DOCUMENT RESUME

ED 473 527 TM 034 735

AUTHOR	Veldkamp, Bernard P.
TITLE	Optimal Test Construction. Research Report.
INSTITUTION	Twente Univ., Enschede (Netherlands). Faculty of Educational Science and Technology.
REPORT NO	RR-02-08
PUB DATE	2002-00-00
NOTE	37p.
AVAILABLE FROM	Faculty of Educational Science and Technology, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands. E-mail: Fox@edte.utwente.nl.
PUB TYPE	Information Analyses (070)
EDRS PRICE	EDRS Price MF01/PC02 Plus Postage.
DESCRIPTORS	Algorithms; *Item Banks; *Selection; *Test Construction; Test Items

ABSTRACT

This paper discusses optimal test construction, which deals with the selection of items from a pool to construct a test that performs optimally with respect to the objective of the test and simultaneously meets all test specifications. Optimal test construction problems can be formulated as mathematical decision models. Algorithms and heuristics have been developed to solve the models that can be used to construct tests. (Contains 2 tables, 2 figures, and 27 references.) (Author/SLD)

Reproductions supplied by EDRS are the best that can be made
from the original document.

TM TMTR 0075

ED 473 527

Optimal Test Construction

Research Report 02-08

PERMISSION TO REPRODUCE AND DISSEMINATE THIS MATERIAL HAS BEEN GRANTED BY

J. Nelissen

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

1

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☑ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

Bernard P. Veldkamp

TM034735

faculty of
EDUCATIONAL SCIENCE AND TECHNOLOGY

University of Twente

Department of
Educational Measurement and Data Analysis

BEST COPY AVAILABLE

Optimal Test Construction

Bernard P. Veldkamp

Requests for information should be sent to: Bernard P. Veldkamp, Department of Educational Measurements and Data Analysis, University of Twente, P.O. Box 217, 7500 AE Enschede, THE NETHERLANDS. Email: Veldkamp@edte.utwente.nl

Optimal Test Construction[1]

Abstract

Optimal test construction deals with the selection of items from a pool to construct a test that performs optimal with respect to the objective of the test and simultaneously meets all test specifications. Optimal test construction problems can be formulated as mathematical decision models. Algorithms and heuristics have been developed to solve the models that can be used to construct tests.

Keywords: 0-1 Linear programming, Automated test assembly, Heuristics, Mathematical programming, Test construction, Weighted deviations model.

[1] The paper is a extended version of Veldkamp, B.P. (2002). Optimal test construction. In: *Encyclopedia of Social Measurement*. Submitted.

Introduction

In the area of educational and psychological measurement, tests are often used as data collection instruments. The data are used to assess the score of a testee on an ability. To define the relation between the answers to a test and the ability, a measurement model is formulated. Based on the model and the data, the score of the candidate is estimated.

In the early days of testing, oral tests and interviews were used to assess the abilities. The measurement model was not formulated explicitly, and the score did not depend only on the answers but also on the mood of the rater. In some areas, standardized test forms, like the Binet-Simon test, were introduced to make scores more comparable. These standardized test forms solved one problem, but introduced another. When items in the test become known, candidates might try to influence their scores by formulating answers in advance and learning them by hart. Especially in educational measurement this is a potential problem. Because of this, teachers, and test committees had to formulate new items for every new test administration.

The new items had to be written, pre-tested, calibrated, edited and transported to the test location, before they could actually be used to collect data. This process was quite expensive, and because these items were used only once, it meant quite a waste of efforts and time. To increase efficiency, new items were collected in item pools. In educational measurement, these pools usually contain between a few hundred and a few thousands of items. From these pools, tests can be selected for several purposes. Item selection is based on the test specifications. For a large test construction problem the number of specifications might easily run into a few hundred. When the number of items to choose from is also large, manual test construction will become far from optimal and a computer algorithm has to be used to construct tests optimally.

Item Response Theory.

The introduction of Item Response Theory (IRT) to large scale testing provided new opportunities for test assembly. In IRT measurement models, item parameters and person parameters are modeled separately (Hambleton & Swaminathan, 1985). Apart from sampling variation, the item parameters do not depend on the population or on the other items in the test. For dichotomously scored items the Rasch Model, the 2-

Parameter Logistic Model (2PLM), and the 3-Parameter Logistic Model (3PLM) are most often applied. The relation between the item and the person parameters can be formulated by the following logistic expression:

$$P_i(\theta_j) := c_i + (1-c_i)\frac{e^{a_i(\theta_j - b_i)}}{1+e^{a_i(\theta_j - b_i)}}, \quad (1)$$

where $P_i(\theta_j)$ is defined as the probability of obtaining a correct answer to the item i for person j. The person parameter θ_j denotes the latent ability and the item parameters a_i, b_i, and c_i denote the discrimination, the difficulty and the guessing parameter. For the Rasch model and the 2PLM, the guessing parameter is supposed to be zero, and for the Rasch model all discrimination parameter are supposed to be equal to one. For polytomously scored items polytomous IRT models have also been formulated, for example the Graded Response Model, the Graded Partial Credit Model or the Nominal Response Model.

When IRT models are applied, measurement precision is determined by the amount of information in the test. Test information (Lord, 1980), which is a function of the person parameter and the parameters of the items in the test, is defined by:

$$I(\theta) = \sum_{i \in \text{test}} I_i(\theta) = \sum_{i \in \text{test}} \frac{\left[P_i'(\theta)\right]^2}{P_i(\theta)Q_i(\theta)}, \quad (2)$$

where $P_i'(\theta)$ is the first derivative of $P_i(\theta)$, and $Q_i(\theta)$ is the probability of obtaining a wrong answer.

The focus in optimal test construction is therefore to find a computer algorithm that selects items from the pool that maximize the amount of information in the test but also meet the test specifications.

Knapsack problem.

To solve optimal test construction problem, all kinds of smart decision rules have been developed. Birnbaum (1968) presented a rather general approach. His algorithm consisted of the following steps.

1. Decide on the shape of the desired test information function.
2. Select items from the pool with information functions to fill areas under the target information function
3. After each item is added to the test, calculate the test information function
4. Continue selecting items until the test information function approximates the desired shape.

However, if more and more test specifications have to be added to the construction problem, the approach becomes hard to adapt. Theunissen (1985) made the observation that optimal test construction is just one example of a selection problem. Other well-known examples are flight-scheduling, work-scheduling, human resource planning, inventory management, and the traveler-salesman problem.

