Journal of Educational Measurement
Fall 2009, Vol. 46, No. 3, pp. 273–292

Automated Test Assembly for Cognitive Diagnosis Models Using a Genetic Algorithm

Matthew Finkelman
Tufts University School of Dental Medicine
Wonsuk Kim and Louis A. Roussos
Measured Progress

Much recent psychometric literature has focused on cognitive diagnosis models (CDMs), a promising class of instruments used to measure the strengths and weaknesses of examinees. This article introduces a genetic algorithm to perform automated test assembly alongside CDMs. The algorithm is flexible in that it can be applied whether the goal is to minimize the average number of classification errors, minimize the maximum error rate across all attributes being measured, hit a target set of error rates, or optimize any other prescribed objective function. Under multiple simulation conditions, the algorithm compared favorably with a standard method of automated test assembly, successfully finding solutions that were appropriate for each stated goal.

In recent years, much work in educational assessment has focused on providing detailed diagnostic information to examinees, going beyond the reporting of a single test score. For example, it might be desired that a mathematics test measure not only an examinee's general proficiency in the subject, but also specific abilities like factoring polynomials, solving quadratic equations, using laws of exponents, and manipulating fractions. To meet this need for diagnostic information, psychometricians have developed new models often referred to as "cognitive diagnosis models" (CDMs) because they are intended to provide more sophisticated parameterizations that better handle the complexities of cognitive psychological theories (Nichols, Chipman, & Brennan, 1995).

In CDMs, the ability of examinee j is characterized not by a single latent trait, θ_j, but by a vector $\boldsymbol{\alpha}_j = (\alpha_{1j}, \ldots, \alpha_{Kj})$ of K values, which are usually referred to as "attributes" or "skills." Of these terms, the former is more accurate because CDMs are applied to measuring a wide variety of examinee latent characteristics, from fine-grained cognitive skills to knowledge in broad content areas as well as personality traits (Roussos, Templin, & Henson, 2007). In fact, they are now considered relevant in any situation that involves obtaining information that is richer and more useful than a single score (Roussos et al., 2007). The goal of CDMs is to effectively measure and provide feedback on each attribute; parents, teachers, and the examinees themselves can then refocus instruction on the basis of this feedback.

Because the quality of a test's diagnostic information depends directly on the quality of its items, test assembly is a critical step in developing assessments. Indeed, many examinations are constructed by first developing a large pool of valid, reliable items, and then carefully selecting items from this pool to meet the substantive and statistical design constraints of the test. The goal of test development in this

context is the construction of a test of sufficient statistical performance (such as reliability, psychometric information, or measurement precision) yet also meeting a variety of substantive constraints (such as sufficient numbers of items measuring the skill and knowledge areas of concern, variety of types of items, key balancing for multiple-choice items, etc.). Within the context of using CDMs, statistical performance constraints are applied not simply to the test as a whole, but also to the individual attributes, and substantive constraints may also apply within as well as between attributes.

To streamline the process of selecting items from a pool, many researchers have studied automated test assembly (ATA), where a computer is used to select an appropriate set of items without human intervention (Theunissen, 1985; van der Linden, 2005; van der Linden & Boekkooi-Timminga, 1989). Henson and Douglas (2005) combined CDMs with ATA by developing a cognitive diagnostic index (CDI) for item discrimination and proposing the selection of items that maximize the index. Although this heuristic method was shown to substantially improve test precision compared to randomly determined tests, it has two potential shortcomings. First, as pointed out by Henson, Roussos, Douglas, and He (2008), the CDI is an overall measure of an item's discrimination; therefore, it does not give psychometric information at the attribute level. In many applications of test assembly, it is desired that a diagnostic test not only provide overall precision, but also accurate measurement of each attribute. Second, because Henson and Douglas recommended the selection of items with largest CDI values, the resulting test will always be maximally informative with respect to that index. This property of an ATA method is appropriate when the practitioner seeks to achieve as accurate a test as possible. However, in many applications of test assembly, the goal is to attain a specified (rather than maximized) level of precision; Birnbaum (1968), Lord (1980), and Theunissen (1985) discussed this *target function* approach when conducting ATA alongside a unidimensional item response theory (IRT) model. The application of target functions is common when the analysis of results includes year-to-year comparisons. In such cases, the current form is often designed to have psychometric properties similar to those of the previous year's form, promoting between-year comparability of results. To date, however, no CDI-based method exists to select items that meet a target level of overall precision. Achieving a target level of precision at each attribute would be even more difficult, if not impossible, for a CDI-based method in light of the fact that it does not measure discrimination at the attribute level.

