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

Matthew D. Finkelman, Wonsuk Kim, Louis Roussos and Angela Verschoor

Applied Psychological Measurement 2010 34: 310 originally published online 18 May 2010
DOI: 10.1177/0146621609344846

Applied Psychological Measurement
34(5) 310–326

DOI: 10.1177/0146621609344846
http://apm.sagepub.com

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

Matthew D. Finkelman[1], Wonsuk Kim[2], Louis Roussos[2], and Angela Verschoor[3]

Abstract

Automated test assembly (ATA) has been an area of prolific psychometric research. Although 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 and to use a genetic algorithm. Each of these procedures has a disadvantage: The cognitive diagnostic index cannot control attribute-level information and the genetic algorithm 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.

Keywords

cognitive diagnosis models, automated test assembly, binary programming, test construction

One of the most common problems in test design is the construction of a linear form from a precalibrated 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.

Although BP is now considered a standard ATA solution for unidimensional models, it has yet to be used alongside cognitive diagnosis models (CDMs). One reason for this is that BP objective

[1]School of Dental Medicine, Tufts University, Boston, Massachusetts
[2]Measured Progress, Dover, New Hampshire
[3]Cito, Arnhem, The Netherlands

Corresponding Author:
Matthew D. Finkelman, School of Dental Medicine, Tufts University, One Kneeland Street, Boston, MA 02111
Email: matthew.finkelman@tufts.edu

functions typically use the concept of Fisher information (see Table 3 of van der Linden & Boekkooi-Timminga, 1989), and Fisher information is undefined for CDMs (Henson & Douglas, 2005). As a result, researchers have had to devise 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 (2009) 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, Roussos, Douglas, & He, 2008). Therefore, a test consisting of items with high CDI values may still produce poor measurement for some attributes. Second, the GA involves simulation and a local search algorithm; thus, although 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 model 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 used is based on the attribute-level item discrimination indexes of Henson et al. (2008) and therefore provides adequate measurement of all attributes. In addition, the procedure is much less computationally intensive than the GA approach.

To avoid confusion, it is noted that any test assembly problem can technically be formulated as a binary programming problem, because each item from the pool is either included in or excluded from the solution. Therefore, the term *BP* may refer to the broad class of methods whereby certain items are selected from a larger pool. In this article, however, the term *BP model* is used to refer specifically to a model that is proposed in the present study (Equations 17 and 18), which can be solved with branch-and-cut methods as implemented in commercially available software packages like CPLEX or LINGO. Such packages are often referred to as LP solvers.

First, a brief introduction to the concepts and notation of CDMs is provided. Then the current ATA methods for CDMs, namely CDI and GA, are reviewed, before proposing a specific BP model for CDMs. This BP model is compared with CDI and GA in multiple simulation sets. The article concludes by discussing the situations in which each model is appropriate.

CDMs

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 $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K)$ of K variables; α_k indicates the examinee's true ability along attribute k. Like most CDM research, this study assumes only two ability levels for each attribute, so that $\alpha_k = 1$ indicates mastery of attribute k and $\alpha_k = 0$ indicates nonmastery 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 $j = 1, \ldots, J$ index the items in a given pool, the $[j, k]$ entry of the Q-matrix (hereafter denoted q_{jk}) is equal to one if item j measures attribute k, and zero 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. Available CDMs include the restricted latent class model (Haertel, 1984, 1990) or deterministic input, noisy ''and'' gate (DINA) model (Junker & Sijtsma, 2001); noisy input, deterministic ''and'' gate (NIDA) model (Junker & Sijtsma, 2001); and the compensatory multiple classification latent class model (MCLCM; Maris, 1999; von Davier, 2005). Finkelman et al. (2009) 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, the present study also focuses on the reduced RUM.

The reduced RUM assumes that all items are dichotomously scored as correct or incorrect. Intuitively, its idea is that 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 nonmastery, 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 nonmaster of attribute k. Because masters are assumed to have greater acumen than nonmasters, $\pi_{jk} \geq r_{jk}$ is required; equivalently, the ratio $r_{jk}^* \equiv \frac{r_{jk}}{\pi_{jk}}$ is required 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 (Roussos et al., 2007)

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

From this formula, it can be seen that if $\alpha_k = 1$ for all k being measured (i.e., 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 nonmastery status along attribute k reduces the probability of a correct response by a factor of r_{jk}^*. It is noted 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. (2007) for further information.

