that may be extended by constraints that can be formulated as linear expressions of the decision variables x_c, where x_c is defined as above. For instance, exact 10 items have to be included in the test from the ability intervals 1 to 5.

Step 2. *Determine the characteristics of the present item bank.*
Compute the number of items, z_c, available in each ability interval c of the present item bank.

Step 3. *Compute the differences between the desired and present item bank.*
In order to be able to select the tests in Group g from the item bank, it is necessary that the number of items z_c available in each ability interval of the present item bank

is larger than the required numbers s_{gc}. Now compute for each Group g the difference, v_{gc}, between z_c and s_{gc} at each ability interval through:

$$(7) \quad v_{gc} = z_c - s_{gc}, \qquad c = 1, 2, \ldots, C, \quad g = 1, 2, \ldots, G.$$

If $v_{gc} \geq 0$ for all groups and all ability intervals, the present item bank is perfectly suited for the test construction needs stated, and the procedure is STOPPED. A negative v_{gc} value gives the numbers of items to be added to ability interval c of the item bank, such that it will become possible to select the tests in Group g best.

Step 4. *Determine the numbers of items, n_c, to be added to each ability interval of the item bank.*
In this step, first, the lowest values, m_c, of the v_{gc}'s over the Groups g are obtained as follows:

$$(8) \quad m_c = \underset{g = 1, \ldots, G}{\text{minimum}} \{v_{gc}\}, \qquad \text{for } c = 1, 2, \ldots, C.$$

If m_c is negative, $|m_c|$ items should be added to this ability interval of the item bank. If it is positive, m_c gives the number of items left in this interval.

The desire to select the most perfect tests according to the test specifications, might not be too strong in practice. If this is the case, it is possible to compensate for the shortage of items in an interval by items from adjacent intervals. This option is of interest, considering the fact that it is difficult for item writers, to construct exactly the items required. Another consideration is that, if narrow intervals are used and no compensation is allowed, it will turn out that, compared to a situation with wider intervals, a larger number of items will have to be added to the bank. If no compensation should be considered $n_c = |\min(0, m_c)|$ for all c, and Goto Step 5.

Items in intervals with $m_c > 0$ can be used to compensate a shortage of items in an adjacent interval, if the amount of information in these intervals, at the midpoint of the interval with the shortage, is at least equal to the amount of information the n_c items that had to be added would provide. It is obvious that the number of items needed for compensation from adjacent intervals, is larger than n_c. Also, the wider the intervals the more items are needed for compensation, and, the less perfect the tests will be that are constructed from the item bank at a later stage. This is caused by the fact that the midpoints of the intervals, are used as the item difficulty values for computing the information values of the items within a cluster. Note that the m_c items left in an interval can only be used once for compensation purposes.

Step 5. *Write items to be added to the item bank.*
Item writers should try to construct the n_c items as determined by Step 4 for each ability interval c. Next, the items are calibrated. Because in practice, it will never be possible to construct exactly the items that were desired, Goto Step 2 and compare the new item bank with the desired one.

Example
The suitability of a bank with percentage measurement items, developed by the National Institute for Educational Measurement in the Netherlands (Cito), was checked. The item bank consisted of 470 items, after a Rasch calibration 416 items remained. Ability intervals were formed by taking widths of 0.6 on the ability continuum from -4.5 to 4.5 logits; 17 intervals were the result.

The Basic Method was applied. Three groups of non-overlapping tests were taken into account. Group 1 consisted of four tests, all having the same test specifications: Minimal test length, and target test information values at θ = -1, 0, 1 , 2 of 4, 8, 8, 4, respectively. So the model proposed by Theunissen (1985) was used. Three selective tests were included in Group 2, each test consisted of 30 items, and test information was maximized for Test 1 at θ = -1, for Test 2 at θ = 0, and for Test 3 at θ = 1. The five tests in Group 3 consisted also of 30 items, they were constructed using the Maximin Model, considering the ability points θ = -

2, -1, 0, 1, 2 for which the relative target information values were equal.