In this article, we introduce a genetic algorithm (GA) method that overcomes the above difficulties. It is an ATA procedure that chooses item sets holistically rather than item by item, ensuring that the selected items complement one another. The method takes initial solutions as starting points, then assembles a test that always performs as well as or better than these starting points. This new method can optimize item selection either at the attribute level or based on overall precision, in accord with the practitioner's specifications. It is equally applicable whether minimizing the error rates of a test, hitting a set of target error rates, or optimizing any other prescribed objective function.

After introducing the GA, we turn our focus to comparing this method to the CDI in simulation. Our research design includes the evaluation of these two procedures in

terms of three fitness functions: average number of classification errors, maximum error rate, and ability to hit attribute-level target error rates. Multiple prior distributions, item pools, and levels of constraints are considered. The relative advantages and disadvantages of the GA and CDI are discussed.

Cognitive Diagnosis Models

CDMs are a class of statistical models that are capable of measuring multiple traits simultaneously. They are distinguished from multidimensional item response theory (MIRT) models (Reckase, 1985) in that CDMs classify examinees into a discrete number of levels (usually two) on each of the K attributes, whereas MIRT models typically measure the ability vector of examinee j, $\boldsymbol{\theta_j} = (\theta_{1j}, \ldots, \theta_{Kj})$, along a continuous K-dimensional spectrum of values. Hence, item and ability parameters of CDMs are especially tailored to classification, facilitating the interpretation of results when the goal is to categorize examinees into discrete levels. Without loss of generality, this article assumes that two levels (mastery and nonmastery) are specified for each attribute. An examinee's ability is defined by his or her $\boldsymbol{\alpha}_j$ vector: $\alpha_{kj} = 1$ if examinee j has achieved mastery on attribute k, and $\alpha_{kj} = 0$ otherwise ($k = 1, \ldots, K$).

A fundamental component of CDMs is the so-called Q-matrix (Tatsuoka, 1985), which indicates the attributes that are measured by each item. If there are I items (indexed by i), then the Q-matrix is an $I \times K$ matrix. A given element q_{ik} is 1 if item i measures attribute k, and 0 otherwise.

Many candidate CDMs have been proposed in the psychometric literature. These include the restricted latent class model (Haertel, 1984; also called DINA, Junker & Sijtsma, 2001), NIDA (Junker & Sijtsma, 2001), compensatory MCLCM (Maris, 1999; von Davier, 2005); and the reparameterized unified model (RUM; Roussos et al., 2007). We only provide a formal definition of the RUM here because this was the model utilized in simulation. However, we emphasize that both ATA procedures discussed in this article (CDI and GA) are general; they can be applied alongside any CDM. Thus, although the RUM was used for illustrative purposes, the comparison between the CDI and GA could have been made with any other CDM. It is also noteworthy that CDMs, like all other psychometric models, should always be tested for misfit when applying them to real data. See de la Torre and Douglas (2004), Roussos et al. (2007), and Sinharay (2006) for information on assessing the fit of CDMs.

To begin our review of the RUM, let Y_{ijk} be a latent variable representing the event that attribute k is correctly applied to item i by examinee j. Let $r_{ik} = P(Y_{ikj} = 1 \mid \alpha_{kj} = 0)$ and $\pi_{ik} = P(Y_{ikj} = 1 \mid \alpha_{kj} = 1)$, and define $r^*_{ik} = r_{ik}/\pi_{ik}$. In words, r_{ik} is the probability that attribute k is correctly applied to item i by an examinee who has not mastered this attribute, π_{ik} is the corresponding probability for an examinee who has mastered the attribute, and r^*_{ik} is the ratio between the two. Smaller r^*_{ik} values indicate better discernment between masters and nonmasters. Thus, r^*_{ik} essentially acts as a "reverse discrimination" parameter: the smaller it is, the higher the discrimination. Related to π_{ik} is another variable π^*_i, which denotes the probability of correctly applying all attributes that are required for item i, given that an examinee