In the area of Operations Research or Mathematical Programming, algorithms are developed to solve such problems (Papadimitriou & Steiglitz, 1982). To find the best algorithm for optimal test construction, algorithms from this area have been adapted and applied. One class of selection problems are the so-called knapsack-problems. Before a traveler leaves, he has to fill his knapsack. All possible items he may wish to pack represent a certain value to him, but the volume of the knapsack is limited. The problem is how to maximize the value of all the items in the knapsack, while the volume restriction is met. More formally stated:

$$\max \sum_{i=1}^{n} c_i x_i \tag{3}$$

subject to:

$$\sum_{i=1}^{n} a_i x_i \leq b \tag{4}$$

$$x_i \in \{0,1\} \tag{5}$$

where c_i denotes the value of item i, a_i denotes the volume of item i, and the volume of the knapsack is denoted by b. The decision variables x_i denote whether an item is selected (x_i =1) or not (x_i =0). In Mathematical Programming terms, the formula in Equation 3 is called the objective function, and Equations 4 and 5 are called the constraints.

In optimal test construction, a problem can be described as a knapsack problem. The value of a test, that is the information in the test, has to be maximized, in order to obtain optimal measurement precision. The volume can be interpreted as all possible tests that meet the constraints defined by the test attributes, and an item has either to be selected ($x_i = 1$) or not ($x_i = 0$) for the test.

Overview

In the remainder of this chapter it is demonstrated how to model several kinds of optimal test construction problems using mathematical programming, and some algorithms for solving the problems are described. The following topics were selected. In Section II, the problem of constructing one linear test form is described. Several objective functions and different kinds of constraints are suggested. This section results in a general formulation of a test construction problem. Section III addresses models for several major test construction problems. The algorithms and heuristics to solve the problems are described in Section IV. In Section V, a numerical example of a test construction problem is given. Finally, Section VI discusses the topic and gives some recommendations about the use.

Constructing a Single Linear Test Form

The traditional format in both educational and psychological testing is the linear test form. A linear test form is a paper-and-pencil (P&P) test that can be used for a population of candidates. To select a linear test form from an item pool, first the objective of the test has to be specified. Then, the test specifications have to be written as a set of constraints. For a review of the literature on test construction that uses these steps, see van der Linden (1998).

Objective functions.

How to specify the objective function in a test construction model depends on the goals or objectives of the test. Three examples of objective functions are given.

A simple objective deals with security of the item pool and the costs of testing. When more items are exposed to candidates, the item pool becomes known faster and the costs of maintaining the pool will be higher. So when the objective of the test is to

maximize security of the pool, or to minimize costs of testing, a reasonable objective function is to minimize the number of items in a test. In this case, the objective function can be formulated as:

$$\min \sum_{i=1}^{n} x_i \,. \tag{6}$$

The objective of the test can also be chosen to depend on the decisions that have to be taken based on it's scores. In criterion-referenced testing, a cut-off score θ^* is specified in advance. When the estimated ability $\hat{\theta}$ is larger than or equal to θ^*, the candidate passes, otherwise the candidate fails. For candidates who clearly pass or clearly fail, measurement precision need not be optimal. However, for candidates who are close to the cut-off score, measurement precision should be high. To construct a test that will serve this purpose, the following objective function can be used:

$$\max \sum_{i=1}^{n} I_i(\theta^*) x_i \,. \tag{7}$$

where $I_i(\theta^*)$ denotes the information item i provides for the cut-off score.

A third example is a broad ability test. The test should measure the abilities of a population of candidates, for example in diagnostic testing in school classes. Before the test is constructed, targets for the information in the test are defined. The objective of the test construction problem is to minimize the distance between the target information function and the test information function. A few points on the θ-scale are chosen, and for these points the distance between target and test function is minimized. This problem leads to the following objective function:

$$\min y \tag{8}$$

subject to:

$$\left| \sum_{i=1}^{n} I_i(\theta_k) x_i - T(\theta_k) \right| \leq y \quad \forall k, \tag{9}$$

where y is the maximum distance between the test information function and the target information function T for the k ability points. If no targets for the test information functions are defined a Maximin approach can be applied:

$$\max y \tag{8'}$$

subject to:

$$\sum_{i=1}^{n} I_i(\theta_k)x_i \geq y \quad \forall k, \tag{9'}$$

In this approach the minimum value of the test information function over the k ability points is maximized.

These three objective functions are most commonly used in optimal test construction. The number of objective functions can easily be extended because almost every property of the test can be used to define an objective function. In some optimal test construction problems, even multiple objectives are necessary (Veldkamp, 1999). Later, some other examples objective functions are given, but first several possible constraints are introduced.

Constraints based on test specifications

Test specifications can be categorized in several ways. In this chapter, they are categorized based on properties of the test construction model. Three kinds of constraints are distinguished.

First, categorical constraints can be distinguished. Categorical item attributes partition the pool in a number of subsets. Examples of categorical item attributes are item content, cognitive level, item format, author, or answer key. In a categorical constraint, the number of items in a category is specified:

$$\sum_{i \in V_c} x_i \leq n_c \quad c = 1,...,C \tag{10}$$

where V_c is the subset of items in category c, n_c is an upper bound to the number of items chosen from category c, and C denotes the number of categories.

The second kind of constraints are the quantitative constraints. These constraints do not impose direct bounds on the numbers of items, but on a function of

the items. Examples of quantitative attributes are word count, exposure rates, expected response times, but also item parameters. To limit the expected response time for a test, the following constraint can be added:

$$\sum_{i=1}^{n} t_i x_i \leq T_t \tag{11}$$

The sum of the expected response times t_i, is bounded from above by a time limit T_t. Quantitative constraints are usually indexed with the symbol q.

The third type of constraints deal with inter-item dependencies. They are also called logical constraints. If, for example, one item contains a clue to the solution to another, these items can not be selected for the same test. If one item is chosen, the other one should thus be excluded. An example of an inclusion constraint deals with item sets. If one item in the set is chosen, all items in the set have to be chosen. These kind of constraints can be formulated as:

$$x_i \leq y_l \quad \forall i \in V_l \tag{12}$$

$$\sum_{i \in V_l} x_i = n_l y_l \tag{13}$$

where V_l denotes a logical set l, and n_l defines the number of items to be chosen from the set. The variable y_l is equal to 1, if an item from the set is chosen. Equation 13, implies that if one item is chosen, all items will be chosen. For an exclusion constraint $n_l = 1$, for an inclusion constraint n_l is equal to the number of items to be chosen from the set.

For an overview of different constraints, see van der Linden and Boekkooi-Timminga (1989).