The mathematical programming package PcProg (Quantitative Management Software, 1986) was used, on a MsDos personal computer under 8 MHz with mathematical co-processor and hard disk, to "construct" the tests in Step 1. The optimization times needed for "constructing" the tests ranged from 0.2 to 0.6 seconds, the computation times for the matrix generation ranged from 4.70 to 9.30 seconds. In Figure 1 the results of Step 1, 2, 3, and 4 are shown. The first three lines indicate the numbers of items needed in each of the ability intervals to be able to construct the tests in each group as optimal as possible (Step 1). The fourth line gives the numbers of items available in each interval of the present item bank (Step 2). Lines 5 - 7 show the differences between Line 4 and each of the lines 1 - 3 (Step 3). The total difference between the present item bank and the desired one is shown in Line 8 (Step 4). The negative values indicate the numbers of items to be added to the item bank to become suited.

Insert Figure 1 about here

From Figure 1 it is obvious that the shortage of items could not be completely compensated from the adjacent ability intervals, as the number of -42 could not be compensated by the number of items beforehand in its adjacent intervals (= 18). After compensation it turned out that still 26 items had to be constructed for this interval. The shortage of items in the other two intervals could be completely compensated. The 52 items could be compensated by considering all 34 items from the lower adjacent interval and 23 items (out of 24) from the upper interval. The 20 items could be compensated by the remaining item from its lower interval, and 22 (out of 23) items from the upper interval.

Additional Practical Constraints

Suppose that beside psychometric constraints, that can be formulated using the decision variables x_c in (2) - (6), other practical constraints (see van der Linden & Boekkooi-Timminga, 1989) need to be considered in the test specifications of Step 1. Then, the Basic Method can be generalized in two manners depending on the nature of the item characteristics considered in these constraints. The generalizations called Procedures 1 and 2 are described next.

Procedure 1

Observing the items on a certain item characteristic in the present item bank, it is expected that items can be written for each individual ability interval, if the present items one contained in almost all ability intervals. Practical constraints on such item characteristics can be treated by adapting the Basic Method as follows.

In Step 1, first, formulate the test specifications as is done in the Basic Method. Additional specifications regard the numbers of items reflecting certain item characteristics that have to be included in each test. It is possible that the items in the bank reflect more than one of these characteristics. If this is the case, each combination of characteristics that may occur, is treated like a new characteristic. Thus, the resulting set of item characteristics ($j = 1, 2, \ldots, J$) is mutually exclusive. Next, the numbers of items, s_{tj}, reflecting item characteristic j that are needed for tests t are given by the practitioner. Then, the x_c's are obtained for each test, as described for the Basic Method. Note that the constraints on the item characteristics are not regarded yet. The number of items, s_{tc}, needed in interval c for test t is set equal to x_c. Let H_t be the collection of ability intervals with $s_{tc} > 0$ for test t. Determine in Step 2 the z_c's as described for the Basic Method, and the numbers of items, z_{cj}, on each item characteristic available in ability interval c of the item bank. Next, compute w_{tj} for each test t, indicating the numbers of items reflecting item characteristic j available in the ability intervals in H_t of the item bank, through:

$$(9) \quad w_{tj} = \sum_{c \in H_t} z_{cj}, \qquad j = 1, 2, \ldots, J, \quad t = 1, 2, \ldots, T.$$

In Step 3 the differences v_{tc} between z_c and s_{tc} are obtained by taking $v_{tc} = z_c - s_{tc}$ at all ability intervals for each test t. Also, the differences between s_{tj} and w_{tj} are determined for each item characteristic j and each test t. These differences, $l_{tj} = |\min\{0, w_{tj}-s_{tj}\}|$ give the shortage of items on characteristic j for test t.