has mastered all of these attributes. Note that if the application of attributes is locally independent, then

$$\pi_i^* = \prod_{k=1}^{K} \pi_{ik}^{q_{ik}}$$

(Roussos et al., 2007). Now let η_j denote the "supplemental ability" of examinee j; this variable represents a composite of residual abilities for attributes $\boldsymbol{\alpha}_b$ that are not included in the Q-matrix, but potentially have an effect on item responses. An associated quantity is the probability of correctly applying attributes $\boldsymbol{\alpha}_b$ to item i, given η_j; this value is denoted $P_{ci}(\eta_j)$. Finally, let X_{ij} be an indicator variable equal to 1 if examinee j answers item i correctly, and 0 otherwise. Under the RUM, the probability that examinee j answers item i correctly is

$$P(X_{ij} = 1 \mid \boldsymbol{\alpha}_j, \eta_j) = \pi_i^* \prod_{k=1}^{K} r_{ik}^{*\,(1-\alpha_{jk})q_{ik}} P_{ci}(\eta_j) \tag{1}$$

(Roussos et al., 2007). Our simulation study employed the "reduced" RUM, which is given by Equation 1 with the additional assumption that $P_{ci}(\eta_j) = 1$ for all i and j. Note that parameters are constrained to satisfy $0 < \pi_i^* < 1$ and $0 \leq r_{ik}^* \leq 1$. These conditions place probabilities in the appropriate range and guarantee that they are monotonic, that is, that a master of attribute k always has at least as high a chance of applying this attribute correctly as a nonmaster of the attribute.

Review of the CDI and ATA

When a unidimensional IRT model is employed, one common approach to ATA is to optimize selected items in terms of their combined Fisher information, or test information function (Birnbaum, 1968; van der Linden, 2005; van der Linden & Boekkooi-Timminga, 1989). A necessary assumption for computing Fisher information is that the log-likelihood function has a second derivative (Lord, 1980). This assumption does not hold for CDMs, which are formulated using discrete levels of ability rather than continuous distributions. Therefore, ATA based on Fisher information is inapplicable to tests employing a CDM (Henson & Douglas, 2005).

As an alternative to Fisher information, Henson and Douglas (2005) proposed a discrimination index for CDMs utilizing the concept of Kullback-Leibler information (Chang & Ying, 1996). Kullback-Leibler information is easily applicable to classification problems and does not require differentiability of the log-likelihood function. It is a measure of the distance between two distributions; here, the distributions in question are those induced by two different attribute vectors (say, $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}'$). An item that produces a large distance between the distributions induced by $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}'$ will in turn provide high discriminatory power for discerning these vectors. Therefore, the Kullback-Leibler information of item i between $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}'$, defined as

$$K_i(\boldsymbol{\alpha}, \boldsymbol{\alpha}') = E_\alpha \left[\log \frac{L(\boldsymbol{\alpha}; X_i)}{L(\boldsymbol{\alpha}'; X_i)} \right], \tag{2}$$

may be used as an index of the item's ability to distinguish $\boldsymbol{\alpha}$ from $\boldsymbol{\alpha}'$. Note that in Equation 2, the subscript j has been dropped from $\boldsymbol{\alpha}$, $\boldsymbol{\alpha}'$, and X for the purpose of simplicity (X_i still denotes an indicator variable for a correct response to item i). $L(\boldsymbol{\alpha}; X_i)$ is the likelihood of $\boldsymbol{\alpha}$ for the datum X_i. Intuitively, $L(\boldsymbol{\alpha}; X_i)$ indicates the degree to which the result of X_i is consistent with the hypothesis that $\boldsymbol{\alpha}$ is the true state of nature. The expectation is over the distribution of X_i given $\boldsymbol{\alpha}$; thus, for the dichotomous X_i assumed in this article, $K_i(\boldsymbol{\alpha}, \boldsymbol{\alpha}')$ can be written more explicitly as

$$K_i(\boldsymbol{\alpha}, \boldsymbol{\alpha}') = P_{\boldsymbol{\alpha}}(X_i = 1)\log\left[\frac{P_{\boldsymbol{\alpha}}(X_i = 1)}{P_{\boldsymbol{\alpha}'}(X_i = 1)}\right] + P_{\boldsymbol{\alpha}}(X_i = 0)\log\left[\frac{P_{\boldsymbol{\alpha}}(X_i = 0)}{P_{\boldsymbol{\alpha}'}(X_i = 0)}\right]$$

(Henson & Douglas, 2005).

