Optimizing Balanced Incomplete Block Designs for Educational Assessments

Wim J. van der Linden, University of Twente
Bernard P. Veldkamp, University of Twente
James E. Carlson, National Assessment Governing Board

A popular design in large-scale educational assessments as well as any other type of survey is the balanced incomplete block design. The design is based on an item pool split into a set of blocks of items that are assigned to sets of "assessment booklets." This article shows how the problem of calculating an optimal balanced incomplete block design can be formulated as a problem in combinatorial optimization. Several examples of structural and practical requirements for balanced incomplete block designs are shown to be linear constraints on the optimization problem. In addition, a variety of possible objective functions to optimize the design is discussed. The technique is demonstrated using the 1996 Grade 8 Mathematics National Assessment of Educational Progress (NAEP) as a case study. *Index terms: balanced incomplete block design, large-scale educational assessments, (mixed) integer programming, optimal design.*

For large-scale educational assessments, the simultaneous sampling of items and students is the only practical way to obtain representative indications of student performance across a wide spectrum of subject matter. Typically, sampling of students takes place through a complex probabilistic, multistage sampling plan involving several levels of units. A description of the sampling plan used for sampling students in the National Assessment of Educational Progress (NAEP) is given in Rust and Johnson (1992).

When educational assessments were still based on classical test theory, the parameters of interest were the mean scores of the population of students on the individual items in the pool. An efficient strategy for estimating these parameters is multiple-matrix sampling. In multiple-matrix sampling, both the students and the items are sampled randomly by assigning subsets of items to subsets of students (Sirotnik, 1974). An important result on multiple-matrix sampling was given in Lord (1962; see also Lord & Novick, 1968, sec. 11.12), who showed that the mean scores of a population of students on a pool of items are estimated best if each single item is administered to a random, nonoverlapping subset of students. In practice, this design is not feasible because of the complicated logistics involved in delivering single items to examinees, but it has served as an important benchmark in evaluations of sampling designs for classical educational assessments.

With the advent of item response theory (IRT) (e.g., Birnbaum, 1968; Lord, 1980), the interest in educational assessments shifted from mean scores on individual items to the full population distribution on the person parameter in the model, θ. One of the features of IRT helpful in educational assessments is that, although different item-student combinations yield different statistical precision,

Applied Psychological Measurement, Vol. 28 No. 5, September 2004, 317–331
DOI: 10.1177/0146621604264870

random assignment of items to students is not a necessary condition for consistent estimation of the θ distribution. Hence, a possible approach is to assemble booklets with assessment items from a pool according to some practical principle and assign them to students optimizing, for example, an important statistical psychometric objective for the assessment.

Both in the NAEP in the United States and in the Dutch *Periodiek Peilingsonderzoek van het Onderwijs* (PPON) projects, tests are assembled following the structure of a "balanced incomplete block" (BIB) design (Johnson, 1992; Wijnstra, 1988). The design assumes that the pool of items is split into a set of blocks. The split need not be random but may be based on such practical issues as the wish to offer students blocks with motivating combinations of items or to match blocks across booklets with respect to the time needed to complete them. Also, the number of booklets that have to be designed is predetermined. Finally, booklets are spiraled across students in the lowest unit (usually school classes) to minimize the cluster effects involved in sampling a hierarchically structured population.

It should be noted that this use of the term "*BIB design*" is not in agreement with the definition of this term in the literature on experimental design (e.g., Winer, 1970, sec. 9.5). One of the differences between their definitions is an interchange of the roles of the blocks and treatments. However, because both types of designs share important analogies and the use of the term is well established in the educational assessment literature, this article will follow the tradition and refer to an assessment design as a BIB design if the assignment of blocks to booklets is controlled by the following constraints: (a) the number of blocks assigned to each booklet is between certain bounds, (b) the number of booklets to which each block is assigned is between certain bounds, and (c) combinations of blocks are assigned to a minimum number of booklets. The third type of constraint is needed only if statistical relations between items in different blocks (e.g., their covariances) have to be estimated, which is often the case. This set of constraints will be referred to as *structural constraints*. Figure 1, which is derived from Johnson (1992, Fig. 1), gives an example of a BIB design according to the definition that has become standard in the educational assessment literature.

If no other constraints had to be imposed on BIB designs, the actual assignment of the blocks to the assessment booklets would be a simple task. As the example in Figure 1 suggests, a procedure in which the blocks are systematically rotated across the booklets would already do. However, in practice, several additional constraints—for example, on item content, format, and response time—may have to be imposed on the composition of the booklets. Such constraints will be referred to as *practical constraints*. If both structural and practical constraints are to be imposed on the assignment of the blocks to the booklets, the assignment process quickly becomes too complicated for manual execution. The conclusion holds a fortiori if the assignments have to be optimized with respect to an important psychometric objective.

The purpose of this article is to show how techniques from combinatorial optimization (e.g., Nemhauser & Wolsey, 1999) can be exploited to assemble sets of booklets following a BIB design. In the remainder of this article, first several practical constraints on BIB design and possible objective functions are discussed. Then, a general optimization model for assembling booklets from a pool of blocks is introduced. The article concludes with the discussion of a case study based on a pool of blocks from the 1996 Grade 8 Mathematics NAEP.

