Measurement and Research Department Reports 2008-1

A Binary Programming Approach to Automated Test Assembly for Cognitive Diagnosis Models

Matthew Finkelman
Wonsuk Kim
Louis Roussos
Angela J. Verschoor

Measurement and Research Department Reports 2008-1

A Binary Programming Approach to Automated Test Assembly for Cognitive Diagnosis Models

Matthew Finkelman, Tufts University School of Dental Medicine, Boston, MA

Wonsuk Kim, Measured Progress, Dover, NH

Louis Roussos, Measured Progress, Dover, NH

Angela J. Verschoor, Cito

Cito
Arnhem, 2008

Abstract

Automated test assembly (ATA) has been an area of prolific psychometric research. While ATA methodology is well-developed for unidimensional models, its application alongside cognitive diagnosis models (CDMs) is a burgeoning topic. Two suggested procedures for combining ATA and CDMs are to maximize the cognitive diagnostic index (CDI) and to employ a genetic algorithm (GA). Each of these procedures has a disadvantage: CDI cannot control attribute-level information, and GA is computationally intensive. The goal of this article is to solve both problems by using binary programming, together with the item discrimination indexes of Henson et al., for performing ATA with CDMs. The three procedures are compared in simulation. Advantages and disadvantages of each are discussed.

Introduction

One of the most common problems in test design is the construction of a linear form from a pre-calibrated item pool. In particular, the *automated test assembly* (ATA) of a form without human intervention has been well-studied for unidimensional models; see Birnbaum (1968) and Lord (1980) for two examples of early work. Recently, the use of binary programming (BP; Theunissen, 1985; van der Linden, 2005; van der Linden & Boekkooi-Timminga, 1989) has been popularized in the ATA literature. BP methodology is advantageous because it allows a numerical objective function to be optimized subject to the practitioner's desired content constraints. It therefore produces a test that is suitable from both psychometric and content standpoints.

While BP is now considered a standard ATA solution for unidimensional models, it has yet to be proposed alongside cognitive diagnosis models (CDMs). This is because BP objective functions typically utilize the concept of Fisher information (see Table 3 of van der Linden and Boekkooi-Timminga, 1989), and Fisher information is undefined for CDMs (Henson & Douglas, 2005). As a result, researchers have devised alternative methods for combining ATA with CDMs. Henson and Douglas introduced the cognitive diagnostic index (CDI) for CDMs and recommended the selection of items with the largest CDI values; Finkelman, Kim and Roussos (in press) suggested using a genetic algorithm (GA) to find the items that optimize a given fitness function. Both of these methods were shown to exhibit vastly better accuracy than randomly generated forms; however, they each have a drawback. First, although CDMs are designed to measure multiple skills (often referred to as *attributes*), CDI does not provide the attribute-level information of each item (Henson & Douglas, 2005; Henson, et al., in press). Therefore, a test consisting of items with high CDI values may still produce poor measurement for some attributes. Second, the GA of Finkelman et al. involves simulation and a local search algorithm; thus, while it is able to control error rates at the attribute level, it requires high computational intensity.

The goal of this article is to develop a BP test assembly procedure that can be used alongside CDMs, thereby bridging the gap between the ATA methodologies of CDMs and unidimensional models. As will be seen, the particular objective function

employed is based on the attribute-level item discrimination indexes of Henson et al. (in press) and therefore provides adequate measurement of all attributes. Additionally, the procedure is much less computationally intensive than the GA approach. Hence, it avoids the problems of CDI and GA while offering the familiarity and power of BP.

We begin by providing a brief introduction to the concepts and notation of CDMs. We then review the current ATA methods for CDMs, namely CDI and GA, before explaining the BP framework and proposing our specific procedure for CDMs. This BP procedure is compared with CDI and GA in multiple simulation sets. We conclude by discussing the situations in which each method is appropriate.

Cognitive Diagnosis Models

In diagnostic testing, the goal is not to measure an examinee's overall ability in some area of scholastics, but rather to assess multiple attributes simultaneously so that strengths and weaknesses can be identified. CDMs were developed to facilitate such diagnostic inferences. In a CDM, each examinee's latent trait is formalized as a vector $\boldsymbol{\alpha} = (\alpha_1,...,\alpha_K)$ of K variables; α_k indicates the examinee's true ability along attribute k. Like most CDM research, we assume only two ability levels for each attribute, so that $\alpha_k = 1$ indicates mastery of attribute k and $\alpha_k = 0$ indicates non-mastery of this attribute. Only certain attributes are measured by each item; information relating items to attributes is typically given by the Q-matrix (Tatsuoka, 1985). Letting $j = 1,...,J$ index the items in a given pool, the $[j,k]$ entry of the Q-matrix (hereafter denoted q_{jk}) is equal to 1 if item j measures attribute k, and 0 otherwise.

Once the Q-matrix has been specified and items have been administered in a field test, the items are calibrated to a CDM of choice. As cited by Finkelman et al. (in press), available CDMs include the Restricted Latent Class Model (Haertel, 1984, 1990) or DINA model (Junker & Sijtsma, 2001), NIDA (Junker & Sijtsma, 2001), and compensatory MCLCM (Maris, 1999; von Davier, 2005). Finkelman et al. used the

"reduced" version of the Reparameterized Unified Model (RUM; DiBello, Stout, & Roussos, 1995; Roussos et al., 2007) in their simulations; to allow comparison with this previous study, we will also focus on the reduced RUM in this article.