One property of $K_i(\boldsymbol{\alpha}, \boldsymbol{\alpha}')$ is that it only measures the extent to which an item discerns between the two specific attribute patterns, $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}'$. The CDI takes into account all possible attribute pattern pairs; this index is a weighted average of the 2^K $(2^K - 1)$ pairwise $K_i(\boldsymbol{\alpha}, \boldsymbol{\alpha}')$ values. Because it is more difficult to disentangle patterns with many identical elements than those with many nonidentical elements, the CDI gives more weight to pairs of patterns with more identical elements. In particular, the weight, which we denote $\xi(\boldsymbol{\alpha}, \boldsymbol{\alpha}')$, is the reciprocal of the number of nonidentical elements:

$$\xi(\boldsymbol{\alpha}, \boldsymbol{\alpha}') = \frac{1}{\sum_{k=1}^{K}(\alpha_k - \alpha_k')^2}.$$

The CDI for item i, CDI_i, is then defined by the following weighted average of $K_i(\boldsymbol{\alpha}, \boldsymbol{\alpha}')$ values (Henson & Douglas, 2005):

$$CDI_i = \frac{\sum_{\boldsymbol{\alpha} \neq \boldsymbol{\alpha}'} \xi(\boldsymbol{\alpha}, \boldsymbol{\alpha}') K_i(\boldsymbol{\alpha}, \boldsymbol{\alpha}')}{\sum_{\boldsymbol{\alpha} \neq \boldsymbol{\alpha}'} \xi(\boldsymbol{\alpha}, \boldsymbol{\alpha}')}.$$

Henson and Douglas (2005) also provided an algorithm for using the CDI to conduct ATA, incorporating practical constraints such as content balance if necessary. In their algorithm, items are picked one at a time. Suppose that the test design specifies N items to be chosen out of the pool of I items, subject to a set of general constraints set by the practitioner (examples of constraints include balance across attributes and balance across the answer key). At each stage of selection, $n = 1, \ldots, N$, every candidate item is checked to determine whether all constraints can be met, given that the item is chosen. Items for which this statement is not true cannot be selected. Among the remaining items, the one with the largest CDI_i is added to the test. In the special case where there are no constraints, this reduces to selection of the N items with the highest CDI_i values.

The GA for CDMs

The CDI is an analytic procedure for finding a solution to the test assembly problem. In this section, a different approach based on GAs is undertaken. The GA and CDI are both heuristic methods, but GA uses a local search heuristic instead of an explicit formula to find its solution. As will be seen, a benefit of the GA is that it can be tailored to the outcome of interest (namely, the error rates of the test), rather than through an intermediate function based on Kullback-Leibler information. Thus, although the CDI is a reasonable index and performs significantly better than randomly determined tests (Henson & Douglas, 2005), the GA's direct attention to error rates carries the potential to enhance this improvement. Additionally, unlike the CDI, the GA offers the ability to achieve a target error rate for each attribute.

Genetic Algorithms

GAs are a class of techniques to find or approximate the optimal solution to a problem that is difficult to solve analytically. They were introduced by Holland (1973) in the computer science literature and can be applied to a diverse set of problems. GAs are a type of local search algorithm; they gradually converge to a final solution through a series of slight adjustments. The term "genetic" derives from an analogy to evolution: Candidate solutions compete with one another, with "stronger" solutions more likely to survive and produce other potential solutions known as "children." The system of candidates thus evolves and improves until it stabilizes. The GA then selects the best candidate in the system as its official solution.

In the case of ATA, the goal is to pick a set of N items out of a larger pool of I items. There are thus $\binom{I}{N}$ candidate sets of items from which to select. For most realistically sized item pools, there are far too many candidates to examine each one and choose the best. An analytic method like CDI may be used to make the selection; in this section, we instead study the potential of GAs to propose a suitable set.