Although this article was motivated by an interest in optimizing educational assessments and derives its terminology and empirical examples from this type of survey, the methodology in this article applies to any other large-scale survey for which a balanced incomplete block design is possible. Examples of such surveys include quality-of-life surveys, social indicator studies, and marketing studies into consumer behavior.

Figure 1
Feasible Balanced Incomplete Block Design for the National Assessment of Educational Progress (NAEP) Grade 8 Mathematics Project

Booklets	Blocks												
	1	2	3	4	5	6	7	8	9	10	11	12	13
1				x	x				x				
2				x				x			x		
3		x		x								x	
4	x				x			x					
5				x						x			x
6							x				x	x	
7	x		x						x				
8	x			x			x						
9		x			x					x			
10							x	x		x			
11						x		x					x
12	x									x	x		
13					x	x	x						
14					x							x	x
15			x							x		x	
16			x		x						x		
17		x					x		x				
18			x	x		x							
19		x	x					x					
20	x	x											x
21			x				x						x
22		x				x					x		
23									x		x		x
24								x	x			x	
25	x					x						x	
26						x			x	x			

Some Practical Constraints and Objective Functions

Practical constraints on test assembly can be classified in various ways. A convenient classification is the following (van der Linden, 1998, 2004): First, constraints can be based on categorical item attributes, such as item content, format, cognitive level, and whether an item has graphics. Each categorical attribute is represented by a class of items in the pool, and constraints on these attributes specify a desired distribution of the items in the assessment over these classes. Second, constraints can be based on quantitative item attributes—that is, on parameters or

coefficients with numerical values—such as item p values, word counts, and (expected) response times. Quantitative constraints require sums or averages of attribute values to be between certain bounds. Third, logical (or Boolean) constraints deal with certain dependencies between the items in the pool. Two important cases are items organized around common stimuli ("item sets") and items that cannot be in the same form because of content overlap ("enemies"). Fourth, constraints can be used to set the length of the test form or some of its sections to a prespecified number of items.

If assessment booklets are assembled from a set of blocks, the main focus may be on constraints on categorical and quantitative attributes. These first two types of constraints can be formulated at different levels in the tests, ranging from item set to booklet level. Logical constraints are relevant, for example, if an item in one block contains a clue to the solution to an item in another block. If so, these blocks should be treated as enemies themselves. Item sets occur only within blocks and therefore need no special concern when blocks are combined into booklets. Finally, if the blocks are matched on the time needed to complete them, the constraints on test length in the last category boil down to those on the number of blocks per booklet. However, an alternative to matching blocks on time is to leave the number of items per block free, use these numbers as an attribute, and constrain their sum per booklet.

Examples of several of these above types of constraints are given in the 0-1 LP model for calculating BIB designs below.

Possible Objective Functions

In combinatorial optimization, solutions satisfying the full set of constraints are known as feasible solutions. An objective function is used to find an optimum in the set of feasible solutions. For the current problem of calculating a BIB design, if there exist no further preferences between designs that meet the constraints, all feasible solutions are equally good. In this case, an arbitrary objective function defined on (a subset of) the decision variables can be used to find a solution.

However, though not common practice in current assessment programs, if a solution is to be found using combinatorial optimization, it makes sense to exploit the need of an objective function and optimize the design with respect to an important statistical or psychometric objective. In the following sections, a few possible objective functions are discussed.

Estimating Subpopulation Densities

In most assessments, the primary goal is to estimate the distribution of person trait levels in one or more populations. In the IRT framework, these distributions are represented by density functions $g(\theta)$, where θ is the trait level variable (see the model equation in (1) below). Typically, these density functions have multiple parameters, and the goal of the assessment is thus is to estimate the parameters from the response data. Minimization of a suitable function of the covariance matrix of the (MML) estimators of these parameters, such as its determinant or trace, is an obvious choice of objective function. This choice makes sense in particular if multiple distributions have to be evaluated and different booklets have to be optimized with respect to different distributions. Because this objective function is nonlinear, the problem typically has to be solved by a heuristic, or a linear approximation to the objective function has to be introduced (for examples in IRT-based test assembly, see van der Linden, 1996; Veldkamp, 2002).

Reporting Individual Performances

If the interest is in reporting trait levels at the level of distributions not only of certain populations but also of individual students, it becomes useful to increase the efficiency of the

individual scores maximizing the booklet information functions over well-chosen intervals. A favorable side effect of the introduction of this objective is improved estimation of the scores and, consequently, an increase in the robustness of marginal analyses of group differences against model misspecifications (Mislevy, Beaton, Kaplan, & Sheehan, 1992).

Test information functions have been used as an objective function in a variety of test assembly problems (van der Linden, 1998, 2004). Formulating an objective function for this application, which should have a lower bound on the number of items per examinee to guarantee sufficient accuracy for the estimator of θ, does not involve anything new. A further discussion of this objective is therefore omitted here.