General model for construction linear test forms.

Now that the objective functions and constraints have been formulated, a general model for optimal test construction can be given. In this model generic constraints will be used to denote the different kinds of constraints. For a typical high-stakes

achievement test, such as the LSAT, the GMAT or the TOEFL, the total number of constraints easily runs into the hundreds. The general model can be formulated as:

$$\min y \tag{14a}$$

subject to:

$$\left|\sum_{i=1}^{n} I_i(\theta_k)x_i - T(\theta_k)\right| \le y \quad \forall k, \qquad \text{(target information)} \tag{14b}$$

$$\sum_{i \in V_c} x_i \le n_c \quad c = 1,\ldots,C \qquad \text{(categorical constraints)} \tag{14c}$$

$$\sum_{i=1}^{n} f_q(x_i) \le n_q \qquad \text{(quantitative constraints)} \tag{14d}$$

$$\sum_{i \in V_l} x_i \le n_l \qquad \text{(logical constraints)} \tag{14e}$$

$$\sum_{i=1}^{n} x_i = n_t \qquad \text{(total test length)} \tag{14f}$$

$$x_i \in \{0,1\} \qquad \text{(decision variables)} \tag{14g}$$

The constraint in Equation 14g guarantees that each item is selected or not. In the remainder of the chapter, this model will be used to formulate optimal test construction problems.

Weighted-Deviations Model

In the general model, all test specifications are considered as constraints that have to be met. For some test construction problems, the test specifications are considered to be desirable properties rather than constraints. As a result, they are allowed to be violated in the test construction process. When properties are considered as desirable properties, a weighted deviation model can be formulated (Stocking & Swanson, 1993). In this model, targets are defined for all test attributes. The objective function is a weighted sum of all violations or deviations. This model can be formulated as:

$$\min \sum_{j} w_j d_j \qquad \text{(minimize weighted deviation)} \tag{15a}$$

subject to:

$$\sum_{i \in V_c} x_i - n_c \le d_1 \qquad \text{(categorical constraints)} \tag{15b}$$

$$\sum_{i=1}^{n} f_q(x_i) - n_q \leq d_2 \qquad \text{(quantitative constraints)} \qquad (15c)$$

$$\left| \sum_{i \in V_l} x_i - n_l \right| = d_3 \qquad \text{(logical constraints)} \qquad (15d)$$

$$x_i \in \{0,1\} \quad d_j \geq 0 \qquad \text{(decision variables)} \qquad (15e)$$

where the variables d_j denote the deviations, and w_j denotes the weight of deviation j. In this model the difference between the target information function and the test information functions is formulated as a quantitative constraint. When some specifications are considered to be of paramount interest, their weights get high values. When other specifications are considered to be less important, the weights get low values. Because the specifications do not have to be met, the model is less restrictive. A less favorable feature of this model is that two different tests constructed by the same model might have different attributes.

Models for construction other testing formats.

In educational and psychological measurement a wide variety of testing formats have been developed. Models for the majority of these formats can be formulated by slightly changing the general test construction model defined in the previous section. For a number of testing formats it will be explained how they differ from the general test construction problem and how the model should be adapted to construct the desired test form.

Parallel test forms

In many applications, several linear test forms have to be constructed that meet the same set of specifications. For security reasons, several versions of the same test might be needed or the test can be offered to candidates on different occasions. When tests meet the same set of specifications they are considered to be parallel. Several definitions of parallel tests are given in the literature, but the concept of weakly parallel tests is often applied. This means that the same set of constraints is met by the tests and the test information functions are identical. If a model is developed for constructing parallel tests, a few changes have to be made to the general model stated

in Equation 14 (van der Linden & Adema, 1998). First, the decision variables have to change. Where in the general model variable x_i indicates whether or not an item is selected for the test, an additional index j is needed to determine for which test the item is selected, where j runs from 1 to the number of parallel tests that have to be constructed. The new decision variables x_{ij} are defined as

$$\begin{cases} x_{ij} = 1 & \text{item } i \text{ is selected for test } j, \\ x_{ij} = 0 & \text{otherwise.} \end{cases} \tag{16}$$

The same sets of constraints should hold for all parallel tests. However, the objective function has to change slightly. For all parallel tests, the maximum difference between the target information function and the test information function should be minimized. It might also happen that no targets for the information functions have been defined. It that case, the maximum difference between the test information functions of the parallel test can be minimized.

Tests with item sets

Some items in the pool may be grouped around a common stimulus. The stimulus can be a text passage, a table, a figure, a video or music fragment (sometimes denoted as vignettes), e.g.. Whenever the stimulus is selected for the test, all items, or at least a minimum number of items, that belong to the stimulus have to be selected. Several ways of dealing with these inclusion constraints have been presented (van der Linden, 2000). One of them was discussed above when constraints on inter-item dependencies were introduced. However, when test specifications at stimulus level have also been met, this approach does not work.

An alternative approach is to introduce decision variables z_s, where

$$\begin{cases} z_s = 1 & \text{if stimulus } s \text{ is selected,} \\ z_s = 0 & \text{otherwise.} \end{cases} \tag{17}$$

Categorical, quantitative, and logical constraints can be formulated both at stimulus and item level. To make sure that the relation of inclusion between the stimulus and the items is also met the following constraint can be added to the model:

$$\sum_{i \in V_s} x_i = n_s z_s \qquad (18)$$

Whenever stimulus s is selected, this constraint guarantees that n_s items from V_s, the set of items that belong to the stimulus, are selected.

Classical test construction

Even though classical item parameters depend on the population and the other items in the test, in practice classical test theory is often applied to construct tests. When the assumption can be made that the population for the test does hardly change, test construction may be possible for classical test forms. In general, the objective function for these tests is to optimize reliability of the test. The reliability of the test is hard to estimate, but Cronbach's α defines a lower bound to it. The objective function for maximizing Cronbach's α can be defined for a fixed length test as:

$$\max \frac{n}{(n-1)} \left[1 - \frac{\sum_{i=1}^{n} \sigma_i^2}{\left(\sum_{i=1}^{n} \sigma_i \rho_{ix} \right)^2} \right], \qquad (19)$$

where σ_i^2 is the observed score variance, and ρ_{ix} is the item test correlation of item i. These parameters are based on earlier administrations of the items. The expression for Cronbach's α is a non-linear function of the decision variables. In order to formulate the test construction problem as a linear programming problem the following modification is often made. Instead of maximizing the expression in Equation 19, the denominator of the last term is maximized and its numerator is bounded from above (Adema & van der Linden, 1989):

$$\max \sum_{i=1}^{n} \sigma_i \rho_{ix} \qquad (20)$$

subject to:

$$\sum_{i=1}^{n} \sigma_i \leq c \qquad (21)$$

Tests measuring multiple abilities.