The reduced RUM assumes that all items are dichotomously scored as correct or incorrect. Intuitively, its idea is that in order to answer an item correctly, each attribute measured by the item must be successfully applied. For item j, let π_j^* represent the probability of successfully applying all such attributes by an examinee who has mastered each one. It is natural that this probability should be at least as high as that of an examinee who has not mastered some required attributes. To quantify the decrement in probability associated with non-mastery, let π_{jk} denote the probability that a master of attribute k would successfully apply this attribute to item j, and let r_{jk} denote the analogous probability for a non-master of attribute k. Because masters are assumed to have greater acumen than non-masters, we require $\pi_{jk} \geq r_{jk}$; equivalently, we require the ratio $r_{jk}^* \equiv \frac{r_{jk}}{\pi_{jk}}$ to be less than or equal to one. Under the reduced RUM, the probability of a correct response to item j, given a true ability vector of $\boldsymbol{\alpha}$, is

$$P_j(\boldsymbol{\alpha}) = \pi_j^* \prod_{k=1}^{K} r_{jk}^{*\,(1-\alpha_k)q_{jk}} \tag{1}$$

(Roussos et al., 2007). From this formula, we see that if $\alpha_k = 1$ for all k being measured (that is, for all k such that $q_{jk} = 1$), then $P_j(\boldsymbol{\alpha}) = \pi_j^*$, as prescribed above. Furthermore, if item j measures attribute k, then a non-mastery status along attribute k reduces the probability of a correct response by a factor of r_{jk}^*. We note that in the original RUM, the right-hand side of Equation 1 is multiplied by an additional term related to the examinee's "supplemental ability" that may affect performance on the item, but is not part of the Q-matrix. The reduced RUM's omission of this term simplifies the model by assuming that such supplemental ability does not exist, or that it is always applied successfully. See Roussos et al. for further information.

In all that follows, we assume that a particular CDM has been chosen by the practitioner, the Q-matrix has been fixed, and item parameter estimates have been obtained. The task at hand is then to select which items from the pool will actually appear

on the test. We emphasize that as in Finkelman et al. (in press), the reduced RUM is used only as an example; the CDI, GA, and BP methods for ATA can be utilized alongside any of the CDMs listed above.

Previous ATA Methods for CDMs

CDI

In the formulation above, each of the K attributes has two possible states: mastery and non-mastery. Thus, examinee abilities can be classified in 2^K different ways. This discretization of the ability space is a departure from unidimensional IRT models like the 3-parameter logistic model (Birnbaum, 1968), where ability is defined along a continuous spectrum. As alluded to in the Introduction, such discretization precludes the use of Fisher information, which is the traditional psychometric tool utilized in ATA (Henson & Douglas, 2005). To overcome this problem, Henson and Douglas developed an ATA procedure based on Kullback-Leibler information (Chang & Ying, 1996; Cover & Thomas, 1991; Veldkamp & van der Linden, 2002). An item's Kullback-Leibler information between two candidate ability vectors, say $\boldsymbol{\alpha}'$ and $\boldsymbol{\alpha}''$, is defined as the expected log likelihood ratio of these vectors, assuming $\boldsymbol{\alpha}'$ is the true state of nature. It can be thus be thought of as the item's power to discern between $\boldsymbol{\alpha}'$ and $\boldsymbol{\alpha}''$ for examinees with an attribute vector of $\boldsymbol{\alpha}'$. For dichotomous items, the Kullback-Leibler information of item j between $\boldsymbol{\alpha}'$ and $\boldsymbol{\alpha}''$ can be expressed as

$$K_j(\boldsymbol{\alpha}',\boldsymbol{\alpha}'') = P_j(\boldsymbol{\alpha}')\log\left[\frac{P_j(\boldsymbol{\alpha}')}{P_j(\boldsymbol{\alpha}'')}\right] + (1-P_j(\boldsymbol{\alpha}'))\log\left[\frac{1-P_j(\boldsymbol{\alpha}')}{1-P_j(\boldsymbol{\alpha}'')}\right] \qquad (2)$$

(Henson & Douglas, 2005).

It is important to note that Equation 2 only measures item j's discriminatory power with respect to a pair $(\boldsymbol{\alpha}',\boldsymbol{\alpha}'')$ of candidate attribute patterns. To assess the item's overall discriminatory power, it is prudent to combine the information from each pair into a single index. Henson and Douglas (2005) created such an index by computing a weighted average of all possible pairs' Kullback-Leibler information values, including

both $K_j(\alpha',\alpha'')$ and $K_j(\alpha'',\alpha')$ since these terms may be different from one another. Reasoning that it is most difficult to discern α' from α'' when these patterns have many equal elements, Henson and Douglas gave higher relative weights to such pairs. In particular, they first quantified the distance of two patterns by their Hamming distance (Hamming, 1950), which is equal to the number of elements where they differ. Henson and Douglas then defined the weighting function as the inverse of this distance. Using the notation of Finkelman et al. (in press), the resulting weight of the pair (α',α'') can be expressed as

$$\xi(\alpha,\alpha') = \frac{1}{\sum_{k=1}^{K} |(\alpha_k' - \alpha_k'')|}. \quad (3)$$

The average of the $K_j(\alpha',\alpha'')$ values, thus weighted, is the CDI of item j:

$$CDI_j = \frac{\sum_{\alpha' \neq \alpha''} \xi(\alpha',\alpha'') K_j(\alpha',\alpha'')}{\sum_{\alpha' \neq \alpha''} \xi(\alpha',\alpha'')} \quad (4)$$

(Henson & Douglas, 2005).

To perform ATA, Henson and Douglas proposed the selection of items with the highest CDI values. For situations where constraints on content have been specified, they suggested the following iterative algorithm. At each iteration, every item is checked to ascertain whether its inclusion would allow the ultimate satisfaction of all constraints. Among those for which constraint satisfaction is possible, the one with the highest CDI value is added. This heuristic method finds a solution with large CDI values and without violating any constraints, assuming that the satisfaction of all constraints is possible.

The CDI's summary of each item by a single value is convenient for practitioners who seek an overall index of an item's information. However, as explained in the Introduction, this reduction to a single value comes at a cost: it does not allow an attribute-level analysis of each item's discriminatory power. Moreover, even if the test assembly procedure is constrained to measure each attribute a certain number of times, the use of CDI in ATA may result in differential accuracy across attributes (Finkelman et al., in press).