Motivating Student Participation

Students' participation in educational assessments is obligatory, and careless, unmotivated test taking is a known problem in such studies. Careless behavior is expected to increase if the items are much too easy, and motivation decreases if they are much too difficult for the students. It may therefore pay off to design the assessment booklets such that each population receives items with a probability of success as close as possible to an optimum, say, .60. (The best target value may be different for different types of students or subject areas.) Typically, in a running assessment program, subpopulations can be identified with low and high trait levels from a previous assessment. If the items in the booklets are calibrated using an IRT model such as the one in (1), it is possible to predict the probabilities of success on the items for examinees with trait levels typical of these populations (e.g., their average on a previous assessment). In this case, an obvious goal is to assemble booklets for populations using an objective function that minimizes the distances between the probabilities of success on the items for typical trait levels and target values for these probabilities. This type of objective function was used in the case study reported below.

It should be observed that this type of objective function can also have a favorable impact on the booklet information functions. For the IRT model in (1), a booklet has maximum information for student-item combinations with probabilities of success close to .50.

Controlling Speededness

As a last example, an application to controlling the speededness of the assessment tests is discussed. If these tests are speeded, too many examinees may not reach the final items in the tests. However, if estimates of the time needed to complete the items are available for the various populations of students, it is possible to introduce an objective function that optimizes the match between the expected time the booklets require and the slots available for administering them.

This type of objective function is possible if the items have been pretested to obtain empirical estimates of their response time distributions or if good subjective estimates exist. For an example of test assembly based on empirical estimates of response time distributions, see van der Linden, Scrams, and Schnipke (1999).

Optimization Model for Balanced Incomplete Block Designs

A general framework for combinatorial optimization of BIB designs is presented. It is assumed that the items have been calibrated previously using the three-parameter logistic (3PL) model:

$$p_i(\theta) \equiv \Pr\{U_i = 1\} \equiv c_i + (1 - c_i)\frac{\exp[a_i(\theta - b_i)]}{1 + \exp[a_i(\theta - b_i)]}, \tag{1}$$

where U_i is the response variable for item i, with $U_i = 1$ for a correct and $U_i = 0$ for an incorrect response; $\theta \in R$ is the examinee parameter; and $a_i \in (0, \infty)$, $b_i \in R$, and $c_i \in [0, 1)$ are the discrimination, difficulty, and guessing parameter for item i, respectively (e.g., Lord, 1980).

In addition, the following notation is needed. The individual blocks in the pool are represented by indices $j = 1, \ldots, N$. To represent pairs of blocks, a second index $k = 1, \ldots, N$ is used. Booklets are denoted by $b = 1, \ldots, B$. Binary variables x_{jb} are used to decide whether ($x_{jb} = 1$) or not ($x_{jb} = 0$) block j is assigned to booklet b. Likewise, binary variables z_{jkb} are used to assign pair (j, k) to booklet b. Special constraints will be formulated below to keep the values of these two categories of variables consistent.

The distribution of blocks across booklets is described by the following numbers:

c_1: number of blocks per booklet,

c_2: number of booklets per block,

c_3: number of booklets per pair of blocks.

To illustrate the possibility to control the contents of the booklets, three different kinds of additional constraints are introduced. First, it is assumed that the blocks are classified by content. Content is represented by a categorical attribute $c = 1, \ldots, C$, where V_c is defined as the subset of blocks in the pool belonging to content category c, and n_c is the number of blocks to be selected from V_c. Second, to illustrate the treatment of a categorical attribute, it is assumed that the booklets have to be controlled for response time. The response time permitted for block j is denoted as q_j, whereas the total amount of time permitted for booklet b is T_b. Finally, it is assumed that some blocks are "enemies" in the sense that they cannot be assigned to the same booklet. The sets of indices of enemies are denoted by V_e, $e = 1, \ldots, E$. These types of constraints are given here only as examples of additional features that could be implemented in assessment studies. They are not used in current assessment projects such as those in the NAEP.

As an example of an objective function, the case of minimizing the differences between the probabilities of success on the items and their target values is used. Let τ_b be the target for the success probabilities on the items in booklet b, and let θ_b^* be a typical value for the students for which booklet b is designed. Finally, the set of indices of the items in block j is denoted as V_j, and it is assumed that block j has n_j items.

In the case study below, the items were generally easy for a typical student (see Table 1). This practice is common in assessment studies. If this occurs, the objective function in the following model capitalizes on this feature and minimizes the distances between the probabilities of success and the target values from above:

$$\text{minimize} \left(NB \sum_j n_j \right)^{-1} \sum_b \sum_j \left[\sum_{i \in V_j} P_i(\theta_b^*) - \tau_b \right] x_{jb} \quad \text{(objective function)} \tag{2}$$

subject to

$$\sum_{j=1}^{N} x_{jb} = c_1, \quad b = 1, \ldots, B, \quad \text{(\# blocks per booklet)} \tag{3}$$

$$\sum_{b=1}^{B} x_{jb} \leq c_2, \quad j = 1, \ldots, N, \quad \text{(\# booklets per block)} \tag{4}$$

$$\sum_{b=1}^{B} z_{jkb} \geq c_3, \quad j < k = 1, \ldots, N, \quad \text{(\# booklets per pair)} \tag{5}$$