The use of GAs and other local search algorithms is not new to the measurement field. In the domain of dimensionality analysis, Zhang and Stout (1999) utilized a GA to optimize the DETECT index. In the domain of ATA, Veldkamp (1999) used a local search algorithm called *simulated annealing* to assemble tests in the face of multiple objectives. van der Linden (2005) gave a general discussion of GAs in the ATA framework; Verschoor (2007) considered the topic in great depth, providing specific details on applying GAs to ATA with a unidimensional IRT model. Sun, Chen, Tsai, and Cheng (2008) also applied GAs alongside a unidimensional model, using them to assemble parallel test forms. Such works all illustrate the utility of local search algorithms as tools to solve measurement problems.

A GA begins with S initial candidate solutions that are selected to meet all content constraints; the S solutions may be based on statistical criteria or simply chosen at random from I_N^*, the set of solutions satisfying all constraints. For instance, S identical copies of the solution based on maximum CDI may be used. Alternatively, initial sets may be randomly generated and evaluated; the first S adhering to all constraints

are chosen. The initial S solutions are known as the "parents" of the first stage; each parent is composed of N items. The parents are then blended together or altered to form M children, with $M > S$; typically, each child is constrained to be a member of I_N^*. From this set of M children, S are chosen to "survive" to the next stage, with more desirable children more likely to survive. Desirability is measured by a fitness function (Verschoor, 2007)—a statistical criterion specifically constructed to rate the children. Once the S children have been selected for survival, those children become the parents for the next stage, creating their own children that will in turn be rated by the fitness function for survival. The process continues until either the fitness function ceases to improve from stage to stage (that is, the system converges), or a prescribed number of stages has passed. Once the stopping rule has been reached, the best solution remaining in the system is chosen.

Given a set of parent solutions, there are several ways to create children for the next stage. One method is *recombination*, also called *crossover*, by which two parents are blended into a child (van der Linden, 2005). The GA of this section uses *mutation,* rather than crossover; mutation alters the features of one parent to produce a child. As in previous sections, let i index the items in the pool. We assume that a parent is written as a vector of length N, where the vector components are the indexes of items in that parent. In mutation, a subset of a parent's items is removed and replaced by other items, creating a child. That is, l of the parent's items are exchanged with another l items from the pool (the value of l can be either deterministic or random). As always, the exchange is made judiciously so that the child satisfies all constraints.

Special Considerations for CDMs

A main goal of this article is to develop a GA that is specifically tailored to CDMs. As stated earlier, the combination of GAs and ATA is not novel; what is new here is their joint application to CDMs. Previous ATA methods using GAs do not "pour over" to CDMs because the usual fitness function is inapplicable. Specifically, the test information function (TIF) was instrumental to the fitness function of Verschoor (2007) and is a standard ATA criterion in unidimensional modeling; however, the TIF is generally undefined in CDMs due to these models' discretization of the ability space (Henson & Douglas, 2005). Therefore, a new fitness function is needed in order to use the GA alongside popular CDMs such as DINA, NIDA, compensatory MCLCM, and RUM. This section will introduce three fitness functions, denoted F_1, F_2, and F_3, that are tailored to CDMs.

To determine a suitable fitness function, recall that CDMs operate under a classification framework. Every examinee is categorized into one of multiple levels (here, two) along each of the K attributes. The error rates of the respective attributes are thus a fundamental measure of test accuracy. One natural desire would be to minimize the average number of classification errors that are produced in examination. For an individual examinee with a true attribute vector of $\boldsymbol{\alpha}$, the number of classification errors is $\sum_{k=1}^{K} |\alpha_k - \hat{\alpha}_k|$, where α_k is the true component of $\boldsymbol{\alpha}$ along dimension k and $\hat{\alpha}_k$ is the corresponding estimate based on the test. The expected number of

errors with respect to a prior distribution on $\boldsymbol{\alpha}$, $P(\boldsymbol{\alpha})$, is then given by

$$F_1 \equiv \sum_{\boldsymbol{\alpha}} P(\boldsymbol{\alpha}) E_{\boldsymbol{\alpha}} \left(\sum_{k=1}^{K} |\alpha_k - \hat{\alpha}_k| \right), \tag{3}$$

where $E_{\boldsymbol{\alpha}}(.)$ denotes expected value under $\boldsymbol{\alpha}$. F_1 is our first fitness function, with lower values considered better results. Note that this definition does not conform to the common convention that higher values of the fitness function connote "better" results, but this fact does not hinder its efficacy in any way. The negative of F_1 may be taken rather than F_1 itself, if it is desired that this convention be followed.