GA

Genetic algorithms were first popularized by Holland (1968, 1973, 1975) as a way to solve or approximate the solution to a difficult optimization problem. They have appeared in several psychometric applications (van der Linden, 2005; Verschoor, 2007; Zhang & Stout, 1999) and were used to conduct ATA alongside CDMs by Finkelman et al. (in press). While there are many types of GAs (Verschoor, 2007), we only describe the specific version suggested by Finkelman et al.

The GA begins by defining a *fitness function* that quantifies the performance of a solution (in ATA, a solution refers to a candidate set of items). For CDMs, where the goal is classification, it is natural to define the fitness of a solution in terms of its error rates. A solution's error rate with respect to attribute k is the probability of misclassifying that attribute (i.e., classifying the examinee as a master when the true classification is non-mastery, or vice versa), appropriately averaged over a Bayesian prior distribution on $\boldsymbol{\alpha}$. Letting $\pi(\boldsymbol{\alpha})$ denote the prior distribution on $\boldsymbol{\alpha}$, $\hat{\alpha}_k$ the observed classification of attribute k based on a specified ability estimator, and $E_{\boldsymbol{\alpha}}(X)$ the expected value of a random variable X under $\boldsymbol{\alpha}$, the error rate for attribute k is

$$e_k \equiv \sum_{\boldsymbol{\alpha}} \pi(\boldsymbol{\alpha}) E_{\boldsymbol{\alpha}}(|\alpha_k - \hat{\alpha}_k|) \tag{5}$$

(Finkelman et al., in press). Three fitness functions were proposed based on the values $(e_1,...,e_K)$. They were 1) the sum of the error rates across all attributes; 2) the maximum error rate across all attributes; and 3) the absolute distance of each error rate to a target error rate, summed across all attributes. For this last option, a set of target error rates, $(\varepsilon_1,...,\varepsilon_K)$, is determined in advance. Note that only one of the three fitness functions should be chosen, with lower values considered better.

In general, it is not possible to compute the exact error rates of a solution because they are complicated functions of both item parameters and prior probabilities. As a result, Finkelman et al. (in press) proposed that they be estimated through a "training set" of preliminary simulations. First, "true" attribute patterns of B simulees are drawn proportional to the prior distribution on $\boldsymbol{\alpha}$. Then each simulee is administered every item in the pool. From the resulting simulation set, it is possible to estimate any given

solution's error rate along each attribute k. This is done by obtaining each simulee's observed classification $\hat{\alpha}_k$ based on only the items of that solution (ignoring all other items), then computing the proportion of simulees for whom $\alpha_k \neq \hat{\alpha}_k$. Letting $\bar{e}_k$ denote this observed error rate along attribute k, the three aforementioned fitness functions can be calculated as $\sum_{k=1}^{K} \bar{e}_k$, $\max_{k=1}^{K} \bar{e}_k$, and $\sum_{k=1}^{K} |\bar{e}_k - \varepsilon_k|$, respectively. To reduce variability, the observed error rates may be replaced by their expectations; see Finkelmanet al. for details.

The above method allows the estimation of any solution's fitness, using a single set of B simulees. The goal of the GA is to find and select the solution with the best (lowest) such estimated fitness from the simulations, among the set of solutions satisfying each content constraint. Because there are generally too many candidate solutions to analyze them all, an iterative computer search for the optimal solution is utilized instead. This search begins with S initial solutions that satisfy each constraint; Finkelman et al. (in press) used $S = 3$ in their application. From these initial "parent" solutions, more candidate solutions (called "children") satisfying each constraint are created in a specified manner (described in the next paragraph). The fitness of every parent and child is computed, and the best S solutions are retained. These become the parents of the next iteration and give rise to children of their own. The process continues until a convergence criterion or a pre-specified number of iterations has been reached. Once the computer search has ended, the solution in the system with the best fitness is chosen as the "official" form of the GA.

More precisely, let N denote the desired number of items to be selected for the form, out of $J > N$ items in the pool. The S initial solutions, all of which contain N items, may be selected at random from the set of solutions satisfying each constraint, or they can come from analytic methods like the CDI. From these initial solutions, children are created by the process of *mutation*. Let $(j_{s1}, ..., j_{sN})$ denote the indexes of the items in initial parent s, where $s = 1, 2, ..., S$. In the mutation scheme of Finkelman et al. (in press), each child is identical to its parent except for one index. The first child of parent s is created by removing j_{s1} from the parent and replacing it with a new item, j'_{s1}. Here the

new item is randomly selected from the set of all items that allow the resulting child to be feasible, i.e., from the set whose addition to the vector $(j_{s2},..., j_{sN})$ creates a child that satisfies each constraint. Similarly, the second child removes j_{s2} and replaces it with a second item, j'_{s2}, where j'_{s2} is randomly chosen from the set of items whose addition to $(j_{s1}, j_{s3},..., j_{sN})$ allows feasibility. The resulting child is $(j_{s1}, j'_{s2}, j_{s3},..., j_{sN})$. Other children are created analogously, with each of the N parent items replaced in exactly one child. In this way, every parent spawns N children; since the parents themselves are also eligible for selection, there are $S(N+1)$ solutions to choose from at every iteration. As explained previously, solutions are then compared based on their estimated fitness values from the simulations; the best S are retained and become parents at the next iteration. Finkelman et al. proposed continuing the computer search until either (a) the best solution remains the same for 50 iterations or (b) 500 iterations are run. Once one of these conditions is invoked, the GA ceases and the best solution is selected. Note that since all children are required to satisfy each constraint, the GA's official form always satisfies each constraint as well.

By defining the fitness function to be the maximum attribute-level error rate, or by setting equal target error rates across all attributes, the GA solves the CDI's problem of unbalanced attribute-level accuracies. However, it requires more computational complexity. After all, to implement the GA, simulations and a computer search must both be performed. Finkelman et al.'s (in press) GA described above was specifically chosen for its relative simplicity; nevertheless, its running time may be burdensome for some applications.

A New Binary Programming Method

Introduction to Binary Programming

Under a general BP framework, the goal is to choose elements that optimize a specified numerical objective function, subject to various constraints on those elements. In the context of assessment, the elements are items and many of the constraints are on

content. The objective function may be interpreted analogously to the GA's fitness function.