In all that follows, it is assumed 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. It is emphasized that the reduced RUM is used only as an example; the CDI, GA, and BP models can be adapted to 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 nonmastery. 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 three-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 used in ATA (Henson & Douglas, 2005). To overcome this problem, Henson and Douglas developed an ATA procedure based on Kullback–Leibler distance (Chang & Ying, 1996; Cover & Thomas, 1991; Kullback & Leibler, 1951; Veldkamp & van der Linden, 2002). An item's Kullback–Leibler distance 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 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 distance of item j between $\boldsymbol{\alpha}'$ and $\boldsymbol{\alpha}''$ can be expressed as (Henson & Douglas, 2005).

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

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 distance values, including both $K_j(\boldsymbol{\alpha}',\boldsymbol{\alpha}'')$ and $K_j(\boldsymbol{\alpha}'',\boldsymbol{\alpha}')$, because these terms may be different from one another. Reasoning that it is most difficult to discern $\boldsymbol{\alpha}'$ from $\boldsymbol{\alpha}''$ 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. The resulting weight of the pair $(\boldsymbol{\alpha}', \boldsymbol{\alpha}'')$ can be expressed as

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

The average of the $K_j(\boldsymbol{\alpha}',\boldsymbol{\alpha}'')$ values, thus weighted, is the CDI of item j (Henson & Douglas, 2005):

$$\mathrm{CDI}_j = \frac{\sum_{\boldsymbol{\alpha}' \neq \boldsymbol{\alpha}''} \xi(\boldsymbol{\alpha}',\boldsymbol{\alpha}'')K_j(\boldsymbol{\alpha}',\boldsymbol{\alpha}'')}{\sum_{\boldsymbol{\alpha}' \neq \boldsymbol{\alpha}''} \xi(\boldsymbol{\alpha}',\boldsymbol{\alpha}'')}. \quad (4)$$

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., 2009).

GA

GAs 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. (2009).

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 nonmastery, 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 α, 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. (2009) proposed three fitness functions based on the values $e_1, \ldots, e_K$. They were (a) the sum of the error rates across all attributes; (b) the maximum error rate across all attributes; and (c) 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, $\varepsilon_1, \ldots, \varepsilon_K$, is determined a priori. The target error rate for attribute k is defined directly as the probability of an incorrect classification along this attribute, rather than as a function of a discrimination index such as the Kullback–Leibler distance. 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. (2009) 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 Finkelman et al. (2009) 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 used instead. This search begins with S initial solutions that satisfy each constraint; Finkelman et al. (2009) 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 prespecified 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 $(j_{s1}, \ldots, j_{sN})$ denote the indexes of the items in initial parent s, where $s = 1, 2, \ldots, S$. In the mutation scheme of Finkelman et al. (2009), 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, that is, from the set whose addition to the vector $(j_{s2}, \ldots, j_{sN})$ 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 $(j_{s1}, j_{s3}, \ldots, j_{sN})$ allows feasibility. The resulting child is $(j_{s1}, j'_{s2}, j_{s3}, \ldots, j_{sN})$. 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, and because 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 because 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, both simulations and a computer search must be performed. Finkelman et al.'s (2009) GA described above was specifically chosen for its relative simplicity; nevertheless, its running time may be burdensome for some applications.

A New Binary Programming Model

To overcome the drawbacks of test construction using the CDI and GA, it seems wise to revert to a model based on binary programming methods. This has the advantages that methods like branch-and-cut are faster and optimality of the presented solution can be proven. On the other hand, BP models are limited to a linear or quadratic objective function and linear constraints.

