Chapter 8
Designing Item Pools for Computerized Adaptive Testing

Bernard P. Veldkamp & Wim J. van der Linden
University of Twente, The Netherlands

Introduction

In existing computerized adaptive testing (CAT) programs, each successive item in the test is chosen to optimize an objective function. Examples of well-known objectives in CAT are maximizing the information in the test at the ability estimate for the examinee and minimizing the deviation of the information in the test from a target value at this estimate. In addition, item selection is required to realize a set of content specifications for the test. For example, item content may be required to follow a certain taxonomy, or, if the items have a multiple-choice format, their answer key distribution should deviate not too much from uniformity. Content specifications are generally defined in terms of combinations of attributes the items in the test should have. They are typically realized by imposing a set of constraints on the item-selection process. The presence of an objective function and constraints in CAT leads to the notion of CAT as constrained (sequential) optimization. For a more formal introduction to this notion, see van der Linden (2000).

In addition to content constraints, item selection in CAT is often also constrained with respect to the exposure rates of the items in the pool. These constraints are necessary to maintain item pool security. Sympson and Hetter (1985) developed a probabilistic method for item-exposure control. In their method, after an item is selected, a probability experiment is run to determine whether the item is or is not administered. By manipulating the (conditional) probabilities in this experiment, the exposure rates of the items are kept below their bounds. Several modifications of this method have been developed (Davey

& Nering, 1998; Stocking & Lewis, 1998). For a review of these methods, see Stocking and Lewis (2000).

Though current methods of item-exposure control guarantee upper bounds on the exposure rates of the items, they do not imply any lower bounds on these rates. In fact, practical experience with CAT shows that item pools often have surprisingly large subsets of items that are seldom administered. The reason for this phenomenon is that such items contribute poorly to the objective function optimized in the CAT algorithm or have attribute values that are overrepresented in the pool relative to the requirements in the constraints on the test. Since item production usually involves a long and costly process of writing, reviewing, and pretesting the items, the presence of unused items in the pool forms an undesired waste of resources.

Though CAT algorithms could be developed to guarantee a lower bound on the exposure rates for the items in the pool as well, a more efficient approach to over- or underexposure of items is trying to prevent the problem at all and $design$ the item pool to produce a more uniform item usage for the population of examinees. It is the purpose of this chapter to propose a method of item pool design that addresses this target. The main product of the method is an optimal blueprint for the item pool, that is, a document specifying what attributes the items in the CAT pool should have and how many items of each type are needed.

The blueprint should be used as a starting point for the item writing process. As will be shown below, if "the identity of the item writer" is used as a categorical item attribute in the design process, the blueprint can also lead to an optimal division of labor among the item writers. However, since some quantitative item attributes, in particular those that depend on statistical parameters estimated from empirical data, are difficult to realize exactly, a realistic approach is to use the method proposed in this chapter as a tool for continuous management of the item writing process. Repeated applications of it can then help to adapt the next stage in the item writing process to the part of the pool that has already been written.

Review of Item Pool Design Literature

The subject of item pool design has been addressed earlier in the literature, both for pools for use with CAT and the assembly of linear test forms. A general description of the process of developing item pools for CAT is presented in Flaugher (1990). This author outlines several steps in the development of an item pool and discusses current practices at these steps. A common feature of the process described in Flaugher and the method in the present paper is the use of computer simulation. However, in Flaugher's outline, computer simulation is used to evaluate the performance of an item pool once the items have been written and field tested whereas in the current chapter computer simulation is used to design an optimal blueprint for the item pool.

Methods of item pool design for the assembly of linear test forms are presented in Boekkooi-Timminga (1991) and van der Linden, Veldkamp and Reese (2000). These methods, which are based on the technique of integer programming, can be used to optimize the design of item pools that have to support the assembly of a future series of test forms. The method in Boekkooi-Timminga follows a sequential approach calculating the numbers of items needed for these test forms maximizing their information functions. The method assumes an item pool calibrated under the one-parameter logistic (1PL) or Rasch model. The method in van der Linden, Veldkamp and Reese directly calculates a blueprint for the entire pool minimizing an estimate of the costs involved in actually writing the items. All other test specifications, including those related to the information functions of the test forms, are represented by constraints in the integer programming model that produces the blueprint. This method can be used for item pools calibrated under any current IRT model. As will become clear below, the current proposal shares some of its logic with the latter method. However, integer programming is not used for direct calculation of the numbers of items needed in the pool—only to simulate constrained CAT.