Mathematically, the BP solution to ATA can be expressed as follows. Let x_j, $j = 1,...,J$, denote a dummy variable such that $x_j = 1$ if item j is selected for the test, and $x_j = 0$ otherwise. Let y be the objective function of interest; while y depends on the items selected (i.e., the x_j), this dependence is suppressed in the notation for greater simplicity. Next, we assume that the constraints are defined in terms of *characteristics* C_i, that every item can be dichotomously classified as possessing a characteristic ($C_{ij} = 1$) or not possessing it ($C_{ij} = 0$), and that lower and upper bounds L_i and U_i have been set for the number of selected items possessing characteristic $i = 1,...,I$. Then the task is to maximize y subject to the constraints

$$L_i \leq \sum_{\{x_j=1\}} C_{ij} \leq U_i, \quad i = 1,...,I. \tag{6}$$

Examples of constraints are:

1. The appropriate number of items measuring each content area. Consider a mathematics test, and suppose that each item assesses at least one of the following content areas: algebra, geometry, probability and statistics, trigonometry, and number sense. We would like to require that between 10 and 15 items measure algebra. In this case, let characteristic 1 denote the assessment of this content area: $C_{1j} = 1$ if and only if item j measures algebra. Then the inequality is written

$$10 \leq \sum_{\{x_j=1\}} C_{1j} \leq 15. \tag{7}$$

   Constraints for the other content areas are defined analogously. If only a lower bound is desired (i.e., if we seek at least 10 items measuring algebra), then U_1 is simply set to N rather than 15.
2. The appropriate balance of items across the answer key. Consider a multiple-choice test where the correct answer for each item is coded A, B, C, or D. To avoid confusion among examinees, we seek to ensure that the different answer choices are represented in approximately equal numbers. Let $C_{2j} = 1$ if and only if the answer for

item j is A, and suppose that we seek between 8 and 12 items with this answer choice. The inequality becomes

$$8 \leq \sum_{\{xj=1\}} C_{2j} \leq 12. \tag{8}$$

Constraints for other answer choices are defined analogously.

3. Enemy items. Suppose that items 82 and 143 cannot both be selected for administration, as one of these items gives a hint about the answer to the other. Let $C_{3j} = 1$ for $j \in \{82,143\}$ and $C_{3j} = 0$ for all other j. The constraint is written as

$$0 \leq \sum_{\{xj=1\}} C_{3j} \leq 1. \tag{9}$$

4. Test length. Let $C_{4j} = 1$ for all items in the pool. Consistent with the desire for N items to be selected, we have

$$N \leq \sum_{\{xj=1\}} C_{4j} \leq N. \tag{10}$$

As stated previously, the objective function y is usually related to Fisher information when BP is used alongside unidimensional models. Since Fisher information does not exist for CDMs, we must utilize a different objective function in the current application. The next subsection is devoted to developing our objective function, which is based on the attribute-level discrimination indexes of Henson et al. (in press).

An Objective Function for CDMs

Motivation

Ideally, the objective function would involve the attribute-level error rates themselves, instructing either that these rates be minimized or that they be as close as possible to target values. The BP paradigm cannot handle such an objective function, however, since the relation between error rates, item parameters, and the prior distribution is too complicated to be used directly (Finkelman et al., in press). Instead, we will utilize the heuristic indexes of Henson et al. (in press), which were designed to measure the information of each attribute and thus may be used as a proxy for their error rates.

Henson, et al. (in press) proposed three such indexes for CDMs; in increasing order of complexity, they will be denoted δ_j^A, δ_j^B, and δ_j^C. As will be seen, our definition of each index results in a vector of K values (one for each attribute), e.g., $\delta_j^A = (\delta_{j1}^A, ..., \delta_{jK}^A)$. The elements δ_{jk}^A, δ_{jk}^B, and δ_{jk}^C are different measures of item j's discriminatory power along attribute k. Thus, unlike an overall measure of discrimination like the CDI, the indexes of Henson et al. allow an attribute-level analysis of each item.

The purpose of this section is to develop an objective function that can be employed alongside any of the three indexes listed above. Only one index should be chosen for a given application; for illustration, we demonstrate our objective function using δ_j^B. The particular index δ_j^B was preferred to δ_j^A because the former takes into account the fact that certain attribute patterns may be more common than others in a population, while the latter does not consider such prior probabilities. δ_j^B was chosen rather than δ_j^C to ensure that each attribute is sufficiently measured through items requiring that particular attribute, rather than through correlational information as included by δ_j^C; see Henson, et al. (in press) for details. We emphasize that while δ_j^B was used in this study, the objective function introduced here is general: it is equally applicable to δ_j^A and δ_j^C by simply substituting either for δ_j^B.

The δ_j^B Index

Before proposing the objective function based on δ_j^B, it is necessary to define the index itself. The logic of using δ_j^B is as follows: to evaluate how much information is provided for attribute k, we restrict attention to those patterns $\boldsymbol{\alpha}'$ and $\boldsymbol{\alpha}''$ that only differ on that one particular attribute (that is, where $\boldsymbol{\alpha}'$ and $\boldsymbol{\alpha}''$ are identical for all K attributes except k). The amount of information between such $\boldsymbol{\alpha}'$ and $\boldsymbol{\alpha}''$ is as usual quantified via the Kullback-Leibler information (Equation 2). The Kullback-Leibler values are then combined into summary statistics, which will be made explicit below.