For certain types of items, several abilities are involved in answering the item correctly. When taking a driving test, the candidate both has to master the car and to show insight in traffic. In some cases, all abilities are intentional, but in other cases, some of them are considered nuisance. When multiple abilities are involved, optimizing the information in the test is more complicated. Fisher's information measure takes the form of a matrix instead of a function (Segall, 1996). From optimum design theory several criteria for optimizing matrices are known, but they all result in non-linear and complicated objective functions. An alternative approach is to use Kullback-Leibler information instead of Fisher information (Veldkamp & van der Linden, 2002). Kullback-Leibler information is a linear expression in the decision variables even in the case of multiple abilities. The resulting test constructing model can be written as:

$$\max \sum_{i=1}^{n} KL_i(\theta) x_i, \tag{22}$$

where $KL_i(\theta)$ denotes the amount of Kullback-Leibler information of the item, and θ is a vector instead of a scalar.

Tests with equated observed-score distributions.

In many large-scale test programs observed scores are presented to the candidates. Expensive equating studies have to be carried out to make the observed scores of different test forms comparable. Adding constraints to the optimal test construction model that would guarantee equal observed-score distributions would decrease the costs of testing. It can be proven (van der Linden & Luecht, 1998) that the conditional distributions of observed scores given θ for two test forms are identical if

$$\sum_{i=1}^{n} P_i^r(\theta) = \sum_{j=1}^{n} P_j^r(\theta) \quad r = 1,\ldots,n \tag{23}$$

where $P_i^r(\theta)$ is the r-th power of $P_i(\theta)$. In practice, the impact of high powers of $P_i(\theta)$ vanishes quickly. To construct tests with equated observed score the constraints in Equation 23 should be added for several values of θ.

Computerized adaptive testing (CAT)

Computerized adaptive testing (Wainer, 1990, van der Linden & Glas, 2000) can be compared to an oral exam. Instead of a teacher, in CAT a computer algorithm adapts the difficulty of the items to the answers of the candidate. After an item is administered the ability level of the candidate is estimated. Based on this estimate the item that provides most information at the examinee's estimated ability level is selected to be presented next. CAT reduces the test length by almost 40 percent. An important dilemma in CAT is that optimal CAT construction requires sequential item selection to maximize the adaptivity of the test but simultaneous test construction to realize all the test specifications. In order to deal with this dilemma the Shadow Test Approach was developed (van der Linden & Reese, 1998). This approach consists of the following steps:

1. Initialize the estimator of the ability parameter.
2. Assemble a shadow test that meets the test specifications, contains all the items already administered and gives maximum information at the current ability estimate.
3. Administer the best eligible item in the shadow test.
4. Update the ability estimate.
5. Adjust the constraints to allow for the attributes of the items already administered.
6. Return all unused items to the pool.
7. Repeat Steps 2 – 6 until n items have been administered.

The main idea in this approach is that to maximize adaptivity, in every iteration a shadow test is constructed. This shadow test contains all previously administered items and meets the test specifications. From the unadministered items in this shadow test the next item is selected. Because of this, the complete CAT will also meet the test specifications.

In Step 2 of the algorithm a shadow test has to be constructed. The objective function in this test construction problem is to maximize the information at the current ability estimate. This objective function can be compared with the objective function in Equation 7. The difference is that the estimated ability is used instead of the cutoff-score. The constraints are formulated by the test specifications. However, an

additional constraint is needed to guarantee that all the items that are administered are contained in the test. For selecting the k-th shadow test, the following constraint is added:

$$\sum_{i \in S_{k-1}} x_i = k - 1 \tag{24}$$

where the set S_{k-1} is the set of items that have been administered in the k-1 previous iterations.

Multi-stage testing

A multi-stage test form consists of a network of item sets. The item sets are also called testlets. The path of a candidate through this network of testlets depends on the answers. So, after a testlet is finished, the next testlet is selected adaptively. An example of a network for multi-stage test form is shown in Figure 1.

==========

Insert Figure 1 at about here

==========

In the first two stages, all candidates answer the same items. From the third to the last stage, the group of candidates are assigned to low difficulty, medium difficulty, and high difficulty testlets. Based on an estimate of their ability, the candidates proceed to the high-, medium-, or low difficulty testlet. The path through the network is chosen to maximize the information in the test as well as to meet the test specifications.

The problem of constructing multi-stage tests is rather complicated. First, testlets have to be constructed from the item pool and then they have to be assigned to slots in the network. A testlet can be viewed as a small linear test with its own target information function and a small set of constraints. Constructing one testlet is a straightforward application of the model in Equation 14. However, to assemble a multi-stage test, many testlets have to be constructed. Sequentially solving the model in Equation 13 would result in high quality testlets in the beginning and low quality testlets at the end of the construction process, because high quality items would be

selected first. For simultaneous selection, decision variables x_{ij} have to be introduced, which are defined as

$$\begin{cases} x_{ij} = 1 & \text{item } i \text{ is selected for testlet } j, \\ x_{ij} = 0 & \text{otherwise.} \end{cases} \tag{25}$$

Besides, Equation 14b should be slightly changed because different target information functions are defined for different testlets. The large number of decision variables in this test construction problem makes it very hard to solve (see also, Luecht & Nungester, 1998).

Constructing rotating item pools.

When optimal test constructing methods are applied, some items turn out to be more popular than others are. A typical observation is that 30 percent of the items is selected for 70 percent of the tests. To increase the usage of the less popular items and decrease the exposure rates of the popular items a system of rotating item pools can be used. Rotating item pools are subpools that are constructed from a master pool, and these subpools rotate over timepoints and locations (Stocking & Swanson, 1998, Ariel, Veldkamp & van der Linden, submitted). The subpools have to be weakly parallel, so that they serve the test construction process equally well. Constructing parallel subpools is comparable with constructing parallel tests. The main difference is that all items have to be selected for a sub-pool. The resulting pool construction problem minimizes the maximum differences between the pool information functions.

In order to increase the number of subpools that can be constructed from a master pool, overlap between subpools can also be allowed. Popular items are only allowed to be selected for one subpool, where less popular items can be selected more often. To guide the process of selecting items for multiple subpools, the following constraint has to be added to the rotating item pool construction problem:

$$\sum_{j=1}^{\#\,\text{subpools}} x_{ij} \leq n_i \quad \forall i \tag{26}$$

Each item i can be selected for at most n_i sub-pools. The consequence of using rotating item pools is an increase in item pool usage. However, measurement

precision will slightly decrease, because the best items will be spread over the different subpools.