Both Swanson and Stocking (1998) and Way, Steffen and Anderson (1998; see also Way, 1998) address the problem of designing a system of rotating item pools for CAT. This system assumes the presence of a master pool from which operational item pools

are generated. A basic quantity is the number of operational pools each item should be included in (degree of item-pool overlap). By manipulating the number of pools items are included in their exposure rates can be controlled. A heuristic based on Swanson and Stocking's (1993) weighted deviation model (WDM) is used to assemble the operational pools from the master pool such that, simultaneously, the desired degree of overlap between the operational pools is realized and they are as similar as possible. The method proposed in this paper does not assume a system of rotating item pools. However, as will be shown later, it can easily be adapted to calculate a blueprint for such a system.

Designing a Blueprint for CAT Item Pools

The process of designing on optimal blueprint for a CAT item pool involves the following stages: First, the set of specifications for the CAT is analyzed and all item attributes figuring in the specifications are identified. The result of this stage is a (multivariate) classification table defined as the product of all categorical and quantitative item attributes. Second, using this table, an integer programming model for the assembly of the shadow tests in a series of CAT simulations is formulated. (The notion of CAT with shadow tests will be explained below.) Third, the population of examinees is identified and an estimate of its ability distribution is obtained. In principle, the distribution is unknown but an accurate estimate may be obtained, for example, from historical data. Fourth, the CAT simulations are carried out by sampling examinees randomly from the ability distribution. Counts of the number of times items from the cells in the classification table are administered in the simulations are cumulated. Fifth, the blueprint is calculated from these counts adjusting them to obtain optimal projections of the item exposure rates. Some of these stages are now explained in more detail.

Setting Up the Classification Table

The classification table for the item pool is set up distinguishing the following three kinds of constraints that can be imposed on the item selection by the CAT algorithm: (1)

constraints on categorical item attributes, (2) constraints on quantitative attributes, and (3) constraints needed to deal with inter-item dependencies (van der Linden, 1998).

Categorical item attributes, such as content, format, or item author, partition an item pool into a collection of subsets. If the items are coded by multiple categorical attributes, their Cartesian product induces a partitioning of the pool. A natural way to represent a partitioning based on categorical attributes is as a classification table. For example, let C1, C2, and C3 represent three levels of an item content attribute and let F1 and F2 represent two levels of a item format attribute. Table 1 shows the classification table for a partition that has six different cells, where n_{ij} represents the number of items in the pool that belong to cell (i, j).

Examples of possible quantitative item attributes in CAT are: word counts, values for the item difficulty parameters, and item response times. Classifications based on quantitative attributes are less straightforward to deal with. Some of them may have a continuous range of possible values. A possible way to overcome this obstacle is to pool adjacent values. For example, the difficulty parameter in the three parameter logistic IRT model takes real values in the interval $(-\infty, \infty)$. This interval could be partitioned into the collection of the following subintervals: $((-\infty, -2.5), (-2.5, -2), \ldots, (2, 2.5), (2.5, \infty))$. The larger the number of intervals, the more precise the approximation to the true item parameter values. After such partitioning, quantitative attributes can be used in setting up classification tables as if they were categorical. If single numbers are needed to represent intervals of attribute values, their midpoints are an obvious choice.

Table 1: Classification table

	F1	F2
C1	n_{11}	n_{21}
C2	n_{12}	n_{22}
C3	n_{13}	n_{23}

Inter-item dependencies deal with relations of exclusion and inclusion between the items in the pool. An example of an exclusion relation is the one between items in "enemy sets". Such items can not be included in the same test because they have clues to each

other's solution. However, if previous experience has shown that enemies tend to be items with certain common combinations of attributes, constraints can be included in the CAT algorithm to prevent such combinations from happening. The problem of CAT from item pools with exclusion relations between the items will be addressed later in this chapter. An example of an inclusion relation is the one between set-based items in a test, that is, sets of items organized around common stimuli. When designing pools for the assembly of linear test forms, relations between set-based items can be dealt with by setting up a separate classification table based on the stimulus attributes and then assigning stimuli to item sets. An example of this approach is given in van der Linden, Veldkamp and Reese (2000). In this chapter, the problem of CAT from pools with item sets is not addressed.