Formally, let Ω_{1k} denote the set of pairs $(\boldsymbol{\alpha}', \boldsymbol{\alpha}'')$ such that $\boldsymbol{\alpha}'$ and $\boldsymbol{\alpha}''$ are identical for every attribute except k, $\boldsymbol{\alpha}'$ indicates mastery on attribute k, and $\boldsymbol{\alpha}''$ does not. That is,

$$(\boldsymbol{\alpha}', \boldsymbol{\alpha}'') \in \Omega_{1k} \text{ if } \alpha_k' = 1,\ \alpha_k'' = 0, \text{ and } \alpha_v' = \alpha_v'' \ \forall v \neq k \tag{11}$$

(Henson, et al., in press). Similarly, Ω_{0k} is defined as the set of pairs $(\boldsymbol{\alpha}', \boldsymbol{\alpha}'')$ such that $\boldsymbol{\alpha}'$ and $\boldsymbol{\alpha}''$ are identical for every attribute except k, $\boldsymbol{\alpha}''$ indicates mastery on attribute k, and $\boldsymbol{\alpha}'$ does not:

$$(\boldsymbol{\alpha}', \boldsymbol{\alpha}'') \in \Omega_{0k} \text{ if } \alpha_k' = 0,\ \alpha_k'' = 1, \text{ and } \alpha_v' = \alpha_v'' \ \forall v \neq k. \tag{12}$$

Henson, et al. actually proposed two indexes for each attribute, one for Ω_{1k} and the other for Ω_{0k}. These indexes are linear combinations of the corresponding Kullback-Leibler information numbers, with weights proportional to the Bayesian prior probability that $\boldsymbol{\alpha}'$ is the true state of nature. For item j and attribute k,

$$\delta_{jk}^{B}(1) = \sum_{(\boldsymbol{\alpha}', \boldsymbol{\alpha}'') \in \Omega_{1k}} w_1(\boldsymbol{\alpha}') K_j(\boldsymbol{\alpha}', \boldsymbol{\alpha}'') \tag{13}$$

and

$$\delta_{jk}^{B}(0) = \sum_{(\boldsymbol{\alpha}', \boldsymbol{\alpha}'') \in \Omega_{0k}} w_0(\boldsymbol{\alpha}') K_j(\boldsymbol{\alpha}', \boldsymbol{\alpha}'') \tag{14}$$

where $w_1(\boldsymbol{\alpha}') = P(\boldsymbol{\alpha}' \mid \alpha_k = 1)$ and $w_0(\boldsymbol{\alpha}') = P(\boldsymbol{\alpha}' \mid \alpha_k = 0)$.

Equations 13 and 14 break down the attribute-specific information into two parts. The first part, $\delta_{jk}^{B}(1)$, is a measure of the item's discrimination along attribute k, assuming that the true classification of attribute k is mastery. Similarly, $\delta_{jk}^{B}(0)$ measures the item's discrimination along attribute k, assuming that the true classification is non-mastery. These indexes were kept separate by Henson, et al. (in press) because Kullback-Leibler information is asymmetric: it is not necessarily the case that $K_j(\boldsymbol{\alpha}', \boldsymbol{\alpha}'') = K_j(\boldsymbol{\alpha}'', \boldsymbol{\alpha}')$, nor is it always the case that $\delta_{jk}^{B}(1) = \delta_{jk}^{B}(0)$. However, these indexes are typically expected to exhibit significant positive correlation: after all, if an item can discern masters from non-masters along attribute k, it can generally do so whether mastery or non-mastery is assumed. Hence, to create an overall index of item

discrimination for attribute k, we take the average of Equations 13 and 14:

$$\delta_{jk}^{B} = \frac{\delta_{jk}^{B}(1) + \delta_{jk}^{B}(0)}{2}. \tag{15}$$

As claimed, the use of Equation 15 results in an index that is a vector of K values, $\delta_j^B = (\delta_{j1}^B, ..., \delta_{jK}^B)$.

While δ_j^B is a measure of item j's discriminatory power along attribute k, the ATA paradigm is concerned with the total discrimination of all items in the test. One convenient property of δ_j^A, δ_j^B, and δ_j^C is that they are additive (Henson et al., in press). Thus, to obtain the test's overall discrimination along attribute k, it suffices to sum the elements of the individual items. For example, when using δ_j^B as an index, the total discrimination along attribute k is given by $\delta_{tk}^B = \sum_{\{xj=1\}} \delta_{jk}^B$.

The Proposed Objective Function

While the focus of Henson et al. (in press) was not ATA, they did state that their three discrimination indexes could be used to aid test construction. In particular, they suggested the selection of items such that the resulting test has high discrimination for all attributes; however, they did not propose a specific method to accomplish that goal. This subsection formalizes their suggestion by introducing the use of δ_j^A, δ_j^B, or δ_j^C as part of a BP objective function, thus optimizing with respect to these indexes.

We observe that the above description can be thought of as a *maximin* problem: to ensure that all attributes are measured adequately, we seek to maximize the minimum attribute-level discrimination. This maximin approach has been utilized by van der Linden and Boekkooi-Timminga (1989) in the unidimensional setting, where Fisher information was the quantity of interest. In the application to CDMs, we use a similar procedure, with the discrimination indexes of Henson et al. (in press) substituted for

Fisher information. Again using δ_j^B as an example, the objective function is

$$y = \min_{k=1}^{K} \delta_{ik}^B. \quad (16)$$

The BP solution is achieved by maximizing Equation 16 subject to all constraints. The optimization can be performed using existing software such as CPLEX.

Simulation Studies

Method

Conditions

A previous study (Finkelman et al., in press) compared CDI and GA under eight simulation conditions. To promote comparability with this study, our design (comparing CDI, GA, and BP) was very similar to theirs. In particular, we used the same two item pools and prior distributions, while the imposed constraints were slightly different; details are presented in this section.

Each pool contained 300 simulated items following the reduced RUM model. The same Q-matrix was common to both pools; this Q-matrix defined a total of five attributes. 80 items measured one of the five attributes, 140 measured two attributes, and 80 measured three attributes. To study the effect of item information on the methods' classification properties, one pool was constructed of items with relatively high Kullback-Leibler information values, and the other contained items with relatively low such values. In the former pool, r_{ik}^* parameters were simulated from the uniform [0.40, 0.85] distribution; in the latter pool, they were simulated from the uniform [0.65, 0.92] distribution (note that lower r_{ik}^* values yield higher information). π_i^* values were simulated from the uniform [0.75, 0.95] distribution in each of the two pools. In every condition, the task of each ATA method was to select 40 items out of the 300.