Algorithms and heuristics for solving optimal test construction problems.

In the previous section, models for optimal test construction were described. An important question is, how to solve the models, that is, how to construct optimal tests. Several algorithms and heuristics have been proposed. In this section 0-1 linear programming techniques, network-flow programming, and a number of heuristics will be discussed.

0-1 Linear programming techniques

When a problem is formulated in mathematical programming terms, many algorithms are available for solving the model. For example, linear programming, 0-1 linear programming, quadratic programming and interior point methods can be applied. It depends on the kind of decision variables, and on the formulation of the constraints which algorithms will perform best. In optimal test construction, the decision variables are 0-1 variables. For the general formulation of a test construction problem in Equation 14, all constraints consist of sums of decision variables. Because the general test construction model only consists of linear constraints, 0-1 linear programming techniques can be applied. 0-1 Linear programming problems are known to be NP-hard, which means that it is not guaranteed that the optimal solution is found in polynomial time. However, this is only a worst case performance (see also, Papadimitriou & Steiglitz, 1982).

To find optimal values for the decision variables standard 0-1 linear programming software such as CPLEX can be used. CPLEX employs an efficient implementation of the Branch-and-Bound (B&B) algorithm. For most test construction problems, a solution can be found in a reasonable amount of time. Only such problems as the multiple-stage testing problem are too time consuming.

Some models described in the previous section had non-linear objective functions or constraints. These have to be linearized before the 0-1 linear programming techniques can be applied.

Network-flow programming

For some special test construction problems a much faster 0-1 linear programming algorithm is available, the network-flow programming algorithm (Armstrong, Jones, & Wang, 1995). In order to apply this algorithm the model is allowed to only have categorical constraints. When this algorithm is applied, even large test construction problems with thousands of variables can be solved quickly.

Unfortunately, most optimal test construction problems also have to deal with quantitative constraints. To embed these constraints in a network-flow programming algorithm they are added to the objective function as penalty terms times a Lagrangian multiplier. For example if a time limit of thirty minutes is to be imposed on a test, the following term is added to the objective function

$$\lambda(30-\sum_{i=1}^{n} t_i x_i) \qquad (27)$$

The remaining problem is to find appropriate values for the Lagrangian multipliers λ. These values are usually found iteratively. Even when this iterative process is needed to find the solution of the test construction problem, the algorithm is fast, but the solution might accidentely suffer from constraint violation.

Logical constraints might also be part of the optimal test construction problem. Some of them can be incorporated in the same way as the quantitative constraints. When it not possible to use Lagrangian multipliers, a heuristic is needed to calculate a solution under these constraints.

Heuristics

For some optimal test construction problems, 0-1 Linear Programming techniques can not be applied because of non-linearity of the objective function or the constraints, or the techniques may need too much time. Besides, it might not be possible to formulate the problem as a network-flow model. In those cases, heuristical methods can be applied to find a solution. A heuristic is an approximation method that works fast but tends to result in a solution that is only close to optimal. In optimal test construction, the greedy algorithm, simulated annealing, and genetic algorithms have been applied successfully.

Greedy algorithms work very fast. They select items sequentially. In every iteration, the item is selected that contributes most to the objective function. The NWADH (Luecht, 1998) is a well known application of a greedy heuristic. It has also been applied very successfully in combination with the weighted-deviations model in Equation 15. However, because these heuristics operate sequantially, the algorithm may run into infeasibility problems at the end of the test.

Simulated annealing is a much more time-consuming method. First, an initial test is constructed that meets all the specifications. Than, one item is swapped with an item in the pool. If the new test performs better with respect to the objective function it is accepted, otherwise it is accepted with a probability that decreases during the test assembly process. The method stops when the probability of accepting a worse test is smaller than a lower bound.

When genetic algorithms are applied several tests are constructed that meet all the specifications. New tests are constructed by selecting one part from one test and another part from a second test. If the new test performs better with respect to the objective function it is added to the set of candidate tests. At the end, the best candidate in the set is selected.

Infeasibility analysis

Sometimes, 0-1 linear programming techniques, network-flow programming, and heuristical methods might not be able to construct a test from the item pool that meets all the test specifications. When this happens the model is said to be infeasible. The reason might be that there is a logical contradiction between some of the specifications, a writing error may have occurred in the modeling process, or the item pool may be poor. The exact reason of infeasibility is often very hard to detect. A typical test construction model might consist of thousands of variables and hundreds of constraints.

Several methods have been developed to diagnoze infeasibility (e.g. Timminga, 1998, Huitzing, submitted). Since infeasibility is always caused by the specifications in combination with the item pool, the focus of the methods is on the interaction of individual specifications as well as interaction of specifications and the item pool. The main idea in most methods is to isolate a small group of specifications that have to be modified in order to construct a test from the pool. Closer

investigations of such a group of specifications has to reveal the exact reasons of infeasibility.

Numerical Example

To illustrate the optimal test construction modeling process and some of the algorithms and heuristics, a numerical example is presented. An item pool for the ACT Assessment Program Mathematics test consisting of 176 items was calibrated using a two-dimensional version of the 2PLM. The calibration of these items was carried out using the computer program NOHARM (Fraser & McDonald, 1988). The probability of obtaining a correct answer in this model is defined as

$$P_i(\theta_1,\theta_2) := \frac{e^{a_{i1}\theta_1 + a_{i2}\theta_2 + b_i}}{1 + e^{a_{i1}\theta_1 + a_{i2}\theta_2 + b_i}} \qquad (28)$$

where a_{1i} is the discrimination index for the first ability and a_{2i} is the discrimination index in for the second ability. The parameter b_i is a difficulty parameter, and θ_1 and θ_2 are two ability parameters for each person. For the items, the content, and the item types for these items have been specified. The original pool did not contain speeded response times for the items, but to illustrate the use of quantitative constraints these were added to the pool.

The objective of the test is to measure both abilities as precise as possible, therefore the information in the test has to be maximized. Test length was 25, the time limit was one hour, items 79 and 124 contained clues to each other, Table 1 and Table 2 describe the item type and the content specifications.

==========

Insert Table 1 at about here.

==========

==========

Insert Table 2 at about here.