$$x_{jb} + x_{kb} \geq 2z_{jkb}, \quad j < k = 1, \ldots, N,\ b = 1, \ldots, B, \quad \text{(consistent assignment)} \tag{6}$$

$$\sum_{j \in V_c} x_{jb} \geq n_c, \quad c = 1, \ldots, C,\ b = 1, \ldots, B, \quad \text{(content attributes)} \tag{7}$$

$$\sum_{j=1}^{N} q_j x_{jb} \leq T_b, \quad b = 1, \ldots, B, \quad \text{(response time)} \tag{8}$$

$$\sum_{j \in Ve} x_{jb} \leq 1, \quad e = 1, \ldots, E,\ b = 1, \ldots, B, \quad \text{(enemies)} \tag{9}$$

$$x_{jb} \in \{0, 1\}, \quad j = 1, \ldots, N,\ b = 1, \ldots, B, \quad \text{(definition } x_{jb}\text{)} \tag{10}$$

$$z_{jkb} \in \{0, 1\}, \quad j < k = 1, \ldots, N,\ b = 1, \ldots, B. \quad \text{(definition} z_{jkb}\text{)} \tag{11}$$

The constraints in (3) and (4) define the size of the booklet in terms of the numbers of blocks and the number of times a block is assigned to a booklet, respectively, whereas (5) sets the minimum number of booklets to which each possible pair is assigned equal to c_3. The constraints in (6) stipulate that a pair of blocks is assigned ($z_{jkb} = 1$) only if both individual blocks are assigned ($x_{jb} = 1$ and $x_{kb} = 1$). Observe that for each pair (j, k), it is still possible to assign one of the blocks without assigning the entire pair. If this option is not desired, the inequality in (6) should be replaced by an equality.

Due to the constraints in (7), at least n_c blocks from the content category are assigned to a booklet, and the constraints in (8) guarantee that for booklet b, no more than T_b minutes are needed. The constraints in (9) prevent from assigning more than one block from each set of enemies. Finally, the constraints in (10) and (11) define the ranges of the decision variable.

Because all decision variables are 0-1, the model in (2) through (11) belongs to the class of linear integer programming (IP) models (e.g., Nemhauser & Wolsey, 1999).

Alternative Objective Function

If the items are relatively too easy for the typical examinees, the probabilities $P_i(\theta_b^*)$ are larger than their target values τ_b, and the objective function in (2) leads to a solution for which the distance between the probabilities and target values is minimal. On the other hand, if the items show a larger variety of difficulty and the probabilities tend to be centered around the target values, the distances can compensate, and the following objective function is recommended:

$$\text{minimize } y \tag{12}$$

subject to

$$\left[n_j^{-1} \sum_{i \in V_j} P_i(\theta_b^*) - \tau_b \right] x_{jb} \leq y, \quad b = 1, \ldots, B,\ j = 1, \ldots, N, \tag{13}$$

$$\left[n_j^{-1} \sum_{i \in V_j} P_i(\theta_b^*) - \tau_b \right] x_{jb} \geq -y, b = 1, \ldots, B,\ j = 1, \ldots, N, \tag{14}$$

$$y \geq 0. \tag{15}$$

The constraints in (13) through (15) require the sum of the differences between the actual average success probabilities and the targets to be in the interval $[-y, y]$. The size of this interval is minimized in the objective function in (12). This objective function, along with the constraints in (13) and (14), is of the maximin type. It minimizes the maximum deviation between the targets and success probabilities across all booklets.

Although the replacement of (2) by (12) through (15) does keep the model linear, it introduces a real-valued decision variable. The problems therefore now belong to the class of mixed-integer programming (MIP). The introduction of a real-valued variable may slightly complicate the solution process for some algorithms.

Algorithms

Branch-and-Bound

Problems as in (2) through (11) or (12) through (15) are NP-hard; that is, their solution time is not bounded by a polynomial in the size of the problem. The size of the problem is dependent on the number of variables in the model. For the model in (2) through (11), the number is equal to $BN[(N + 1)/2]$—namely, BN variables x_{jb} and $BN(N - 1)/2$ variables z_{jkb}. In the case study below, B was equal to 26 and N to 13, yielding a model with 2,366 variables. If the number of variables is not too large, the search for a solution to the problem can be found using a branch-and-bound algorithm (Nemhauser & Wolsey, 1999, sec. II.4.2). A powerful optimizer based on this type of algorithm, which has been found capable of solving problems with up to a few thousand variables successfully, is available in CPLEX 8.1 (ILOG, 2002).

However, if the number of constraints is also large, search algorithms may run into occasional memory overflow. The number of constraints in the core of the current model—that is, (3) through (6)—is equal to $(2B + N)B + B$. In the case study below, the number was equal to 390. This number is not particularly large, but adding practical constraints to this part of the model, particularly when they are applied at the item level, can increase the number of constraints quickly.

Finally, the structure of the constraints plays an important role. For example, the presence of equality constraints can lead to a severely constrained problem that occasionally is difficult to solve for optimality. The case study in the next section illustrates this point. Hence, as an alternative to a branch-and-bound algorithm, its solution was found using a heuristic.