Step 4. If $v_{tc} < 0$ then $|v_{tc}|$ gives the number of items to be added to interval c of the item bank. Again, compensation can be used, if this is done H_t and l_{tj} should be adapted. Given the l_{tj}'s and $|v_{tc}|$'s $(v_{tc} < 0)$, the practitioner has to decide on the number of items, t_{tcj}, on characteristic j that have to be added to interval c for test t. It is obvious that these numbers can be chosen in several ways, there is no unique solution. While choosing these numbers the practitioner should keep an eye on the spread of items, reflecting the characteristic of interest, in the present item bank over the ability continuum. He/she should fix the numbers t_{tcj} such that it is expected that these items can easily be constructed, because several items with approximately the same characteristics are already included in the item bank.

Two possibilities are distinguished for each test t:

(a) $\sum_{c \in K_t} |v_{tc}| < \sum_{j=1}^{J} l_{tj}$, and

(b) $\sum_{c \in K_t} |v_{tc}| \geq \sum_{j=1}^{J} l_{tj}$,

where K_t is the collection of ability intervals for which $v_{tc} < 0$. In case of (b) the t_{tcj}'s are chosen such that the items l_{tj} are spread over the ability intervals such that at most $|v_{tc}|$ items have to be constructed for interval c. For case (a), part of the l_{tj} items have to be constructed such that the required numbers $|v_{tc}|$ are obtained. The remaining items to be added to the bank should be constructed such that the numbers of items, reflecting the item characteristics of interest, added to the bank in each of the ability intervals in H_t is not larger than s_{tc} determined in Step 1.

At this point of Procedure 1, the groups g(= 1, 2, ..., G) of non-overlapping tests identified in Step 1 are regarded. Given the t_{tcj}'s for all tests within Group g (V_g), the numbers of items, s_{gcj}, reflecting characteristic j to be added to ability interval c of the item bank, can be computed as follows:

$$(10)\quad s_{gcj} = \sum_{t \in V_g} t_{tcj}, \qquad \begin{aligned} g &= 1, 2, \ldots, G, \\ c &= 1, 2, \ldots, C, \\ j &= 1, 2, \ldots, J. \end{aligned}$$

After the s_{gcj}'s have been obtained for each Group g, the n_{cj}'s, giving the number of items on characteristic j to be added to ability interval c of the bank, are obtained through:

$$(11)\quad n_{cj} = \underset{g = 1, \ldots, G}{\text{maximum}} \{s_{gcj}\}, \qquad \text{for } c = 1, 2, \ldots, C, \quad j = 1, 2, \ldots, J.$$

In Step 5 item writers should try to construct these n_{cj} items.

Procedure 2

Procedure 2 should be used if item characteristics have to be considered in the practical constraints that represent items that can not be constructed for a number of ability intervals. This is the case, for instance, if items regarding a certain learning objective all have low difficulty values. Procedure 2 should also be applied if typical test specifications have to be taken into account, for instance, specifications regarding relational aspects between item characteristics. It is emphasized that Procedure 1 should be applied as much as possible, because the item bank design process will be simpler. Only item characteristics that definitely will cause problems applying Procedure 1 should be treated by Procedure 2.

The Basic Method is adapted as follows. As for Procedure 1, the item bank designer decides in Step 1 which item characteristics ($j = 1, 2, \ldots, J$) have to be taken into account. Next, the decision variables x_c are replaced by x_{cj}. The x_{cj}'s denote the numbers of items selected for the test from ability interval c that reflect item characteristic j. Model (2) - (6) is generalized as follows:

(12) maximize y,

subject to

(13) $$\sum_{c=1}^{C} \sum_{j=1}^{J} I_c(\theta_k) x_{cj} - r_k y \geq 0, \qquad k = 1, 2, \ldots, K,$$

(14) $$\sum_{c=1}^{C} \sum_{j=1}^{J} x_{cj} = N,$$

(15) $$\sum_{c=1}^{C} x_{cj} = N_j, \qquad j = 1, 2, \ldots, J,$$

(16) $$0 \leq x_{cj} \leq u_{cj}, \qquad c = 1, 2, \ldots, C, \quad j = 1, 2, \ldots, J,$$

(17) $y \geq 0$.