==========

The first step in formulating an optimal test construction problem was formulation of the objective function. In a two-dimensional context, Fisher's information for a test of n items is the following matrix:

$$I(\theta) = \begin{bmatrix} \sum_{i=1}^{n} a_{1i}^2 P_i(\theta) Q_i(\theta) & \sum_{i=1}^{n} a_{1i} a_{2i} P_i(\theta) Q_i(\theta) \\ \sum_{i=1}^{n} a_{1i} a_{2i} P_i(\theta) Q_i(\theta) & \sum_{i=1}^{n} a_{2i}^2 P_i(\theta) Q_i(\theta) \end{bmatrix}$$

(29)

where θ is a two-dimensional vector. Several approaches for optimizing this matrix have been proposed in the literature, but to illustrate the differences between the algorithms and heuristics, the D-optimality criterion (e.g. Segall, 1998) was applied. This criterion comes down to maximizing the determinant of the matrix. The determinant is a continuous function of the two person parameters. Therefore, a small grid $\theta_{st} \in \{-1,0,1\} \times \{-1,0,1\}$ of (θ_s, θ_t)-points was chosen and the minimum value of the determinant for these points was maximized.

In the set of specifications, categorical constraints were defined by the item type and content specifications, the time limit defined a quantitative constraint, and the enemies defined a logical constraint. The optimal test construction model could be formulated as:

$$\max y \tag{30}$$

subject to:

$$\sum_{i=1}^{n} a_{1i}^2 P_i(\theta_{st}) Q_i(\theta_{st}) x_i \sum_{i=1}^{n} a_{1i}^2 P_i(\theta_{st}) Q_i(\theta_{st}) x_i - \left(\sum_{i=1}^{n} a_{1i} a_{2i} P_i(\theta_{st}) Q_i(\theta_{st}) x_i \right)^2 \geq y \quad \forall s,t$$

(Determinant of the matrix) (31)

$$\sum_{i \in V_{PG}} x_i \geq 3 \qquad \text{(Plane geometry)} \tag{32}$$

$$\sum_{i \in V_{PA}} x_i \geq 3 \qquad \text{(Pre-algebra)} \tag{33}$$

$$\sum_{i \in V_{EA}} x_i \geq 3 \qquad \text{(Elementary algebra)} \tag{34}$$

$$\sum_{i \in V_{CG}} x_i \geq 4 \quad \text{(Coordinate geometry)} \quad (35)$$

$$\sum_{i \in V_{TG}} x_i \geq 4 \quad \text{(Trigonometry)} \quad (36)$$

$$\sum_{i \in V_{IA}} x_i \geq 3 \quad \text{(Intermediate algebra)} \quad (37)$$

$$\sum_{i \in V_{AN}} x_i \geq 8 \quad \text{(Analysis)} \quad (38)$$

$$\sum_{i \in V_{AP}} x_i \geq 8 \quad \text{(Application)} \quad (39)$$

$$\sum_{i \in V_{BS}} x_i \geq 5 \quad \text{(Basic Skills)} \quad (40)$$

$$\sum_{i=1}^{n} t_i x_i \leq 60 \quad \text{(Time limit)} \quad (41)$$

$$x_{79} + x_{124} \leq 1 \quad \text{(Enemy set)} \quad (42)$$

$$\sum_{i=1}^{n} x_i = 25 \quad \text{(Test length)} \quad (43)$$

$$x_i \in \{0,1\} \quad \text{(Decision variables)} \quad (44)$$

The determinant of the matrix was a non-linear function of the decision variables. This function was linearized (Veldkamp, 2002) in order to apply 0-1 linear programming techniques. The greedy heuristic and simulated annealing were also applied to construct an optimal test. The results are shown in Figure 2.

==========

Insert Figure 2 at about here

==========

The greedy heuristic performed best for this problem. For all the elements of the grid, the greedy heuristic performed slightly better than simulated annealing. Both heuristics performed much better than the linear programming approach. The performance of the methods is illustrated in Figure 2, where two test information functions are shown.

For this complicated problem, the greedy heuristic performed better than expected. Because the item pool was well suited for this problem, the heuristic did not encounter any infeasibility problems. Simulated annealing performed almost as well.

This heuristic is time-consuming, but at the end it will almost allways result in a solution close to optimallity. Linear programming did not perform that well. The reason was that the objective function for linear programming was an approximation of the D-optimality criterion. Allthough, the approximated problem was solved optimally, the solution was worse than the solutions of both heuristics which optimized the D-optimality criterion itself.

In this example, several aspects of test construction were illustrated. First, it illustratres the process of formulating a model that describes the test assembly process. Several ways of defining the objective function and the constraints are often available. In this example, it was chosen to use the D-optimality criterion and a linear approximation of the criterion. As can be seen in Figure 2, the results for the D-optimality criterion were much better than for it's linear approximation. In general, the test assembler has to be sure, that the model truly describes the problem. Otherwise, the solution will be optimal to the model, but not for the problem. The second step is choice of an assembly method. Several algorithms and heuristics are available. They all have their own merits, and might result in different tests.

Conclusion and discussion

The main issue in optimal test construction is how to formulate a test assembly model. In this chapter, models for a number of optimal test construction problems have been introduced. However, all these models are based on the general test construction model in Equation 14. They may need a different objective function, some additional constraints, or different definitions of the decision variables, but the structure of the model remains the same. When different optimal test construction problems have to be solved, the question is not how to find a new method, but how to define an appropriate objective function and a set of constraints.

In the second section, the weighted-deviations model was introduced as an alternative to the linear programming model. All the models in the third section could also be written as weighted deviation models. It depends on the nature of the specifications whether linear programming models or weighted deviation models should be applied. When the specifications have to be met, linear programming models are more suitable but when the specifications are less strict, the weighted-deviations model can be used. In practical testing situations it may even happen that a

combination of both models is applied if only some of the specifications have to be met. Both the linear programming models and the weighted deviation models can be solved by the exact algorithms and heuristics.

Finally, some remarks have to be made about the quality of optimal test construction methods. The models, algorithms and heuristics presented in this chapter are very effective in constructing optimal tests. Additional gain is possible by improving the quality of the item pool. Some efforts have already been made to develop optimal blueprint for item pool design. These blueprints combine test specifications and optimal test construction methods to develop better item pools. In doing so measurement precision is increased further.

Bibliography

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.

Ariel, A., Veldkamp, B.P., & van der Linden, W.J. (submitted). Methods for constructing sub pools in constrained adaptive testing.

Armstrong, R.D., Jones, D.H., & Wang, Z. (1995). Network optimization in constrained standardized test construction. *Applications of Management Science, 8,* 189-212.