Equation 3 combines the error rates of the different attributes into a single index. However, as explained in the introduction, a practitioner may seek adequate precision along each attribute, rather than reducing the problem to an overall index. In this case, the error rates of the individual attributes may be considered. The error rate for attribute k is

$$e_k \equiv \sum_{\boldsymbol{\alpha}} P(\boldsymbol{\alpha}) E_{\boldsymbol{\alpha}}(|\alpha_k - \hat{\alpha}_k|).$$

To ensure that all attributes are measured precisely, a minimax approach may be taken, whereby we seek to minimize the maximum of the K error rates. That is, we minimize

$$F_2 \equiv \max_{k=1}^{K} \sum_{\boldsymbol{\alpha}} P(\boldsymbol{\alpha}) E_{\boldsymbol{\alpha}}(|\alpha_k - \hat{\alpha}_k|).$$

F_2 may be used as a fitness function, with lower values considered better.

Either of the previous fitness functions is appropriate if the practitioner's goal is to achieve maximal precision. However, some applications of testing may require that a test meets a target level of precision. As stated in the introduction, one notable example is the case where a test is given annually, and it is desired that the error rates of the current year match those of the previous year as closely as possible. In this situation, the estimated error rates of the previous test become the target.

Let ε_k denote the target error rate for attribute k; these values can be set a priori or to match the error rates of a previous form. A measure of the difference between the true error rates e_k and their targets ε_k is the sum of all absolute distances:

$$F_3 \equiv \sum_{k=1}^{K} |e_k - \varepsilon_k|.$$

As with F_1 and F_2, lower values of F_3 values are considered better.

One complicating factor in applying the above fitness functions to CDMs is that they are very difficult to compute analytically. Therefore, we propose that they be evaluated via simulation. In particular, B simulees are produced from the prior

distribution $P(\boldsymbol{\alpha})$, with $B * P(\boldsymbol{\alpha})$ simulees exhibiting a true state of $\boldsymbol{\alpha}$. Each simulee is administered all I items in the pool; that is, there is a single run in which a response for every simulee and every item is simulated. Based on this simulation run, the fitness functions may be evaluated for any candidate set of items. First, for simulee j and candidate set of items $\boldsymbol{u_i}$, the posterior distribution of $\boldsymbol{\alpha}_j$ is computed based only on the set of responses, $\boldsymbol{X_{ij}}$, that correspond to $\boldsymbol{u_i}$:

$$P(\boldsymbol{\alpha}_j \mid \boldsymbol{X_{ij}}) = \frac{P(\boldsymbol{\alpha}_j \mid \boldsymbol{X_{ij}})L(\boldsymbol{\alpha}_j; \boldsymbol{X_{ij}})}{\sum_{\boldsymbol{\alpha}} P(\boldsymbol{\alpha}_j \mid \boldsymbol{X_{ij}})L(\boldsymbol{\alpha}_j; \boldsymbol{X_{ij}})}.$$

From this posterior distribution, a classification of simulee j is made with respect to the candidate set $\boldsymbol{u_i}$. In particular, the simulee's ability estimate $\hat{\boldsymbol{\alpha}}_j$ is the vector that minimizes the expected number of errors with respect to the posterior distribution. That is, $\hat{\boldsymbol{\alpha}}_j$ is the vector minimizing

$$\sum_{\boldsymbol{\alpha}} \left[P(\boldsymbol{\alpha}_j \mid \boldsymbol{X_{ij}}) \left(\sum_{k=1}^{K} |\alpha_k - \hat{\alpha}_k| \right) \right]. \tag{4}$$