Simulated Annealing

A powerful heuristic, proven to be widely applicable in combinatorial optimization, is simulated annealing (for an introduction, see, e.g., Aarts & Lenstra, 1997). Annealing is a thermal process for obtaining a low-energy state of a solid body in a heat bath in condensed-matter physics. The state is obtained by melting the solid followed by cooling it according to a carefully designed cooling schedule. Simulation of this process using a Monte Carlo technique was introduced by Metropolis, Rosenbluth, Teller, and Teller (1953). The analogy with an iterative solution process in a combinatorial minimization problem arises if one equates the iteration steps in the minimization process to the thermodynamic states of the solid body, the value of the objective function to its energy level of the body, and the global minimum to its final low-energy state.

As is typical of the Metropolis et al. (1953) technique, which also forms the basis of the recent wave of Markov chain Monte Carlo (MCMC) techniques for posterior computation in Bayesian statistics (e.g., Gilks, Richardson, & Spiegelhalter, 1996), at each iterative step, a new solution is proposed that is then accepted or rejected with a certain probability. In combinatorial optimization, these proposals are created by random perturbation of the previous candidate solution.

In summary, for a minimization problem, a simulated annealing algorithm operates as follows:

(1) Initialize control parameter t.

(2) Choose an incumbent solution and calculate its value for the objective function.

(3) For $l := 1$ to L,
   (a) perturb the solution randomly;
   (b) calculate the value of the objective function for the proposed solution;
   (c) if the proposed solution has a lower value than the incumbent solution, accept the proposed solution as the new incumbent solution;
   (d) if the proposed solution has a larger value than the incumbent solution, accept the proposed solution as the new incumbent with a probability $p(t)$, which is a decreasing function of t, and keep the incumbent solution otherwise.

(4) Decrease the value of t according to $t := ct$, where c is a constant satisfying $0 < c < 1$, and repeat Step 3 until $ct < t_{\min}$.

The outer loop decreases the control parameter t and, hence, the probability of accepting a proposed solution with a worse value for the objective function. The inner loop runs the Metropolis Monte Carlo process of proposing and accepting solutions with a given probability. The fact that the algorithm sometimes accepts worse solutions is critical because it avoids it ending in a local minimum.

The probability of accepting a worse solution is generally chosen to be

$$p(t) = \exp\left[\frac{o(i) - o(p)}{t}\right], \tag{16}$$

where $o(i)$ and $o(p)$ are the values of the objective function for the incumbent and proposed solution, respectively. This expression is derived from the Boltzmann distribution law in physics, in which the denominator in the exponent is $k_B t$, with k_B and t representing the Boltzmann constant and the temperature of the solid body, respectively. The feature that makes (16) suitable for application in combinatorial optimization is the fact that, for $o(i) > o(p)$, parameter t defines a family of similarly ordered functions. Hence, after each Step 4, lowering t gives probabilities of accepting a worse solution that are always uniformly lower in the difference $o(i) - o(p)$.

When implementing a simulated annealing algorithm, appropriate values for the parameters L, c, and $t_{\min}$ have to be chosen. These values have an impact on the efficiency of the algorithm. The smaller the value of c, the larger the steps at which t, and hence the probability of accepting a worse solution, decrease. The choice of the value of L has the same effect. However, if this probability decreases too fast, the likelihood of ending in a local minimum increases. Also, the smaller the values of $t_{\min}$, the longer the algorithm runs. However, for values of $t_{\min}$ close to zero, hardly any worse solution gets accepted, and the incumbent solution hardly changes.

Case Study: 1996 Grade 8 Mathematics NAEP

The goal of this study was to provide a post hoc illustration of the optimization of the design for the 1996 Grade 8 Mathematics NAEP project (Reese, Miller, Mazzeo, & Dossey, 1997). The pool for this assessment consisted of 13 blocks of dichotomously and polytomously scored items, which had been combined in 26 booklets in the NAEP assessment. All dichotomously scored items

were calibrated using the 3PL model in (1) for the dichotomously scored items and the generalized partial-credit model for the polytomously scored items (Muraki, 1992, 1997). In all, the pool had 139 dichotomously and 25 polytomously scored items. The following scales were needed to calibrate the item pool: (a) Number, Sense, and Operations; (b) Measurement; (c) Geometry and Spatial Sense; (d), Data Analysis, Statistics, and Probability; and (e) Algebra and Functions.

The model used to calculate an optimal BIB design was the one in (2) through (6), with the definitions of the decision variables in (10) and (11). An objective function was formulated to select the blocks with average probabilities of success as closely as possible to .50 on the dichotomously scored items for typical θ values in the subpopulations of students. For the polytomously scored items, the differences between the expected scores and the midpoint of their score intervals were minimized. To remove the effects of scale differences between the polytomous and dichotomous scores in (3) and (4), the expected scores and midpoints on the polytomously scored items were scaled back to [0,1). The subpopulations were fictitious; they were chosen to be functioning at the 25th, 50th, and 75th percentiles of the national distributions on the five mathematics scales in the 1996 NAEP assessment. The total number of booklets assembled was equal to 26. Ten booklets were assembled to be optimal at the 50th percentile of the population and 8 booklets each at the 25th and 75th percentiles.