Two prior distributions were used in the generation of simulees' latent abilities. The first was to simply generate an equal number of simulees for each of the 32 possible $\boldsymbol{\alpha}$ vectors, i.e., to take a discrete uniform distribution on $\boldsymbol{\alpha}$ with $P(\boldsymbol{\alpha}) = 1/32$. This

specification implicitly dictates that the probability of mastery is 50% for each attribute. The second prior distribution assumed that abilities come from an underlying continuous distribution and are discretized through cut points. As in Finkelman et al. (in press) and Henson and Douglas (2005), latent abilities were first generated from the multivariate standard normal distribution, with a tetrachoric correlation of 0.5 for each pair of attributes. α values were then created by dichotomizing each attribute into "master" or "non-master" categories depending on whether the continuous variable exceeded specified cut points. As in Finkelman et al., the cut points were defined so that the proportions of mastery in the population were 0.45, 0.50, 0.55, 0.60, and 0.65 for the five attributes.

All methods were examined both under conditions of no constraints and conditions where constraints were applied. In the latter conditions, the two types of constraints were (a) adequate representation of each attribute and (b) adequate answer key balance. Specifically, a solution was only feasible if it measured each attribute at least 20 times and had between eight and 12 items (inclusive) with each answer choice (A, B, C, and D). The requirement that each attribute be measured at least 20 times was different from that of Finkelman et al. (in press), who only required the inclusion of at least 15 items per attribute. This change was made because previous work had found little difference between the unconstrained solutions and solutions constrained to measure each attribute at least 15 times.

Summarizing the above, eight conditions were considered:

- Constraints, uniform prior, high-discriminating item pool;
- Constraints, uniform prior, low-discriminating item pool;
- Constraints, correlated prior, high-discriminating item pool;
- Constraints, correlated prior, low-discriminating item pool;
- No constraints, uniform prior, high-discriminating item pool;
- No constraints, uniform prior, low-discriminating item pool;
- No constraints, correlated prior, high-discriminating item pool;
- No constraints, correlated prior, low-discriminating item pool.

Outcome Measures and $\boldsymbol{\alpha}$ Estimates

The three different methods were compared with respect to their overall accuracy and balance of accuracy across attributes. Each outcome measure was calculated on the basis of 20,000 "test set" simulees; the number of simulees with each $\boldsymbol{\alpha}$ vector was proportional to the prior distribution. Overall accuracy was defined as the average number of classification errors made over the 20,000 simulees. Balance was quantified in two ways: the maximum error rate over the five attributes, and the range of error rates over the five attributes. For all three outcome measures, smaller values corresponded to better performance.

In order to determine how many errors were made for a given simulee, an estimate of that simulee's $\boldsymbol{\alpha}$ vector was required. We used the estimate $\hat{\boldsymbol{\alpha}}$ that had been employed by Finkelman et al. (in press); this estimate is defined as the $\boldsymbol{\alpha}$ vector minimizing the posterior expected number of classification errors, given the prior distribution and the observed data. Because each ATA method selects different items, there is a different estimate $\hat{\boldsymbol{\alpha}}$ for each method. We note that although the use of the correct prior distribution is favorable to methods incorporating prior information (BP and GA), a fair comparison can be made in conditions with a uniform prior (Finkelman et al., in press). Robustness to a misspecified prior will be considered in future work.

Operationalization of Each ATA Method

The BP solution was obtained by maximizing Equation 16 subject to the constraints, when specified. The program CPLEX 11.0 was used in the optimization. The determination of the CDI solution was trivial under the "no constraints" conditions: the solution simply chose the 40 items that exhibited the highest CDI values. However, Henson and Douglas' (2005) iterative item selection approach (at each step, checking whether each item would allow satisfaction of all constraints, then choosing the one with highest CDI) was difficult to apply under the constrained conditions. Therefore, when constraints were imposed, CPLEX 11.0 was again used to select the CDI solution, with the objective function defined as the items' summed CDI values.

The selection of GA items was more complicated than that of either CDI or BP. As described previously, GA requires a preliminary training set of 20,000 simulees whose responses are used in a local search algorithm to find the optimal item set. For each condition, the three initial parents to the GA were the CDI solution, the BP solution alongside the δ_j^B index of Henson et al. (in press), and the BP solution alongside the δ_j^A index of Henson et al. The δ_j^A index is identical to the δ_j^B index when a uniform prior distribution is imposed; see Henson et al. for details about δ_j^A. A FORTRAN 6.1.0 program was used to take the above three initial parents as inputs, perform the mutation process of the GA, and return the resulting optimal solution. Because one of the advantages of GA is that it can be tailored to the desired fitness function, two different GAs were run: one whose fitness function was the average number of classification errors, and the other whose fitness function was the maximum attribute-level error rate. In the following, the former GA will be referred to as GA1, and the latter will be referred to as GA2.

Results

Table 1 presents the average number of classification errors for every method and condition, with each average taken over the corresponding test set. GA1 exhibited the best average in seven of eight conditions, with GA2 or CDI achieving the second-best average. That GA1 outperformed the other methods based on the average number of classification errors was not surprising, considering that it is specifically designed to optimize with respect to this outcome measure. BP's relatively weak performance was also expected, as its objective function (Equation 16) is not intended to minimize the average number of classification errors. However, the differences between methods were typically modest: the median absolute difference between GA1 and CDI results, for example, was 0.02, and the median percentage improvement of GA1 over CDI was 4.4%. Differences between CDI and BP were even smaller: median absolute difference and percentage improvement of CDI over BP were 0.005 and 0.6%, respectively.