Once $\hat{\boldsymbol{\alpha}}_j$ has been determined, the expected posterior error rate for attribute k can be determined for simulee j. Let $\hat{\alpha}_{kj}$ represent the observed classification for simulee j along attribute k. Let $P(\alpha_k = 1 \mid \boldsymbol{X_{ij}})$ denote the posterior probability of mastery along attribute k for simulee j:

$$P(\alpha_k = 1 \mid \boldsymbol{X_{ij}}) = \sum_{\boldsymbol{\alpha}_j : \alpha_{kj} = 1} [P(\boldsymbol{\alpha}_j \mid \boldsymbol{X_{ij}})].$$

Then the expected posterior error rate, $E(e_{kj} \mid \boldsymbol{X_{ij}})$, for simulee j and attribute k is equal to $P(\alpha_k = 1 \mid \boldsymbol{X_{ij}})$ if $\hat{\alpha}_{kj} = 0$; it is equal to $1 - P(\alpha_k = 1 \mid \boldsymbol{X_{ij}})$ if $\hat{\alpha}_{kj} = 1$. The overall expected posterior error rate for attribute k is then obtained by averaging across all simulees:

$$\bar{e}_k = \frac{\sum_{j=1}^{B} E(e_{kj} \mid \boldsymbol{X_{ij}})}{B}.$$

The values $\bar{e}_k$ are then used as a building block to evaluate any of the fitness functions above—that is, F_1, F_2, or F_3—for the candidate set $\boldsymbol{u_i}$. These three fitness functions are evaluated by $\sum_{k=1}^{K} \bar{e}_k$, $\max_{k=1}^{K} \bar{e}_k$, and $\sum_{k=1}^{K} |\bar{e}_k - \varepsilon_k|$, respectively.

In light of the somewhat complicated simulation procedure described above, two observations are worth making. The first of these observations is that the use of simulation is not unprecedented in item selection. Most notably, simulation is a fundamental part of several popular computerized adaptive testing (CAT) methods to

curb the exposure rates of items (Stocking & Lewis, 1998; Sympson & Hetter, 1985). Although such simulation adds more time to the process, it has not prevented these CAT methods from being used operationally. The second observation is that due to the complexity of the above evaluation criteria, linear programming methods (van der Linden, 2005) for direct optimization are not as easily applicable as GAs. The three fitness functions introduced above define optimality with respect to a Bayesian prior distribution, whether that optimality is with respect to the minimization of the average number of errors, the minimax criterion, or the distance-to-target error rates. The relation between error rates and item parameters cannot be written as a linear function, and therefore linear programming is unable to directly optimize these error rates. However, the GA proposed in the next subsection can be used as a tool for optimizing (or approximating the optimum of) any of the three functions. This application to CDMs is thus an illustration of the added flexibility of GAs over linear programming techniques; such flexibility was previously listed as a benefit of GAs and other local search heuristics by van der Linden (2005) and Verschoor (2007).

The Specific GA for CDMs

As a general rule, there is a tradeoff between the quality of a GA's solution and its running time (van der Linden, 2005). After all, by allowing a GA to run longer, we enhance the chances of finding the optimal value of our selected fitness function. Due to the many calculations performed by the GA (described in the previous subsection), maintaining a reasonable running time is an important consideration in the current application to CDMs. To achieve the correct balance between running time and the solution's fitness, the GA proposed in this subsection is relatively simple. Although this simple GA may be more likely to find a local (rather than global) optimum, it is more practicable in terms of computational expense.

The first manner in which our chosen GA is simplified regards the creation of children from parents. We have used a mutation scheme in which items of the parent vector are perturbed to new values, one item at a time (mathematically, this amounts to setting $l = 1$). As in previous sections, let N denote the prescribed number of items for the test, and let $s = 1, \ldots, S$ index the S parent vectors at the current iteration of the GA. Now let $(i_{s1}, \ldots, i_{sN})$ represent the items comprising parent s. For the current iteration, N children are created for each parent. The first child is created by removing item i_{s1} and replacing it with another item, i'_{s1}, so that the child satisfies all content constraints. Following the notation above, the first child contains items $(i'_{s1}, \ldots, i_{sN})$. Similarly, the second child is composed of items $(i_{s1}, i'_{s2}, \ldots, i_{sN})$, and so forth until child N, which is composed of items $(i_{s1}, \ldots, i'_{sN})$. A clone of each parent is also retained as an additional child; that is, a parent at the current iteration is eligible to be a parent at the next iteration, if it is superior to the children. Thus, there are $N + 1$ children propagated by each parent, for a total of $M = S(N + 1)$ children.