The result of this stage is thus a classification table, $C \times Q$, that is the Cartesian product of a table C based upon the categorical attributes and a table Q based upon the quantitative attributes. Each cell of the table represents a possible subset of items in the pool that have the same values for their categorical attributes and belong to the same (small) interval of values for their quantitative attributes.

Constrained CAT Simulations

To find out how many items an optimal pool should contain from each cell in table $C \times Q$, simulations of the CAT procedure are carried out. Each cell in the $C \times Q$ table is represented by a decision variable in the integer programming model for the shadow test. The variables represents the number of times items of each type are selected for the shadow test. The method of constrained CAT with shadow tests (van der Linden, 2000; van der Linden & Reese, 1998) is briefly explained and a general formulation of an integer programming model for selecting a shadow test is given.

Constrained Adaptive Testing with Shadow Tests

In constrained adaptive testing with shadow tests, at each step a full test ("shadow test") is assembled. The shadow test is assembled to have an optimal value for the objective function and is required to meet a set of constraints that represents all test specifications. The item actually to be administered is selected from this shadow test;

it is the item with the optimal contribution to the value of the objective function for the test. As a result of this procedure, each actual adaptive test eventually meets all constraints and has items selected with optimal values for the objective function.

The algorithm for constrained CAT with shadow tests can be summarized as follows:

Step 1: Choose an initial value of the examinee's ability parameter θ.
Step 2: Assemble the first shadow test such that all constraints are met and the objective function is optimized.
Step 3: Administer an item from the shadow test with optimal properties at the current θ estimate.
Step 4: Update the estimate of θ as well as all other parameters in the test assembly model.
Step 5: Assemble a new shadow test fixing the items already administered.
Step 6: Repeat Steps 3-5 until all n items have been administered.

In an application of this procedure in a real-life CAT program, the shadow tests are calculated using a 0-1 linear programming (LP) model for test assembly. The variables in the model represent for each individual item in the pool the decision to select or not select the item in the shadow test. However, in the CAT simulations in the current application, the more general technique of integer programming is used. The integer variables represent the number of items needed from each cell in the $C \times Q$ table for each simulated examinee.

Integer Programming Model

Let x_{cq} be the integer variable for cell (c, q) in table $C \times Q$. This variable determines how many items are to be selected from cell (c, q) for each simulated examinee. Further, let n be the length of the CAT, $\widehat{\theta}_{k-1}$ the estimate of θ_j after $k-1$ items have been administered, and S_{k-1} the set of cells with nonzero decision variable after $k-1$ items have been administered. Fisher information in the response on item i for an examinee with ability θ_j is denoted as $I_i(\theta_j)$. Finally, V_g denotes the set of cells representing the combination of attributes in categorical constraint $g = 1, \ldots, G$, V_h the set of cells representing the combination of attributes levels in quantitative constraint $h = 1, \ldots, H$, and V_e the set of cells in enemy set $e = 1, \ldots, E$.

The model has an objective function for the shadow tests that minimizes an estimate of the costs involved in writing the items in the pool. The information on the ability parameter at the ability estimate in the CAT is bounded from below by a target value, T. Generally, item writing costs can be presented as quantities k_{cq}, $(c, q) \in C \times Q$. In the empirical example below, k_{cq} is chosen to be the inverse of the numbers of items in cell (c, q) in a representative previous item pool, the idea being that types of items that are written more frequently involve less efforts and, therefore, are likely to be less costly. Several suggestions for alternative estimates of item writing costs are given in van der Linden, Veldkamp and Reese (2000). Also, if these costs are dependent on the item writer and it is known which authors wrote which items in the previous item pool, a convenient option is to adopt the identity of the item writer as a categorical item attribute in the $C \times Q$ table. The blueprint then automatically assigns numbers of items to be written to individual writers.