Table 1: Average Number of Classification Errors, by Condition and Method

Constraints	*Prior*	*Item Discrimination*	*CDI*	*BP*	*GA1*	*GA2*
Yes	Uniform	High	0.47	0.51	0.45	0.51
Yes	Uniform	Low	0.91	0.91	0.90	0.91
Yes	Corr = 0.5	High	0.32	0.32	0.30	0.32
Yes	Corr = 0.5	Low	0.62	0.63	0.63	0.63
No	Uniform	High	0.42	0.42	0.34	0.36
No	Uniform	Low	0.88	0.89	0.84	0.85
No	Corr = 0.5	High	0.29	0.31	0.26	0.27
No	Corr = 0.5	Low	0.62	0.62	0.61	0.62

Turning to the second outcome measure, Table 2 shows the maximum attribute-level error rate for every method and condition. Here GA2 performed the best (or tied for the best) in each of the eight conditions. Again, this result was expected since GA2 is designed to search for the solution with the lowest maximum error rate. BP or GA1 always achieved the second-best outcome. Use of the BP method generally resulted in only a modest decrement in maximum error rate compared to the more intensive GA2 (median absolute difference of 0.4%, median relative difference of 3.8%). CDI always displayed the highest maximum error rate, in some cases substantially higher than that of BP (median absolute difference of 2.2%, median relative difference of 19.1%).

Table 2: Maximum Attribute Error Rate, by Condition and Method

Constraints	*Prior*	*Item Discrimination*	*CDI*	*BP*	*GA1*	*GA2*
Yes	Uniform	High	11.7%	10.5%	10.6%	10.5%
Yes	Uniform	Low	20.2%	18.8%	19.1%	18.6%
Yes	Corr = 0.5	High	8.5%	7.2%	7.5%	6.7%
Yes	Corr = 0.5	Low	13.7%	13.3%	13.1%	13.0%
No	Uniform	High	13.9%	9.7%	7.3%	7.3%
No	Uniform	Low	24.1%	18.4%	18.4%	17.2%
No	Corr = 0.5	High	9.6%	6.6%	6.2%	5.6%
No	Corr = 0.5	Low	16.6%	12.8%	13.8%	12.7%

Finally, Table 3 presents the range of error rates by method and condition. GA2 always had the most balanced accuracy across attributes, as indicated by its range. BP had the second-smallest range in seven of eight conditions, while CDI had the highest range in all eight conditions. Although the relative difference between BP and GA2 was often large (median of 50.0%), the absolute difference was generally low (median of 0.55%). On the other hand, reductions in range of BP compared to CDI tended to be large, both in terms of absolute difference and relative difference (median values of 5.1% and 75.5%, respectively).

Table 3: Range of Error Rates, by Condition and Method

Constraints	*Prior*	*Item Discrimination*	*CDI*	*BP*	*GA1*	*GA2*
Yes	Uniform	High	5.6%	1.2%	3.6%	0.6%
Yes	Uniform	Low	5.7%	1.2%	3.0%	0.7%
Yes	Corr = 0.5	High	3.7%	1.3%	2.0%	0.6%
Yes	Corr = 0.5	Low	3.1%	1.1%	1.2%	0.7%
No	Uniform	High	9.8%	2.7%	1.2%	0.4%
No	Uniform	Low	12.8%	1.2%	4.7%	0.5%
No	Corr = 0.5	High	6.4%	0.7%	2.1%	0.4%
No	Corr = 0.5	Low	7.9%	0.6%	3.0%	0.3%

Discussion

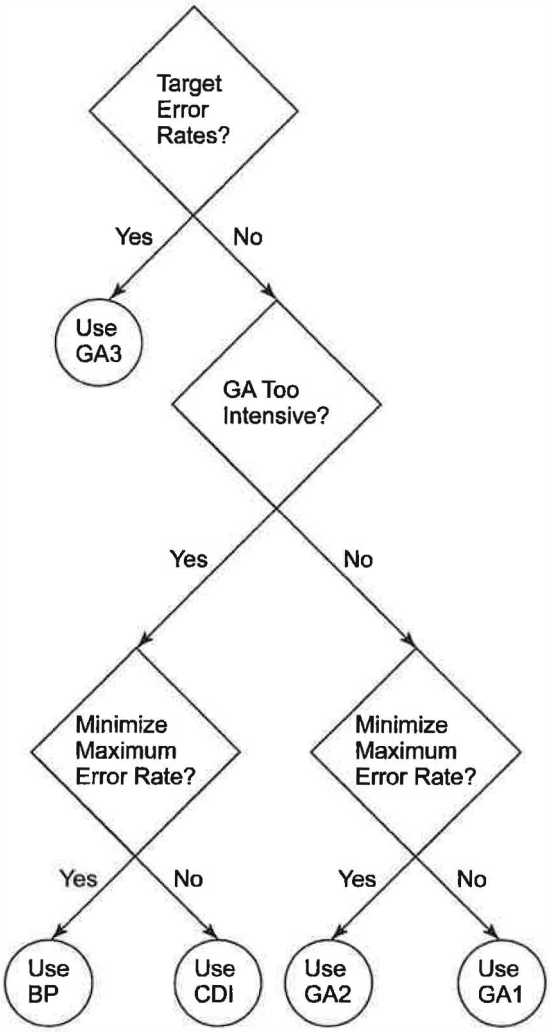
This article noted that the two existing ATA methods for CDMs each have a drawback: CDI cannot control error rates at the attribute level, and GA is computationally intensive. Our goal has been to develop a BP procedure that solves both of these problems while continuing to allow the satisfaction of all practical constraints. To decrease to chance that no attribute is measured with unduly poor accuracy, our BP utilized a maximin objective function alongside the attribute-level indexes of Henson et al. (in press), with special focus on the δ_j^B index.

We evaluated the BP procedure by comparing it with CDI, GA1, and GA2 under eight simulation conditions. In terms of the average number of classification errors, GA1 performed the best, while CDI exhibited better results than BP. However, in terms of both maximum error rate and range of error rates, BP outperformed CDI while serving as a computationally viable alternative to the best method, GA2. These results were all anticipated, considering that GA1 and CDI were designed for average accuracy, whereas BP and GA2 were designed to control attribute-level accuracy.