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, ATA models can be expressed as follows. Let x_j, $j = 1, \ldots, J$, 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; although y depends on the items selected (i.e., the x_j), this dependence is suppressed in the notation for greater simplicity. Next, it is assumed that the constraints are defined in terms of characteristics C_i, such 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 $i = 1, \ldots, I$. Then the task is to maximize y subject to the constraints

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

where $i = 1, \ldots, I$. 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. It is required 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 as

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

Constraints for the other content areas are defined analogously. If only a lower bound is desired (i.e., if at least 10 items measuring algebra are sought), 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, it is desired 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 between 8 and 12 items with this answer choice are sought. The inequality becomes

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

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

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

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

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

As stated previously, the objective function y is usually related to Fisher information in ATA models. Because Fisher information does not exist for CDMs, a different objective function in the current application must be used. The next subsection is devoted to developing the objective function, which is based on the attribute-level discrimination indexes of Henson et al. (2008).

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. In general, LP solvers cannot handle such an objective function, because the relations among error rates, item parameters, and the prior distribution are too complicated to be used directly (Finkelman et al., 2009). Instead, the heuristic indexes of Henson (2004) and Henson et al. (2008) are used here, which were designed to measure the information of each attribute and thus may be used as a proxy for their error rates.

Henson (2004) and Henson et al. (2008) proposed several such indexes for CDMs; in increasing order of complexity, they are denoted as δ_j^A, δ_j^B, and δ_j^C in this article.[1] As will be seen, our definition of each index results in a vector of K values (one for each attribute), for example,

$\delta_j^A = (\delta_{j1}^A, \ldots, \delta_{jK}^A)$. 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 used alongside any of the three indexes listed above. Only one index should be chosen for a given application; for illustration, the objective function is demonstrated 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, whereas the latter does not consider such prior probabilities. δ_j^B was chosen instead of 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 (2004) for details. It is emphasized that although 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, attention is restricted to those patterns, $\boldsymbol{\alpha}'$ and $\boldsymbol{\alpha}''$, that only differ in that one particular attribute (i.e., 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 distance (Equation 2). The Kullback–Leibler values are then combined into summary statistics, which are 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 (Henson et al., 2008),

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

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:

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

Henson et al. (2008) 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 distances, 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 $w_1(\boldsymbol{\alpha}') = P(\boldsymbol{\alpha}' | \alpha_k = 1)$ and $w_0(\boldsymbol{\alpha}') = P(\boldsymbol{\alpha}' | \alpha_k = 0)$.

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 nonmastery. These indexes were kept separate by Henson et al. (2008) because the Kullback–Leibler distance 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 nonmasters along attribute k, it can generally do so whether mastery or nonmastery is assumed. Hence, to create an overall index of item discrimination for attribute k, the average of Equations 13 and 14 is taken:

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

As claimed, the use of Equation 15 results in an index that is a vector of K values, $\delta_j^B = (\delta_{j1}^B, \ldots, \delta_{jK}^B)$. Henson (2004) also described the use of such an averaged version.

Although 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^B is that it is additive (Henson et al., 2008). Thus, to obtain the test's overall discrimination along attribute k, it suffices to sum the elements of the individual items. In other words, when using δ_j^B as an index, the total discrimination along attribute k is given by $\delta_{tk}^B = \sum_{\{x_j=1\}} \delta_{jk}^B$.

The proposed objective function. Although the main focus of Henson et al. (2008) was not ATA, they did state that their 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. Henson (2004) provided heuristics for test construction using the discrimination indexes; he also briefly discussed a method of test construction using integer programming. In this subsection, these suggestions are formalized by introducing the use of δ_j^A, δ_j^B, or δ_j^C as part of the BP objective function, thus optimizing with respect to these indexes.

It is observed that the above description can be thought of as a *maximin* problem: To ensure that all attributes are measured adequately, the minimum attribute-level discrimination is sought to be maximized. This maximin approach has been used 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, a similar procedure is used, with the discrimination indexes of Henson et al. (2008) substituted for Fisher information. Again, using δ_j^B as an example, the objective function is

$$y = \min_{k=1}^{K} \delta_{tk}^{B}. \tag{16}$$

It is important to note that LP solvers like CPLEX cannot maximize Equation 16 directly. One therefore needs to break down this equation into constraints with lower bound y and then maximize y. Similar to van der Linden, Ariel, and Veldkamp (2006), the complete minimax model takes the following form:

$$\text{maximize } y \tag{17}$$

subject to

$$\sum_{\{x_j=1\}} \delta_{jk}^{B} = \delta_{tk}^{B} \geq y,\ k = 1, \ldots, K \tag{18}$$

$$x_j \in \{0, 1\},\ j = 1, \ldots, J$$

$$\sum_{j=1}^{J} x_j = N$$

$$L_i \leq \sum_{\{x_j=1\}} C_{ij} \leq U_i, i = 1, \ldots, I,$$

where the last inequality includes all further constraints on item characteristics.