The general model for the assembly of the shadow test for the selection of the kth item in the CAT can be presented as:

$$\min \sum_{cq \in C \times Q} k_{cq} x_{cq} \qquad \text{(objective function)} \tag{1}$$

subject to

$$\sum_{cq \in C \times Q} I_{cq}\left(\widehat{\theta}_{k-1}\right) x_{cq} \geq T \qquad \text{, (information target)} \tag{2}$$

$$\sum_{cq \in S_{k-1}} x_{cq} = k - 1 \qquad \text{, (items already selected)} \tag{3}$$

$$\sum_{cq \in C \times Q} x_{cq} = n \qquad \text{, (test length)} \tag{4}$$

$$\sum_{cq \in Vg} x_{cq} = n_g, \qquad g = 1, ..., G \qquad \text{, (categorical constraint)} \tag{5}$$

$$f_h(x_{cq}) = n_h, \qquad h = 1, ..., H \qquad \text{, (quantitative constraint)} \tag{6}$$

$$\sum_{cq \in V_e} x_{cq} \leq 1, \qquad e = 1, ..., E \qquad \text{, (enemy sets)} \tag{7}$$

$$x_{cq} \in \{0, 1, 2, ...\}, \qquad (c, q) \in C \times Q \qquad \text{.(range of variables)} \tag{8}$$

The objective function in 1 minimizes the estimated item-writing costs. The constraint in 2 requires the information in the shadow test at the examinees' current ability estimate to meet the prespecified target value T. The constraint in 3 requires the $k-1$ previously administered items to be in the test. The attribute values of the previous items are automatically taken into account when selecting the kth item. In 4, the length of the CAT is fixed at n items. In 5 and 6, the categorical and quantitative constraints are imposed on the shadow test. The function f_h in 6 is assumed to be linear in the decision variables, for example, a (weighted) average or a sum. The constraints in 5 and 6 are formulated as equalities but can easily be turned into inequalities. The constraints in 7 allow the shadow test to have no more than one item from each enemy set.

Practical experience should guide the selection of the target value for the test information function. Generally, to emulate the process of a CAT that maximizes the information in the test, the target value should be chosen as high as possible without making the selection of the shadow tests infeasible. If the CAT program has been operational for some time, it should be known what targets are feasible. Alternatively, the operational CAT procedure can be based on the same target for the information function. The only thing needed to implement the latter option is to insert the constraint 2 into the model for the operational CAT. Other approaches are also possible. For instance, in the empirical example below, no target value for the information in the CAT was available, and the objective function in 1 and the information function in the constraint in 2 were combined into a linear combination optimized by the test assembly model. A review of options to deal with multi-objective decision problems in test assembly is given in Veldkamp (1999).

After the shadow test for the kth item is assembled, the item with maximum information at $\widehat{\theta}_{k-1}$ among the items not yet administered is selected for administration as the kth item in the CAT.

Multidimensionality

When items in the pool are calibrated with a multidimensional IRT model, the item selection model in 1 until 8 should be slightly altered. In 2 targets are set for the information function. In the multidimensional case, Fisher information is a matrix instead of a scalar, and since there is no one-to-one relationship between the elements of the matrix and measurement precision, no targets for these elements can be set. In the previous chapters several approaches have been suggested to deal with this problem.

In Chapter three, an approach based on the a-criterion and the d-criterion for optimizing matrices was described. When this approach is applied, targets can be set for these criteria. However, both criteria are nonlinear functions of the decision variables x_{cq}, and in the simulation study linear programming techniques are used, so targets have to be set for a linear approximation of the criteria. In Chapter three, it was shown that a linear approximation of the d-criterion performs better than an approximation of the a-criterion. Therefore, in the modified version of 2 targets can be set for the linear approximation of the d-criterion.

In Chapter five, a different approach is presented. Instead of Fisher information, Kullback-Leibler information is used. One of the advantages of Kullback-Leibler information is that it is a linear function of the decision variables x_{cq}, even when the items are calibrated with a multidimensional IRT model. Therefore, targets can be set for Kullback-Leibler information rather straightforwardly in the multidimensional case.

Both approaches can be used to do the simulation studies in the multidimensional case. However, about their use, the following remark should be made. When Kullback-Leibler information is used in the simulation study, the blueprint is developed for a CAT where Kullback-Leibler information is used instead of Fisher information. The specifications in the simulation study and in the practical CAT should be the same.