Such simulations demonstrate that there is no universal "best procedure" among existing ATA methods for CDM; therefore, the appropriate method to use must be decided on a case-by-case basis. Our recommendations are summarized in Figure 1, which is a flow chart indicating the best method for each situation. We first observe that GA is the only ATA-CDM procedure in the literature that can match actual attribute-level error rates to desired "target" error rates. Therefore, if a practitioner's goal is to assemble a test that achieves target error rates, then the target error rate version of GA (not explored in this study's simulations, but denoted "GA3" in Figure 1) is currently the only option. If target error rates are not specified, then the practitioner is asked whether the GA's computational intensity would be too burdensome. If not, there is no drawback to using GA, which has the best measurement properties; GA1 should be used among practitioners who seek to minimize the average number of classification errors, and GA2 should be used among those who seek to minimize the maximum attribute-level error rate. If the GA is too burdensome, then one of the less computationally intensive methods (CDI or BP) should be employed as an approximation. CDI is preferred when the goal is

to minimize the average number of classification errors; BP using the objective function of Equation 16 is preferred when the goal is to minimize the maximum attribute-level error rate.

Figure 1: Flow Chart for Determining which ATA Procedure to Use with CDMs



We emphasize that like GA, BP is flexible in that its objective function can be tailored to the goal of a practitioner. That is, while Equation 16 was utilized in our specific application of BP, any linear objective function can be used in the BP paradigm. For example, it was shown that the BP program CPLEX can be used to optimize the CDI alongside practical constraints; such constraints may be too complicated for the heuristic

method of Henson and Douglas (2005) to handle. Additionally, if certain attributes are more important than others, then a linear combination of the δ_{ik}^{B} values may be taken as a BP objective function, with the most important attributes given the most weight. A maximum number of items per attribute may also be listed as a constraint if the maximin approach results in the selection of too many items focusing on one difficult-to-measure attribute. Finally, as mentioned by Finkelman et al. (in press), the attribute-level error rates themselves are non-linear and hence cannot be minimized directly via BP. However, linear indexes like CDI, δ_j^A, δ_j^B, or δ_j^C have already been shown to perform well in ATA, and future linear indexes are likely to be developed as even better approximations to the error rates. All such indexes will be candidates for the BP objective function.

While the adoption of Finkelman et al.'s (in press) eight simulation conditions promoted the comparability of results, it also resulted in the same types of limitations. Additional studies should compare the CDI, BP, and GA methods under new conditions. As cited by Finkelman et al., such studies should use the DINA, NIDA, and compensatory MCLCM as models for generating examinee responses; different Q-matrices, item parameters, and constraints should be inputted; robustness to a misspecified prior distribution should be investigated; performance in operational settings should be analyzed. All such topics will be undertaken in future work.

References

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.* Reading, MA: Addison-Wesley.

Chang, H., & Ying, Z. (1996). A global information approach to computerized adaptive testing. *Applied Psychological Measurement, 20,* 213-229.

Cover, T.M., & Thomas, J.A. (1991). *Elements of information theory.* New York: John Wiley & Sons, Inc.

DiBello, L.V., Stout, W.F., & Roussos, L.A. (1995). Unified cognitive/psychometric diagnostic assessment likelihood-based classification techniques. In P.D. Nichols, D.F. Chipman, & R.L. Brennan (Eds.), *Cognitively diagnostic assessment* (pp. 361-389). Hillsdale, NJ: Erlbaum.

Finkelman, M., Kim, W., & Roussos, L. (in press). Automated test assembly for cognitive diagnosis models using a genetic algorithm.

Haertel, E.H. (1984). An application of latent class models to assessment data. *Applied Psychological Measurement, 8,* 333-346.

Haertel, E.H. (1990). Continuous and discrete latent structure models of item response data. *Psychometrika, 55,* 477-494.

Hamming, R.W. (1950). Error detecting and error correcting codes. *Bell System Technical Journal, 26,* 147-160.

Henson, R., & Douglas, J. (2005). Test construction for cognitive diagnosis. *Applied Psychological Measurement, 29,* 262-277.

Henson, R.A., Roussos, L.A., Douglas, J.A., & He, X. (in press). Cognitive diagnosis attribute level discrimination indices. *Applied Psychological Measurement.*

Holland, J. (1968). *Hierarchical description of universal spaces and adaptive systems.* Technical Report ORA projects 01252 and 08226, Ann Arbor, MI: University of Michigan.

Holland, J. (1973). Genetic algorithms and the optimal allocation of trials. *SIAM Journal of Computing, 2,* 88-105.

Holland, J. (1975). *Adaptation in natural and artificial systems.* Ann Arbor: University of Michigan Press.

Junker, B.W., & Sijtsma, K. (2001). Cognitive assessment models with few assumptions, and connections with nonparametric item response theory. *Applied Psychological Measurement, 25,* 258-272.

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

Maris, E. (1999). Estimating multiple classification latent class models. *Psychometrika, 64,* 187-212.

Roussos, L.A., DiBello, L.V., Stout, W., Hartz, S.M., Henson, R.A., & Templin, J.L. (2007). The fusion model skills diagnosis system. In J.P. Leighton & M.J. Gierl (Eds.), *Cognitive diagnostic assessment for education: Theory and applications* (pp. 275-318). Cambridge, UK: Cambridge University Press.

Tatsuoka, K.K. (1985). A probabilistic model for diagnosing misconceptions in the pattern classification approach. *Journal of Educational Statistics, 12,* 55-73.

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

van der Linden, W.J. (2005). *Linear models for optimal test design.* New York: Springer.

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

Veldkamp, B.P., & van der Linden, W.J. (2002). Multidimensional adaptive testing with constraints on test content. *Psychometrika, 67,* 575-588.

Verschoor, A.J. (2007). Genetic algorithms for automated test assembly. Doctoral dissertation. University of Twente, Enschede, The Netherlands.

von Davier, M. (2005). A general diagnostic model applied to language testing data (ETS Research Report 05-16). Princeton, NJ: Educational Testing Service.

Zhang, J., & Stout, W. (1999). The theoretical DETECT index of dimensionality and its application to approximate simple structure. *Psychometrika, 64,* 213-249.