Simulation Studies

Method

Conditions. A previous study (Finkelman et al., 2009) compared CDI and GA under eight simulation conditions. To promote comparability with this study, the present study's design (comparing CDI, GA, and BP) was very similar to theirs. In particular, the same two item pools and prior distributions were used, whereas 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. Eighty items measured one of the five attributes, 140 measured two attributes, and 80 measured three attributes.

The first pool was constructed so that its items approximately matched those of real-data analyses by Jang (2005, 2006) and Roussos, Hartz, and Stout (2003). r^*_{ik} parameters were simulated from the uniform [0.40, 0.85] distribution, translating to a theoretical mean of 0.625 and a standard deviation about 0.13. These values are similar to those of Jang (2005, 2006), whose r^*_{ik} had a mean of 0.62 and a standard deviation of 0.14, and Roussos et al. (2003), who found a mean of 0.64 and did not report the standard deviation. The present study's π^*_i values were simulated from the uniform [0.75, 0.95] distribution, translating to a theoretical mean of 0.85 and a standard deviation of 0.06; in Jang's (2005, 2006) analyses, the mean and standard deviation of π^*_i were 0.83 and 0.13, respectively, and the standard deviation decreased to 0.08 on the removal of several items with unusually small values. Roussos et al. (2003) also found a mean π^*_i of 0.83. Hence, the present study's choices of parameters for this item pool are supported by real-data analyses.

To study the effect of item information on the methods' classification properties, the case where items had lower average Kullback–Leibler distance values was also examined. In this second pool, r^*_{ik} values were simulated from the uniform [0.65, 0.92] distribution (note that higher r^*_{ik} values yield lower information). The π^*_i were again simulated from the uniform [0.75, 0.95] distribution. Hence, the second pool still exhibited realistic π^*_i values and was intended to investigate whether the results would differ when the pool contained low-discriminating items.

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, that is, 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. (2009) and Henson and Douglas (2005), latent abilities were first generated from the multivariate standard normal distribution, with a tetrachoric correlation of .5 for each pair of attributes. α values were then created by dichotomizing each attribute into ''master'' or ''nonmaster'' categories depending on whether the continuous variable exceeded specified cut points. As in Finkelman et al. (2009), the cut points were defined so that the proportions of mastery in the population were .45, .50, .55, .60, and .65 for the five attributes.

Although the uniform prior distribution is unrealistic in cases where attributes are expected to exhibit positive correlation, this condition was considered useful because of its simplicity and

interpretability. The second condition, with a positive tetrachoric correlation, was more realistic. The correlation of .5 was small enough to avoid the undesirable circumstance where subscores are too highly correlated and therefore provide too little unique information. See Templin and Henson (2006) for an example where the correlations between latent attributes appeared to span a large range of values, from practically uncorrelated to very highly correlated, based on reported factor loadings.

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 8 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. (2009), 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.

In every condition, the task of each ATA method was to select 40 items out of the 300.

Outcome measures and α estimates. The three different methods were compared with respect to their overall accuracy and balance of accuracy across attributes. To standardize the comparison, all ATA methods were evaluated on the same ''test sets'' of simulee data. There were four test sets—one for every combination of item pool and prior distribution—each containing 20,000 simulees. Within a given test set, the number of simulees with each α vector was proportional to the prior distribution. Results were averaged over the 20,000 simulees to determine which methods performed the best in terms of three outcome measures: overall accuracy (defined as the average error rate over the five attributes), the maximum error rate, and the range of error rates. For all three outcome measures, smaller values corresponded to better performance.