Calculating the Blueprint

The blueprint for the item pool is based on the counts of the number of times items from the cells in table $C \times Q$ are administered to the simulated examinees, N_{cq}. These

numbers are adapted to guarantee that the target values for the item-exposures rates are realized for a prespecified number of examinees sampled from the ability distribution. It is assumed that the number of examinees sampled, S, is large enough to produce stability among the relative values of N_{cq}.

The blueprint is calculated from the values of N_{cq} according to following formula:

$$I_{cq} = \left\lceil \frac{N_{cq}}{M} * \frac{C}{S} \right\rceil, \tag{9}$$

where I_{cq} is the number of items in cell (c, q) of the blueprint, M is the maximum number of times an item can be exposed before it is supposed to be known, S is the number of simulees in the CAT simulation, and C is the number of adaptive test administrations the item pool should support.

Application of this formula is justified by the following argument. If the ability distribution in the CAT simulations is a reasonable approximation to the true ability distribution in the population, N_{cq} predicts the number of items needed in cell (c, q). Because the numbers are calculated for S simulees and the item pool should support CAT for C examinees, N_{ij} has to be corrected by the factor $\frac{C}{S}$. This correction thus yields the numbers of items with attribute values corresponding to cell (c, q). However, to meet the required exposure rates, these numbers should be divided by M. The results from 9, rounded upwards to obtain integer values, define the optimal blueprint for the item pool.

Rotating Item Pools

The method can also be used to design a system of rotating item pools from a master pool. The general case is addressed in which overlap between rotating pools is allowed. Let G be the number of item pools the master pool has to support and n_{cq} the number of overlapping pools in which an item from cell (c, q) is allowed to occur. The number of items in a cell of the master pool is equal to:

$$\widetilde{I}_{cq} = \left\lceil \frac{N_{cq}}{M} * \frac{C}{S} * \frac{G}{n_{cq}} \right\rceil. \tag{10}$$

The number of items needed in every rotating item pool is I_{cq}. Because the master pool has to support G rotating item pools, I_{cq} has to be multiplied by G. Finally, since an item in cell (c, q) figures in n_{cq} pools, this number has to be divided by n_{cq}.

Empirical Example

As an empirical example, an item pool was designed for the CAT version of the Graduate Management Admission Test (GMAT). Five categorical item attributes were used which are labeled here as $C1, ..., C5$. Each attribute had between two and four possible values. The product of these attributes resulted in a table C with 96 cells.

All items were supposed to be calibrated by the three-parameter logistic (3PL) model:

$$P_i(\theta_j) \equiv c_i + (1 - c_i)\frac{e^{a_i(\theta_j - b_i)}}{1 + e^{a_i(\theta_j - b_i)}}, \tag{11}$$

where $P_i(\theta_j)$ is the probability that a person $j = 1 \ldots J$ with an ability parameter θ_j gives a correct response to an item $i = 1 \ldots I$, a_i is the value for the discrimination parameter, b_i for the difficulty parameter, and c_i for the guessing parameter of item i. The item parameters in this model were the quantitative attributes in this example. The range of values for the discrimination parameter, a_i, is the interval $[0, \infty)$. This interval was split into nine subintervals, the ninth interval extending to infinity. The difficulty parameter, b_i, takes values in the interval $(-\infty, \infty)$. Likewise, this interval was divided into fourteen subintervals. A previous item pool of 397 items from the GMAT was available. In this pool, the value of the guessing parameter, c_i, was approximately the same for all items. Therefore, in the simulation, c_i was fixed at this common value. The product of the quantitative attributes resulted in a table, Q, with 124 cells. The Cartesian product of the tables C and Q was a table with $96 \times 124 = 12,096$ cells.

As estimates of the item-writing costs, reciprocals of the frequencies of the items in a previous item pool were used (for cells with zero frequencies large number were chosen).

The actual specifications for the GMAT were used to formulate the integer programming model for the shadow tests in the CAT simulation. The model had 30 constraints dealing with such attributes as item content and test length. Because no target for the test information function was available, the following linear combination of test information and item writing costs was optimized:

$$\max\{\lambda \sum_{cq \in C \times Q} I_{cq}\left(\widehat{\theta}_{k-1}\right) x_{cq} - (1-\lambda) \sum_{cq \in C \times Q} k_{cq} x_{cq}\}. \quad \text{(objective function)} \tag{12}$$

The examinees were sampled from $N(1,1)$. The simulations were executed using software for constrained CAT with shadow tests developed at the University of Twente. The integer programming models for the shadow tests were calculated using calls to the linear-programming software package CPLEX 6.0 (ILOG, 1998). The initial estimate for each new simulee was set equal to $\widehat{\theta} = 1$. The estimate was updated using the method of EAP estimation with a uniform prior

The blueprint was calculated using realistic estimates for C and M in 9. For security reasons the blueprint can not be revealed here.