Analysis of Optimization Problem

The problem led to the following choices for the optimization model: The target value for the probabilities of success on the items (for the polytomously scored items, this included target values of the expected scores) was set equal to $\tau = .50$. For the definition of the subpopulations of examinees above, this choice involved success probabilities for the items that were all larger than this target value (see Table 1, presented later). The objective function chosen was therefore the one in (2), which guaranteed approximation of the target value from above.

The specifications for the numbers of blocks and booklets were the regular specifications used in the 1996 assessment. That is, the number of blocks per booklet was set equal to $c_1 = 3$, an upper limit of $c_2 = 6$ booklets was imposed on the number of times a block could be assigned to a different booklet, and the number of times each pair of blocks was assigned to a common booklet was set equal to $c_3 \geq 1$. The 1996 assessment involved no further constraints on booklet content or on any block or item attributes.

The specifications led to the following set of constraints:

$$\sum_{j=1}^{13} x_{jb} = 3, \quad b = 1, \ldots, 26, \text{(\# blocks per booklet)} \tag{17}$$

$$\sum_{b=1}^{26} x_{jb} \leq 6, \quad j = 1, \ldots, 13, \quad \text{(\# booklets per block)} \tag{18}$$

$$\sum_{b=1}^{26} z_{jkb} \geq 1, \quad j < k = 1, \ldots, 13, \quad \text{(\# booklets per pair)} \tag{19}$$

$$x_{jb} + x_{kb} \geq 2z_{jkb}, \quad j < k = 1, \ldots, 13, \; b = 1, \ldots, 26, \quad \text{(consistent assignment)} \tag{20}$$

$$x_{jb} \in \{0, 1\}, \quad j = 1, \ldots, 13, \; b = 1, \ldots, 26, \quad \text{(definition } x_{jb}\text{)} \tag{21}$$

$$z_{jkb} \in \{0, 1\}, \quad j < k = 1, \ldots, 13, \; b = 1, \ldots, 26. \quad \text{(definition } z_{jkb}\text{)} \tag{22}$$

The total number of decision variables and constraints in the model was equal to 2,366 and 390, respectively. Although these numbers are not unfavorable for using the MIP optimizer in the CPLEX 8.1 (ILOG, 2002) software discussed earlier, the method of simulated annealing was used because the problem is actually much more severely constrained than the set of constraints in (17) through (22) suggests. The constraints in (19) are on the number of times a pair of blocks has to be assigned to a booklet. Because we have 13 blocks, the number of pairs is equal to 78. However, each block can contain at most 3 pairs. Because we have 13 blocks, they can contain exactly 78 pairs. Besides, every block has to be assigned to exactly six booklets. Thus, the constraints in (18) and (19) operate in fact as equalities and constrain the solution severely.

The consequences can be demonstrated by the difference between the number of feasible solutions for a model with and without the constraints on the pairs of blocks in (19). Without these constraints, the number of feasible solutions of the problem is 10^{65}, but if the constraint is added, the number reduces to 10^9. A branch-and-bound search for a solution of the full model would therefore have to reject a huge number of candidate solutions as infeasible and was not expected to find an optimum in realistic time.

Implementation of Method of Simulated Annealing

Instead, the method of simulated annealing was used. In this application, the parameters were set equal to $L = 150{,}000$, $c = 9$, and $t_{\min} = .00001$, whereas the process was started with t initialized at 1. This choice of values is rather conservative; this study's primary intention was not to run an efficient process but to find a good solution.

The remaining choice that had to be made was the one of a random perturbator for the algorithm. The choice was based on a feature of the representation of the BIB design as a matrix in Figure 1. If two rows of the matrix are swapped, two combinations of three blocks are assigned to different booklets, but the new design is still a BIB design that meets all constraints if the previous one did. Moreover, if the swap is within the sets of Booklets 1 through 8, 9 through 18, and 19 through 26, the value of the objective function does not change because each set had the same value θ^* for the objective function. Swapping between these sets does result in a new solution with different values for the objective function, though.

Swapping two columns in the matrix is a much more rigorous step. It results in the interchange of two blocks in the design as well as 10 new combinations of three blocks. Again, if the design is feasible, swapping of two columns keeps it feasible. For each swap, a new value for the objective function is obtained.

Swapping of two rows can be considered as fine tuning of the design, whereas swapping of two columns can be viewed as a more rigorous step that moves the solution away from a local minimum if it happens to be in one. The perturbator used in this study consisted of 1 random swap of two columns for every 100 swaps of two rows. The choice of the value of L above was not only large but also made to allow for this pertubator. Because there was only 1 column swap for each 100 row swaps, there were in fact 1,500 column swaps for each value of t. This number was deemed sufficient.

The first incumbent solution was the feasible design in Figure 1.