As an example, the constraints in (15) state that exact N_j items reflecting characteristic j have to be included in the test. Note that, if the above model is applied, it is possible that several solutions with the same objective function value can be obtained, especially if u_{cj} is large, so there is no unique solution. Application of the above model is only meaningful if several small u_{cj} values are to be considered, or if many practical constraints like those in (15) are included in the model. Otherwise Procedure 1 should be applied.

Now for each Group g of non-overlapping tests, the x_{cj}'s are summed for each item characteristic j to get the corresponding s_{gcj}, defining for Group g the number of items on characteristic j needed in ability interval c. In Step 2, the z_{cj}'s, indicating the numbers of items on characteristic j included in ability interval c, are determined for the present item bank.

In Step 3 the differences between the s_{gcj}'s and z_{cj}'s are computed for all ability intervals and all item characteristics of interest. This is done for each Group g of non-overlapping tests.

$$(18) \quad v_{gcj} = z_{cj} - s_{gcj}, \qquad \begin{array}{l} g = 1, 2, \ldots, G, \\ c = 1, 2, \ldots, C, \\ j = 1, 2, \ldots, J. \end{array}$$

If all v_{gcj}'s are all positive, the present item bank is perfect.

In Step 4 of the Basic Method m_c is replaced by m_{cj}. The m_{cj}'s are determined as follows:

$$(19)\quad m_{cj} = \underset{g = 1, \ldots, G}{\text{minimum}} \{v_{gcj}\}, \qquad \text{for } c = 1, 2, \ldots, C, \quad j = 1, 2, \ldots, J.$$

Finally, $n_{cj} = |\min\{0, m_{cj}\}|$, where n_{cj} is the number of items reflecting item characteristic j that has to be added to ability interval c. Again compensation from adjacent ability intervals can be applied. Then, each item characteristic is considered separately.

In Step 5 item writers should try to construct the n_{cj} items indicated by Step 4.

Discussion

In this paper a method for item bank design has been proposed. Its purpose is to check whether an existing IRT-based item bank can fulfil a series of test construction desires. If not, it indicates which items should be added to the item bank, such that it becomes satisfactory. The method is of interest when item banks are developed that have to be satisfactory for a certain period without being extended.

Here it was assumed that the Rasch model holds, however, the method can also be applied to the three-parameter logistic model. In this case the items in the bank have to be divided over mutually exclusive groups (clusters), such that the items in the same cluster have fairly equal information functions. Next, the ability intervals c considered in the method are replaced by these clusters. Considering the two- or three-parameter logistic model instead of the Rasch model will be more complicated. First, there will be much more clusters for these models, than ability intervals for the Rasch model. Second, the compensation process will be more complicated, and it will be even more difficult for item writers to construct items belonging to an explicit cluster.

In this paper linear programming models instead of integer linear programming models were formulated, thus, it is expected that some decision variable values will have fractional parts. It might appear contradictory that the decision variables x_c and x_{cj} reflect numbers of items (integers), however, a linear programming approach is justifiable for this application. The reason for preferring the linear approach is obvious because linear programming problems can be solved very quickly compared to integer linear programming problems. This is very important because several linear programming problems have to be solved when the method is applied. Research on test construction by linear programming showed that the tests obtained by rounding fractional decision variable values are fairly optimal in many cases (e.g. Boekkooi-Timminga, 1989, 1990b). This was especially the case for the Rasch model. So only small errors are made when linear instead of integer linear programming is used. The errors that are made by rounding the fractional values, cause that a little more items have to be constructed, because all fractional values are rounded upwards. However, as it will be difficult for test constructors to give the exact test specifications for tests in advance (Step 1), it is acceptable that the appropriateness of the item bank is tested relatively rough. Thus, the errors made are quite acceptable.

It will be difficult in practice to construct the desired items in Step 5. If it turns out that unrealistic large numbers of items have to be added to the bank there are three options. One is to relax the non-overlapping requirements of tests in Step 1, such that less tests are required to have no overlap. Another is to put upper bounds on the numbers of items that may be selected from certain ability intervals in Step 1. A consequence of this approach is that, when the actual tests are selected from the item bank at a later stage, the same upper bounds are required, otherwise the problem of item bank exhaustion might turn up again. The third option is to consider not only items from adjacent ability intervals for compensation in Step 4, but also those from other intervals. However, care should be taken doing this, because the tests that will be constructed from this item bank will become less perfect, and the possibility of an exhausted item bank becomes actual again.