The simulation study was time intensive. For each examinee the complete test of 28 items took 8-9 minutes. The main reason for this is the large number of decision variables in the model, one for each of the 12,096 cells in the $C \times Q$ table. Large numbers of variables is not typical of real CAT, though. A previous simulation of constrained CAT with shadow tests directly from the previous GMAT pool had only 397 variables and took 2-3 seconds per item. Both the current and previous study were carried out on a Pentium 133 MHz computer.

Use of Item Pool Blueprint

The method presented in this chapter produces a blueprint for an item pool that serves as the best goal available to guide the item writing process. Its primary goals is to prepare the instructions for the item writers. If the identity of the writers is used as an attribute in the $C \times Q$ table for which cost estimates were obtained, the blueprint automatically assigns these instructions to them. In the item writing process, both the categorical item attributes as well as some of the quantitative attributes (e.g., word counts) can easily be realized. However, as already discussed, other quantitative item attributes, in particular those of a statistical nature, are more difficult to realize. If an existing item pool is used to estimate item writing costs, the blueprint for the item pool is automatically based on the empirical correlations between the statistical attributes and the other attributes. For example, if the difficult items tended to have other values for their categorical attributes than the easy items, the blueprint takes this fact automatically into account. This feature may improve the item-writing results but exact realization of statistical item attributes remains an optimistic goal.

The best way to implement the blueprint is, therefore, not in a one-shot approach but in a sequential fashion, recalculating the blueprint after a certain portion of the items has been written and field tested so that their actual attribute values are known. Repeated applications of the method helps to adapt the item writing efforts to the actual numbers of items already present in the pool. The same practice has been proposed for item pool design for assembling multiple linear test forms.(van der Linden, Veldkamp & Reese, 2000).

If the method is implemented sequentially, in each rerun of the CAT simulations the model for the shadow tests in 1-8 should be adapted to allow for the items already admitted to the item pool. The result is a mixed model, with 0-1 decision variables for the items already in the pool model and full integer variables for the new items in the cells of the $C \times Q$ table.

Let $i = 1, .., I$ be the index for the items already in the pool and x_i the variable for the decision to include ($x_i = 1$) or not to include item i in the shadow test ($x_i = 0$). For these

items, the actual attribute values should be included in the model. Also, the actual costs of the writing of item i, k_i, should be specified on the same scale as k_{cq}. The variables x_{cq} still represent the selection of items with attribute values associated with cell (c, q) in the $C \times Q$ table needed.

The adapted model is as follows:

$$\min \sum_{cq \in C \times Q} k_{cq} x_{cq} + \sum_{i=1}^{I} k_i x_i \qquad \text{(objective function)} \tag{13}$$

subject to

$$\sum_{cq \in C \times Q} I_{cq}\left(\widehat{\theta}_{k-1}\right) x_{cq} + \\ + \sum_{i=1}^{I} I_i\left(\widehat{\theta}_{k-1}\right) x_i \geq T \qquad \text{, (information target)} \tag{14}$$

$$\sum_{cq \in S_{k-1}} x_{cq} + \sum_{i=1}^{I} x_i = k - 1 \qquad \text{, (items selected)} \tag{15}$$

$$\sum_{cq \in C \times Q} x_{cq} + \sum_{i=1}^{I} x_i = n \qquad \text{, (test length)} \tag{16}$$

$$\sum_{cq \in Vg} x_{cq} + \sum_{i \in V_g} x_i = n_g,$$

$$g = 1, ..., G \qquad \text{, (categorical constraint)} \tag{17}$$

$$f_h(x_{cq}) + f_h(x_i) = n_h,$$

$$h = 1, ..., H \qquad \text{, (quantitative constraint)} \tag{18}$$

$$\sum_{cq \in V_e} x_{cq} + \sum_{i \in V_e} x_i \leq 1,$$

$$e = 1, ..., E \qquad \text{, (enemy sets)} \tag{19}$$

$$x_{cq} \in \{0, 1, 2, ...\},$$

$$(c, q) \in C \times Q \qquad \text{(range of variables)} \tag{20}$$

$$x_i \in \{0,1\},$$
$$i \in I \quad \text{.(range of variables)} \tag{21}$$

The proposed application is thus to run the model repeatedly during the item writing process. At each next application, the number of decision variables x_i grows whereas the the values of the variables x_{cq} decrease. If the complete item pool is realized and the stage of operational CAT is entered, the model for the shadow test contains only the variables x_i.