Birnbaum, A. (1968). Some latent trait models and their use in inferring an examinee's ability. In: F.M. Lord & M.R. Novick, *Statistical theories of mental test scores* (pp. 397-479). Reading MA: Addison-Wesley.

Fraser, C. & McDonald, R.P. (1988). *NOHARM II: A FORTRAN program for fitting unidimensional and multidimensional normal ogive models of latent trait theory.* Armidale, Australia: University of New England, Centre for Behavioral Studies.

Hambleton, R.K., & Swaminathan, H. (1985). *Item response theory: principles and applications.* Boston, MA: Kluwer Academic Publishers.

Huitzing, H.A. (2002). An interactive method to solve infeasibility in linear programming test assembly models. *Submitted for publication.*

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

Luecht, R.M. (1998). Computer-assisted test assembly using optimization heuristics. *Applied Psychological Measurement, 22,* 224-236.

Luecht, R.M., & Nungester, R.J. (1998). Some practical applications of computer adaptive sequential testing. *Journal of Educational Measurement, 35,* 229-249.

Papadimitriou C.H., & Steiglitz, K. (1982). *Combinatorial optimization.* Englewood Cliffs, NJ: Prentice Hall.

Segall, D.O. (1996). Multidimensional Adaptive Testing. *Psychometrica, 61,* 331-354.

Stocking, M.L., & Swanson, L. (1993). A method for severely constrained item selection in adaptive testing. *Applied Psychological Measurement, 17,* 277-292.

Stocking, M.L., & Swanson, L. (1998). Optimal design of item pools for computerized adaptive tests. *Applied Psychological Measurement, 22,* 271-279.

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

Timminga, E. (1998). Solving infeasibility problems in computerized test assembly. *Applied Psychological Measurement, 22,* 280-291.

van der Linden, W.J. (1998). Optimal assembly of psychological and educational tests. *Applied Psychological Measurement, 22,* 195-211.

van der Linden, W.J. (2000) Optimal assembly of tests with item sets. *Applied Psychological Measurement, 24,* 225-240.

van der Linden, W.J., & Adema, J.J. (1998). Simultaneous assembly of multiple test forms. *Journal of Educational Measurement, 35,* 185-198.

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

van der Linden, W.J., & Glas, C.A.W. (2000). *Computerized adaptive testing: Theory and practice.* Boston, MA: Kluwer Academic Publishers.

van der Linden, W.J., & Luecht, R.M. (1998). Observed score equating as a test assembly problem. *Psychometrica, 63,* 401-418.

van der Linden, W.J., & Reese, L.M. (1998). A model for optimal constrained adaptive testing. *Applied Psychological Measurement, 22,* 259-270.

Veldkamp, B.P. (1999). Multiple objective test assembly problems. *Journal of Educational Measurement, 36,* 253-266.

Veldkamp, B.P. (2002). Multidimensional constrained test assembly. *Applied Psychological Measurement, 26,* 133-146.

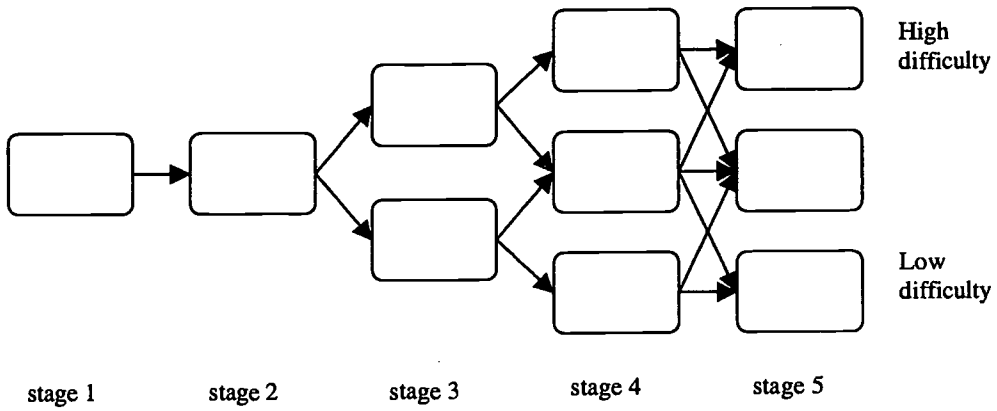
Veldkamp, B.P., & van der Linden, W.J. (2002). Multidimensional adaptive testing with constraints on test content. *Psychometrica, 67.* In press.

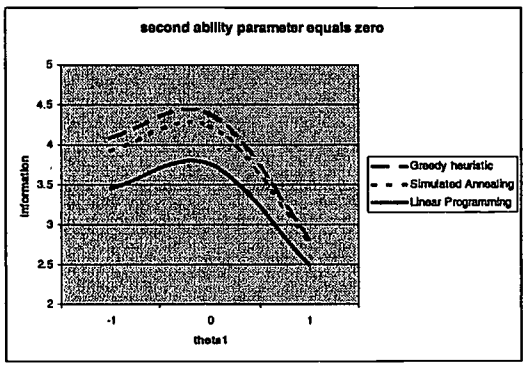
Wainer, H., Dorans, N.J., Flaugher, R., Green, B.F., Mislevy, R.J., Steinberg, L., & Thissen, D. (1990). *Computerized adaptive testing: A primer.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Figure Captions

Figure 1. Multi-stage testing format

Figure 2. Test information functions.


High
difficulty
Low
difficulty
stage 1
stage 2
stage 3
stage 4
stage 5


second ability parameter equals zero
Information
5
4.5
4
3.5
3
2.5
2
-1
0
1
theta1
Greedy heuristic
Simulated Annealing
Linear Programming

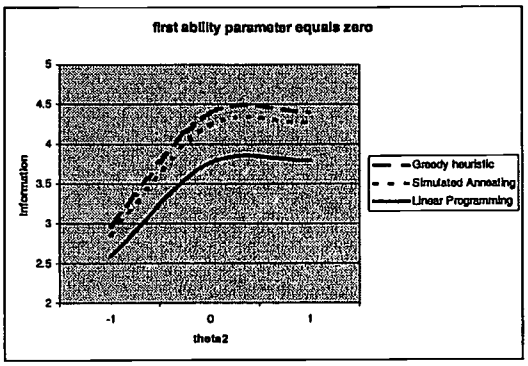

first ability parameter equals zero
Information
5
4.5
4
3.5
3
2.5
2
-1
0
1
theta2
Greedy heuristic
Simulated Annealing
Linear Programming

Table 1

Item type specifications.