To determine how many errors were made for a given simulee, an estimate of that simulee's α vector was required. The estimate $\hat{\alpha}$ that had been used by Finkelman et al. (2009) was utilized; this estimate is defined as the α vector minimizing the posterior expected error rate, given the prior distribution and the observed data. Because each ATA method selects different items, there is a different estimate $\hat{\alpha}$ for each method. It is noted 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. 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's (2005) iterative item selection approach (at each step, checking

Table 1 Average Error Rate, by Condition and Method

Constraint	Prior	Item discrimination	CDI (%)	BP (%)	GA1 (%)	GA2 (%)
Yes	Uniform	High	9.4	10.2	9.0	10.2
Yes	Uniform	Low	18.2	18.2	18.0	18.2
Yes	Correlation = .5	High	6.4	6.4	6.0	6.4
Yes	Correlation = .5	Low	12.4	12.6	12.6	12.6
No	Uniform	High	8.4	8.4	6.8	7.2
No	Uniform	Low	17.6	17.8	16.8	17.0
No	Correlation = .5	High	5.8	6.2	5.2	5.4
No	Correlation = .5	Low	12.4	12.4	12.2	12.4

CDI = cognitive diagnostic index; BP = binary programming; GA = genetic algorithm.

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. Integer programming had previously been suggested by Henson and Douglas as an alternative to their heuristic method.

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 B simulees (here, $B = 20{,}000$) 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. (2008), 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 error rate and the other whose fitness function was the maximum attribute-level error rate. In the following, the former GA is referred to as GA1, and the latter is referred to as GA2.

Results

Table 1 presents the average error rate 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 error rate 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 error rate. 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.

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 because 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 model generally resulted in only a modest decrement in maximum error

Table 2 Maximum Attribute Error Rate, by Condition and Method

Constraint	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	Correlation = .5	High	8.5	7.2	7.5	6.7
Yes	Correlation = .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	Correlation = .5	High	9.6	6.6	6.2	5.6
No	Correlation = .5	Low	16.6	12.8	13.8	12.7

CDI = cognitive diagnostic index; BP = binary programming; GA = genetic algorithm.

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 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, whereas 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).

Finally, Table 4 gives the number and percentage of overlapping items, for each pair of methods and condition. These ranged from 21 items (53%) to 38 (95%). Therefore, more than half the items overlapped in each comparison, but nonnegligible differences in selection occurred. In five of eight conditions, the highest percentage of overlap was that between BP and GA2. Generally, the overlap rates were greater in the constrained conditions than the unconstrained conditions, and also greater in the low-discriminating item pool than the high-discriminating item pool.

Discussion

Although GA supports general objective functions and constraints, it is slow when applied to CDMs because of its computational intensity. CDI also has a drawback: It cannot control error rates at the attribute level. In addition, neither of these models can prove the optimality of the

Table 3 Range of Error Rates, by Condition and Method

Constraint	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	Correlation = .5	High	3.7	1.3	2.0	0.6
Yes	Correlation = .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	Correlation = .5	High	6.4	0.7	2.1	0.4
No	Correlation = .5	Low	7.9	0.6	3.0	0.3

CDI = cognitive diagnostic index; BP = binary programming; GA = genetic algorithm.

Table 4 Pairwise Number (Percentage) of Overlapping Items, by Condition

Constraint	Prior	Item discrimination	CDI:BP	CDI:GA1	CDI:GA2	BP:GA1	BP:GA2	GA1:GA2
Yes	Uniform	High	30 (75)	35 (88)	27 (68)	27 (68)	37 (93)	24 (60)
Yes	Uniform	Low	33 (83)	34 (85)	30 (75)	33 (83)	35 (88)	31 (78)
Yes	Correlation = .5	High	34 (85)	34 (85)	31 (78)	32 (80)	35 (88)	29 (73)
Yes	Correlation = .5	Low	34 (85)	34 (85)	34 (85)	36 (90)	37 (93)	35 (88)
No	Uniform	High	29 (73)	24 (60)	21 (53)	24 (60)	25 (63)	29 (73)
No	Uniform	Low	29 (73)	27 (68)	26 (65)	25 (63)	28 (70)	31 (78)
No	Correlation = .5	High	28 (70)	26 (65)	25 (63)	21 (53)	23 (58)	28 (70)
No	Correlation = .5	Low	29 (73)	32 (80)	28 (70)	30 (75)	38 (95)	30 (75)

CDI = cognitive diagnostic index; BP = binary programming; GA = genetic algorithm.

solution provided. The goal of the present study has been to develop an ATA model that can be solved by binary programming methods like branch-and-cut while continuing to allow the satisfaction of all practical constraints. Branch-and-cut is limited by linear or quadratic objective functions and linear constraints, but solutions are found quickly and optimality can be proven. To decrease the chance that no attribute is measured with unduly poor accuracy, the BP model used a maximin objective function alongside the attribute-level indexes of Henson (2004) and Henson et al. (2008), with special focus on the δ_j^B index.