Results

The whole optimization process took approximately 22 hours of simulation. The solution at the end of this process is given in Figure 2. Visual comparison between the initial solution and the final result shows large differences in assignments of blocks to the booklets. The objective function had a value for the solution equal to 16.93. The average probabilities of success for the assignment of the

Table 1

Average Probabilities of Success After Assignment of Blocks to Booklets

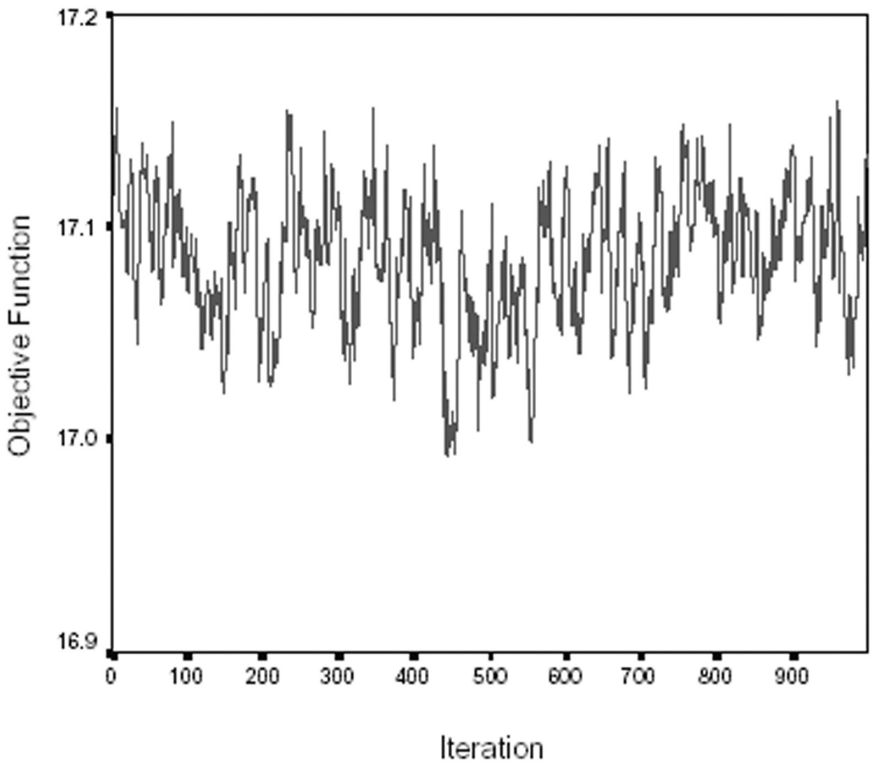
Block	Booklets		
	1–8	9–18	19–26
1	0.70	0.72	0.74
2	0.72	0.74	0.76
3	0.76	0.78	0.79
4	0.81	0.82	0.83
5	0.81	0.82	0.82
6	0.66	0.68	0.70
7	0.53	0.56	0.59
8	0.71	0.73	0.74
9	0.79	0.81	0.82
10	0.63	0.65	0.66
11	0.68	0.70	0.72
12	0.60	0.62	0.64
13	0.73	0.74	0.75

Figure 2

Best Feasible Balanced Incomplete Block Design for the National Assessment of Educational Progress (NAEP) Grade 8 Mathematics Project

Booklet	Blocks												
	1	2	3	4	5	6	7	8	9	10	11	12	13
1	x									x	x		
2						x		x			x		
3		x				x						x	
4							x		x	x			
5		x					x				x		
6				x		x	x						
7	x						x	x					
8	x					x			x				
9											x	x	x
10					x		x					x	
11			x						x		x		
12			x							x		x	
13	x	x	x										
14	x			x								x	
15						x				x			x
16			x				x						x
17								x	x			x	
18		x		x						x			
19			x		x	x							
20					x			x		x			
21		x			x				x				
22				x					x				x
23	x				x								x
24		x						x					x
25				x	x						x		
26			x	x				x					

Figure 3
Value of Objective Function for the First 1,000 Iterations



blocks to the booklets are given in Table 1. These probabilities ranged from .536 to .836, whereas their average deviation from the target value was .217.

As with the use of any heuristic, the question should be asked about how good the solution was. For the current problem, if the constraints on the assignment of pairs of blocks to the booklets are relaxed, a lower bound to the value of the objective function for the optimal solution is easily found by the well-known simplex algorithm in linear programming. The value of the lower bound was 16.81, which shows that the actual result of 16.93 for the solution must have been close to optimality.

Assigning the blocks in the worst possible way, an upper bound to the solution equal to 18.82 was found. In addition, the initial design in Figure 1 had a value of 17.13. All these values suggest that there was not much space for optimization in this case study. The same conclusion can be derived from the plot of the values for the first 1,000 iterations of the simulated annealing process in Figure 3, which reveals a general trend toward a very slow decrease in the value of the objective function. Observe also that the plot shows small spikes, which are the results of the probabilistic backtracking of the algorithm to avoid getting trapped in a local minimum.

The reason for the lack of space for optimization in the current case study was, again, the fact that the assignment problem was severely constrained by the number of booklets, blocks, and pairs that had to be dealt with. A relative increase in the number of blocks to the number of booklets would have turned the actual equality constraint on the assignment of pairs of blocks into an inequality,

which would have had a dramatic impact on the result. If the interest is in optimizing the designs of assessments, such measures should always be considered.