Item type	Lower bound	Upper bound
Analysis	8	15
Application	8	-
Basic Skills	-	2

Table 2.

Content specifications

Content	Lower bound	Upper bound
Plane geometry	3	3
Pre-algebra	3	-
Elementary algebra	3	-
Coordinate geometry	3	-
Trigonometry	3	7
Intermediate algebra	3	-

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

RR-02-08	B.P. Veldkamp, *Optimal Test Construction*
RR-02-07	B.P. Veldkamp & A. Ariel, *Extended Shadow Test Approach for Constrained Adaptive Testing*
RR-02-06	W.J. van der Linden & B.P. Veldkamp, *Constraining Item Exposure in Computerized Adaptive Testing with Shadow Tests*
RR-02-05	A. Ariel, B.P. Veldkamp & W.J. van der Linden, *Constructing Rotating Item Pools for Constrained Adaptive Testing*
RR-02-04	W.J. van der Linden & L.S. Sotaridona, *A Statistical Test for Detecting Answer Copying on Multiple-Choice Tests*
RR-02-03	W.J. van der Linden, *Estimating Equating Error in Observed-Score Equating*
RR-02-02	W.J. van der Linden, *Some Alternatives to Sympson-Hetter Item-Exposure Control in Computerized Adaptive Testing*
RR-02-01	W.J. van der Linden, H.J. Vos, & L. Chang, *Detecting Intrajudge Inconsistency in Standard Setting using Test Items with a Selected-Response Format*
RR-01-11	C.A.W. Glas & W.J. van der Linden, *Modeling Variability in Item Parameters in Item Response Models*
RR-01-10	C.A.W. Glas & W.J. van der Linden, *Computerized Adaptive Testing with Item Clones*
RR-01-09	C.A.W. Glas & R.R. Meijer, *A Bayesian Approach to Person Fit Analysis in Item Response Theory Models*
RR-01-08	W.J. van der Linden, *Computerized Test Construction*
RR-01-07	R.R. Meijer & L.S. Sotaridona, *Two New Statistics to Detect Answer Copying*
RR-01-06	R.R. Meijer & L.S. Sotaridona, *Statistical Properties of the K-index for Detecting Answer Copying*
RR-01-05	C.A.W. Glas, I. Hendrawan & R.R. Meijer, *The Effect of Person Misfit on Classification Decisions*
RR-01-04	R. Ben-Yashar, S. Nitzan & H.J. Vos, *Optimal Cutoff Points in Single and Multiple Tests for Psychological and Educational Decision Making*
RR-01-03	R.R. Meijer, *Outlier Detection in High-Stakes Certification Testing*

RR-01-02 R.R. Meijer, *Diagnosing Item Score Patterns using IRT Based Person-Fit Statistics*

RR-01-01 H. Chang & W.J. van der Linden, *Implementing Content Constraints in Alpha-Stratified Adaptive Testing Using a Shadow Test Approach*

RR-00-11 B.P. Veldkamp & W.J. van der Linden, *Multidimensional Adaptive Testing with Constraints on Test Content*

RR-00-10 W.J. van der Linden, *A Test-Theoretic Approach to Observed-Score Equating*

RR-00-09 W.J. van der Linden & E.M.L.A. van Krimpen-Stoop, *Using Response Times to Detect Aberrant Responses in Computerized Adaptive Testing*

RR-00-08 L. Chang & W.J. van der Linden & H.J. Vos, *A New Test-Centered Standard-Setting Method Based on Interdependent Evaluation of Item Alternatives*

RR-00-07 W.J. van der linden, *Optimal Stratification of Item Pools in a-Stratified Computerized Adaptive Testing*

RR-00-06 C.A.W. Glas & H.J. Vos, *Adaptive Mastery Testing Using a Multidimensional IRT Model and Bayesian Sequential Decision Theory*

RR-00-05 B.P. Veldkamp, *Modifications of the Branch-and-Bound Algorithm for Application in Constrained Adaptive Testing*

RR-00-04 B.P. Veldkamp, *Constrained Multidimensional Test Assembly*

RR-00-03 J.P. Fox & C.A.W. Glas, *Bayesian Modeling of Measurement Error in Predictor Variables using Item Response Theory*

RR-00-02 J.P. Fox, *Stochastic EM for Estimating the Parameters of a Multilevel IRT Model*

RR-00-01 E.M.L.A. van Krimpen-Stoop & R.R. Meijer, *Detection of Person Misfit in Computerized Adaptive Tests with Polytomous Items*

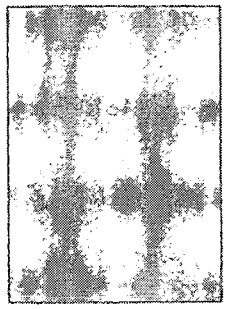
RR-99-08 W.J. van der Linden & J.E. Carlson, *Calculating Balanced Incomplete Block Designs for Educational Assessments*

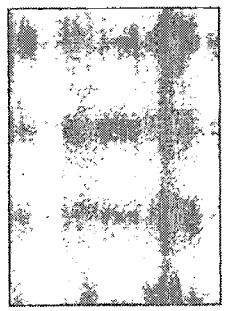
RR-99-07 N.D. Verhelst & F. Kaftandjieva, *A Rational Method to Determine Cutoff Scores*

RR-99-06 G. van Engelenburg, *Statistical Analysis for the Solomon Four-Group Design*

RR-99-05 E.M.L.A. van Krimpen-Stoop & R.R. Meijer, *CUSUM-Based Person-Fit Statistics for Adaptive Testing*

Research Reports can be obtained at costs, Faculty of Educational Science and Technology, University of Twente, TO/OMD, P.O. Box 217, 7500 AE Enschede, The Netherlands.





faculty of

EDUCATIONAL SCIENCE AND TECHNOLOGY

A publication by
The Faculty of Educational Science and Technology of the University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

BEST COPY AVAILABLE

TM034735

U.S. Department of Education
Office of Educational Research and Improvement (OERI)
National Library of Education (NLE)
Educational Resources Information Center (ERIC)

ERIC®

NOTICE

Reproduction Basis

☒ This document is covered by a signed "Reproduction Release (Blanket)" form (on file within the ERIC system), encompassing all or classes of documents from its source organization and, therefore, does not require a "Specific Document" Release form.

☐ This document is Federally-funded, or carries its own permission to reproduce, or is otherwise in the public domain and, therefore, may be reproduced by ERIC without a signed Reproduction Release form (either "Specific Document" or "Blanket").

EFF-089 (3/2000)

ERIC