Concluding Remark

One of the reasons for proposing a optimal blueprint as a target for the item writing process is to create more even item exposure. However, the $C \times Q$ table in this chapter can also be used to realize this goal for a CAT pool that has not been developed using the proposed blueprint. Suppose the items in an existing pool are clustered in the cells (c, q) of table $C \times Q$. Then all items in the same cell have identical values for their categorical attributes and values for their quantitative attributes that differ only slightly. As a consequence, items in the same cell are approximately equally informative at the estimated θ values for the examinees that take the adaptive test. Nevertheless, the actual exposure rates of the items may vary considerably. Adaptive testing involves optimal item selection and therefore tends to capitalize on small differences between the items.

Why not overcome this capitalization by reformulating the CAT algorithm to select cells from the $C \times Q$ table instead of individual items and randomly select one of the actual items from the cells for administration? The result of this algorithm is even exposure of items in the same cell. A similar approach has been proposed by Holmes and Segall (1999). Differences between cells can further be leveled by applying a method for item-exposure control on the selection of the cells in the $C \times Q$ table rather than the individual items in the pool. In addition to the use of more rational methods of item pool design, continuous attempts to fine tune item selection criteria may be needed to produce CATs that combine accurate measurement with a more uniform item exposure.

Boekkooi-Timminga, E. (1991, June). *A method for designing Rasch model-based item banks*. Paper presented at the annual meeting of the Psychometric Society, Princeton, NJ.

Davey, T. & Nering, M. (1998, September). *Controlling item exposure and maintaining item security.* Paper presented at the ETS Computer-Based Testing Colloquium, Philadelphia, PA.

Flaugher, R. (1990). Item Pools. In H. Wainer, *Computerized adaptive testing: A primer* (pp. 41-64). Hillsdale, NJ: Lawrence Erlbaum Associates.

Holmes, R. M. , & Segall, D. O. (1999, April). *Reducing item exposure without reducing precision (much) in computerized adaptive testing.*

Paper presented at the annual meeting of the National Council on Measurement in Education, Montreal, Canada.

ILOG, Inc. (1998). *CPLEX 6.0* [Computer Program and Manual]. Incline Village, NV: ILOG.

Stocking, M. L. & Lewis, C. (1998). Controlling item exposure conditional on ability in computerized adaptive testing. *Journal of Educational and Behavioral Statistics*, 23, 57-75.

Stocking, M. L. & Swanson, L. (1998). Optimal design of item banks for computerized adaptive tests. *Applied Psychological Measurement,* 22, 271-280.

Swanson, L. & Stocking, M. L. (1993). A model and heuristic for solving very large item selection problems, *Applied Psychological Measurement,* 17, 151-166.

Sympson, J. B. & Hetter, R. D. (1985). Controlling item-exposure rates in computerized adaptive testing. *Proceedings of the 27th annual meeting of the Military Testing Association* (pp. 973-977). San Diego, CA: Navy Personnel Research and Development Center.

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

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

van der Linden, W. J., Veldkamp, B. P., and Reese, L. M. (2000). An integer programming approach to item pool design. *Applied Psychological Measurement,* 24. (In press).

Veldkamp, B. P. (1999). Multiple objective test assembly problems. *Journal of Educational Measurement,* 36. (In press)

Way, W. D. (1998). Protecting the integrity of computerized testing item pools. *Educational Measurement: Issues and Practice,* 17, 17-26.

Way, W. D., Steffen, M., & Anderson, G. S. (1998, September). *Developing, maintaining, and renewing the item inventory to support computer-based testing.* Paper presented at the ETS Computer-Based Testing Colloquium, Philadelphia, PA.