Concluding Remark

An important assumption in this article is that the item pool is already organized into blocks of items. Although this assumption is based on the current assessment practice, the use of blocks instead of individual items is already a severe constraint on the assembly of the assessment booklets. This point can easily be demonstrated for the objective function in the empirical example in this article. If the items in the blocks happen to vary considerably in difficulty, it will never be possible to assign the blocks to subpopulations such that the objective function yields a favorable low value relative to those for other feasible assignments. But even if the blocks are homogeneous in difficulty, some levels may be overrepresented or underrepresented, and again more favorable results are automatically given up.

In principle, it is possible to assign items directly to assessment booklets for subpopulations. The problem then boils down to an instance of multiple-test assembly (van der Linden & Adema, 1998), with special constraints to guarantee a balanced incomplete block structure among the set of forms. These constraints are direct generalizations from those in (5) through (8).

The reason that items in educational assessments are often preassembled into blocks is to neutralize possible differences in context effects of the items among students who receive different forms. On the other hand, assembling assessment booklets directly from the items in the pool is likely to result in designs that are better in terms of the objective function used in the assembly process. Whether or not preassembly of item blocks should be preferred ultimately depends on the way the trade-off between these two factors is weighted.

References

Aarts, E., & Lenstra, J. K. (1997). *Local search in combinatorial optimization*. Chicester, UK: John Wiley.

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

Gilks, W. R., Richardson, S., & Spiegelhalter, D. J. (1996). *Markov chain Monte Carlo in practice*. London: Chapman & Hall.

ILOG. (2002). *CPLEX 8.1 users' manual*. Mountain View, CA: Author.

Johnson, E. G. (1992). The design of the national assessment of educational progress. *Journal of Educational Measurement*, *29*, 95-110.

Lord, F. M. (1962). Estimating norms by item sampling. *Educational and Psychological Measurement*, *22*, 259-267.

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

Lord, F. M., & Novick, M. R. (1968). *Statistical theories of mental test scores*. Reading, MA: Addison-Wesley.

Metropolis, N., Rosenbluth, A. W., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, *6*, 1087-1992.

Mislevy, R. J., Beaton, A. E., Kaplan, B., & Sheehan, K. M. (1992). Estimating population characteristics from sparse matrix samples of item responses. *Journal of Educational Measurement*, *29*, 133-162.

Muraki, E. (1992). A generalized partial credit model: Application of an EM algorithm. *Applied Psychological Measurement*, *16*, 159-176.

Muraki, E. (1997). A generalized partial credit model. In W. J. van der Linden & R. K. Hambleton (Eds.), *Handbook of modern item response theory* (pp. 153-164). New York: Springer-Verlag.

Nemhauser, G. L., & Wolsey, L. A. (1999). *Integer and combinatorial optimization*. New York: John Wiley.

Reese, C. M., Miller, K. E., Mazzeo, J., & Dossey, J. A. (1997). *NAEP 1996 mathematics report card for the nation and the states*. Washington, DC: National Center for Education Statistics.

Rust, K. F., & Johnson, E. G. (1992). Sampling and weighting in the national assessment. *Journal of Educational Measurement*, *17*, 111-129.

Sirotnik, K. (1974). An introduction to matrix sampling for the practitioner. In W. J. Popham (Ed.), *Evaluation in education: Current applications* (pp. 453-399). Berkeley, CA: McCutchen.

van der Linden, W. J. (1996). Assembling test for the measurement of multiple traits. *Applied Psychological Measurement*, *20*, 373-388.

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

van der Linden, W. J. (2004). *Linear models for optimal test design*. New York: Springer-Verlag.

van der Linden, W. J., & Adema, J. J. (1998). Simultaneous assembly of multiple test forms. *Journal of Educational Measurement*, *35*, 185-198. (Addendum in *Journal of Educational Measurement*, *36*, 90-91)

van der Linden, W. J., Scrams, D. J., & Schnipke, D. L. (1999). Using response-time constraints to control for speededness in computerized adaptive testing. *Applied Psychological Measurement*, *23*, 195-210.

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

Wijnstra, J. M. (Ed.). (1988). *Balans van het rekenonderwijs in de basisschool: Uitkomsten van de eerste rekenpeiling medio en einde basis onderwijs* [Assessment of artihmetic in elementary education: Results from the first study in elementary education]. Arnhem, the Netherlands: Cito.

Winer, B. J. (1970). *Statistical principles in experimental design*. New York: McGraw-Hill.

Acknowledgments

The article was completed while the author was a fellow of the Center for Advanced Study in the Behavioral Sciences, Stanford, CA. He is indebted to the Spencer Foundation for a grant awarded to the Center to support his fellowship. The case study was initiated while the third author was at Educational Testing Service. The computational assistance of Wim M. M. Tielen for the example is gratefully acknowledged.

Author's Address

Address correspondence to W. J. van der Linden, Department of Research Methodology, Measurement and Data Analysis, University of Twente, P.O. Box 217, 7500 AE Enschede, the Netherlands; e-mail: w.j.vanderlinden@utwente.nl.