The BP model was evaluated by comparing it with CDI, GA1, and GA2 under eight simulation conditions. Each pair of methods exhibited item overlap rates of more than 50% in all conditions. The greatest concordance usually occurred between tests constructed by BP and GA2. The degree of overlap was dependent on the level of constraints and the discrimination of the item pool. Despite the fact that selected items between test construction methods were relatively concordant, a comparison of error rates revealed that the methods had nontrivial differences in accuracy. In terms of the average error rate, GA1 performed the best, whereas 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.

The simulations demonstrate that there is no universal ''best model'' among existing ATA methods for CDM; therefore, the appropriate method to use must be decided on a case-by-case basis. The present study's recommendations are summarized in Figure 1, which is a flow chart indicating the best method for each situation. It is first observed 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 error rate, 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 used as an approximation. CDI is preferred when the goal is to minimize the average error rate; BP is preferred when the goal is to minimize the maximum attribute-level error rate.

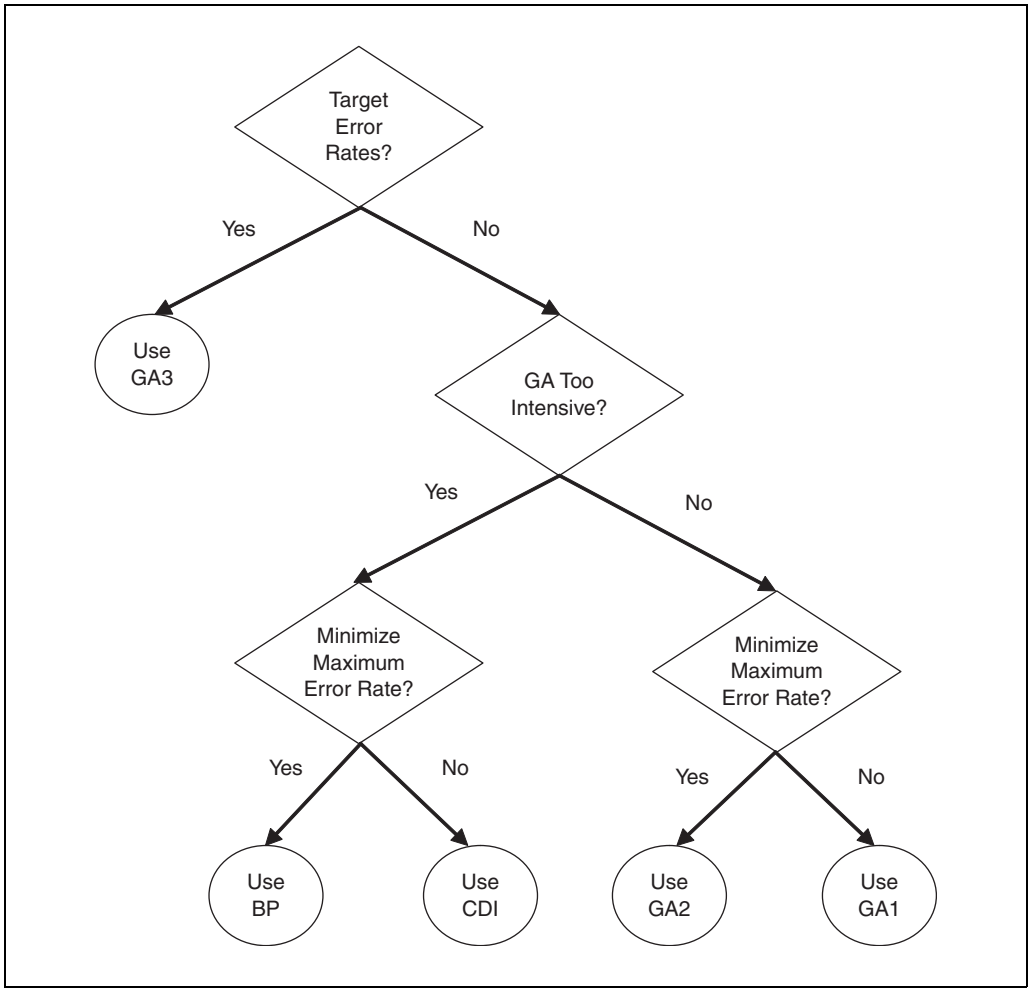

Figure 1 Flow chart for determining which ATA procedure to use with CDMs

It is emphasized that like GA, BP is flexible in that its objective function can be tailored to the goal of a practitioner. That is, although Equation 16 was used in the current application of BP, any linear objective function can be used in the BP paradigm. For example, if certain attributes are more important than others, then a linear combination of the δ_{tk}^{B} values may be taken as an alternative 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. (2009), the attribute-level error rates themselves are nonlinear and hence cannot be minimized directly via an LP solver like CPLEX. However, linear indexes like CDI, δ_j^A, δ_j^B, and 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 objective function.

Although the adoption of Finkelman et al.'s (2009) eight simulation conditions promoted the comparability of results, it also resulted in the same types of limitations. Because only one test

bank was used in a given simulation condition, the within-condition variation was not studied. However, results would not be expected to change appreciably considering that 300 items were generated in the bank; hence, differences between multiple item banks generated from the same parameter distribution would likely be slight. In addition, although the ATA methods considered here are designed to produce tests with high precision, such tests are not guaranteed to be favorable in terms of their Q-matrix complexities and other associated characteristics. This problem may be solved by inputting all important factors, including complexity, as explicit constraints to the solution.