It is also possible that large numbers of items have to be added to the item bank, because the test specifications are unrealistic for the item bank. For instance, if only difficult items can be constructed for the item bank, while only easy items are required for the tests. In this case the desired items can not be written, and, if compensation is allowed, large numbers of difficult items will have to be added to get the amount of information required at the low ability levels. However, it is obvious that this approach is not to be recommended.

The compensation process mentioned in this paper is part of Step 4. However, it is also possible to include it into Step 3. This might be of interest, if it is required to have different ranges of compensation allowed for the respective groups of non-overlapping tests. For instance, for Group 1 no compensation is allowed, while for Group 2 compensation is allowed from both adjacent ability intervals.

Acknowledgements

The Netherlands Organization for Scientific Research (NWO) is gratefully acknowledged for funding this project. This research was conducted while Ellen Boekkooi-Timminga was supported by a PSYCHON-grant from NWO (560-267-030) awarded to W.J. van der Linden. I would like to thank Joop Bokhoven and Jan Janssen for allowing me to use their item bank on percentage measurement.

References

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

Adema, J.J. (1990b). *The construction of weakly parallel tests by mathematical programming*. Submitted for publication.

Adema, J.J. (1990c). *Methods and models for the construction of weakly parallel tests*. Submitted for publication.

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*.

Adema, J.J., & van der Linden, W.J. (1989). Algorithms for computerized test construction using classical item parameters. *Journal of Educational Statistics*, *14*, 279-290.

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 and their use in inferring an examinee's ability. In F.M. Lord, and M.R. Novick, *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: Academisch Boeken Centrum.

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

Boekkooi-Timminga, E. (1990b). A cluster-based method for test construction. *Applied Psychological Measurement*, *14*.

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

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

Quantitative Management Software (1986). *PcProg*: A package for solving mathematical programming problems on MsDos machines [Computer program]. Amsterdam, The Netherlands.

Stodola, Q.C. (1974). Item classification and selection. In G. Lippey (Ed.), *Computer-assisted test construction*. Englewood Cliffs, NJ: Educational Technology Publications.

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

Theunissen, T.J.J.M. (1986). Optimization algorithms in test design. *Applied Psychological Measurement*, *10*, 381-390.

van der Linden, W.J., & Boekkooi-Timminga, E. (1989). A maximin model for test design with practical constraints. *Psychometrika*, *54*, 237-247.

van Thiel, C.C., & Zwarts, M.A. (1986). Development of a testing service system. Applied Psychological Measurement, 10, 391-403.

Figure Captions

<u>Figure 1</u>: Testing the Suitability of a Percentage Measurement Item Bank.

								28	116								s_{1c}
-4.5																4.5	
						30		30		30							s_{2c}
					75					30	50						s_{3c}
3	4	4	8	18	33	30	72	64	64	54	30	23	2	5	0	2	z_c: Present Item Bank
3	4	4	8	18	33	30	72	36	-52	54	30	23	2	5	0	2	v_{1c}: $z_c - s_c$
3	4	4	8	18	33	0	72	34	64	24	30	23	2	5	0	2	v_{2c}: $z_c - s_c$
3	4	4	8	18	-42	30	72	64	64	24	-20	23	2	5	0	2	v_{3c}: $z_c - s_c$
3	4	4	8	18	-42	0	72	34	-52	24	-20	23	2	5	0	2	m_c: Total Difference

Titles of recent Research Reports from the Division of Educational Measurement and Data Analysis. University of Twente, Enschede, The Netherlands.

RR-90-7 E. Boekkooi-Timminga, *A Method for Designing IRT-based Item Banks*

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.

rtment of

EDUCATION

A publication by
the Department of Education
of the University of Twente
ede
nds