Further studies should compare the CDI, BP, and GA methods under new conditions. 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; and performance in operational settings should be analyzed. All such topics will be undertaken in future work.

Acknowledgments

The authors thank the editor and two anonymous reviewers for their suggested improvements to a previous version of this article.

Declaration of Conflicting Interests

The authors declared no conflicts of interest with respect to the authorship and/or publication of this article.

Funding

The authors received no financial support for the research and/or authorship of this article.

Note

1. Henson (2004) used the notations $d_{(A)j}$, $d_{(B)j}$, and $d_{(C)j}$, respectively, rather than the notation used here. Henson et al. (2008) also used the $d_{(A)j}$ and $d_{(B)j}$ notations but did not discuss $d_{(C)j}$.

References

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.

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.

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). Mahwah, NJ: Lawrence Erlbaum.

Finkelman, M., Kim, W., & Roussos, L. (2009). Automated test assembly for cognitive diagnosis models using a genetic algorithm. *Journal of Educational Measurement, 46*, 273-292.

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. A. (2004). *Test discrimination and test construction for cognitive diagnosis*. Unpublished doctoral dissertation, University of Illinois, Urbana.

Henson, R. A., & 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. (2008). Cognitive diagnostic attribute-level discrimination indices. *Applied Psychological Measurement, 32*, 275-288.

Holland, J. (1968). *Hierarchical description of universal spaces and adaptive systems* (Tech. Rep. ORA projects 01252 and 08226). Ann Arbor: 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.

Jang, E. E. (2005). *A validity narrative: Effects of reading skills diagnosis on teaching and learning in the context of NG TOEFL*. Unpublished doctoral dissertation, University of Illinois, Champaign, IL.

Jang, E. E. (2006, April). *Pedagogical implications of cognitive skills diagnostic assessment for teaching and learning*. Paper presented at the annual meeting of the American Educational Research Association, San Francisco.

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.

Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics, 22*, 79-86.

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

Roussos, L. A., Hartz, S. M., & Stout, W. M. (2003, April). *Real data applications of the Fusion Model skills diagnostic system*. Paper presented at the annual meeting of the National Council on Measurement in Education, Chicago.

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

Templin, J. L., & Henson, R. A. (2006). Measurement of psychological disorders using cognitive diagnosis models. *Psychological Methods, 11*, 287-305.

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., Ariel, A., & Veldkamp, B. P. (2006). Assembling a computerized adaptive testing item pool as a set of linear tests. *Journal of Educational and Behavioral Statistics, 31*, 81-99.

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, 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.