The above procedure is a simplified GA in that it uses only mutation, not crossover. By only mutating one item at a time, we facilitate the search for children that meet all constraints. After all, for each item in parent s (which itself has been chosen to

meet each constraint), it is trivial to identify and restrict attention to potential replacement items whose addition would not violate any constraints. Note also that we have proposed each item to be replaced in exactly one child; this rule is more systematic than determining the items to be replaced at random. In particular, it allows for the potential replacement of each item at every stage, facilitating the identification and removal of less informative items. It was believed that this systematic approach would remove such items more quickly and consistently than random replacement: By chance alone, a random replacement might skip the potential removal of a problematic item for several iterations. Though not yet verified empirically, the systematic method was thus expected to result in faster convergence to an adequate solution than a random replacement of items.

The second manner in which our GA is simplified is in the selection of children for survival. As stated earlier, the children with more desirable values of the fitness function should be more likely to survive. If there is an element of randomness to the choice of children for survival, then the selection is said to be stochastic; if, on the other hand, only the strongest children are chosen, then the selection is deterministic (Verschoor, 2007). Although stochastic selection is able to backtrack from local optima, its adoption of interim solutions with suboptimal fitness causes a more gradual progression toward convergence. On the other hand, deterministic selection does not backtrack; because interim solutions always move forward in the direction of better fitness, a more direct path is taken to the ultimate solution. We therefore expect that a deterministic rule will result in faster convergence to an optimum, albeit perhaps a local one. To reduce computation time, as well as to eliminate the need for fine tuning associated with stochastic selection, we utilize deterministic selection in the GA of this article: the S children with the best fitness values are chosen.

Overall, our GA selects a test form for a CDM using the following steps:

1. Generate responses for B simulees and all I items in the pool.
2. From the given pool of items, select S initial "parent" solutions.
3. Perform the mutation scheme described above for each parent, resulting in $S(N + 1)$ children.
4. Evaluate the fitness function for all children. The S children exhibiting the best values of the fitness function are selected as parents for the next iteration.
5. Repeat steps 3 and 4 until the stopping rule is invoked.
6. Choose the solution in the system that exhibits the best value of the fitness function. This becomes the official solution of the GA.

We emphasize that more complicated variations of the GA could also be employed; such variations include the addition of crossover, the random replacement of items (rather than the replacement of each item in exactly one child), the use of stochastic selection, and the allowance of interim solutions to venture into infeasible space (rather than requiring them to satisfy all constraints). Provided that they are well implemented, these complex variations enhance the GA's ability to avoid local optima. The extent to which such complications would change the results obtained from the GA outlined above, as well as the extent to which they would increase

computation time, are questions that will be answered in future work. The goals of this article, however, are to introduce the GA method to CDMs and investigate whether it offers improvement over the CDI; therefore, our current attention will be restricted to these purposes.

Simulations

Background and Design

The previous two sections described algorithms for combining ATA with CDMs, including one previously proposed method (CDI) and one new method (GA). The goal of this section is to compare these methods in simulation. Specifically, the two approaches were evaluated based on their abilities to minimize the fitness functions F_1, F_2, and F_3. Results were analyzed separately for each of these three functions. The simulation design used two item pools, two prior distributions on $\boldsymbol{\alpha}$, and two levels of item constraints; these factors were completely crossed for a total of eight conditions.

Item pools. Each item pool consisted of 300 items; the two pools were based on the same Q-matrix, which contained five attributes. The choice of 300 items and five attributes was similar to the design of Henson and Douglas (2005), whose method involved simulations with a pool of 300 items and either four or eight attributes. Because items typically do not measure all attributes, this study's Q-matrix was constrained to have 80 items measuring one attribute, 140 items measuring two attributes, and 80 items measuring three attributes, for an average of two attributes per item. Within each item, the attribute or attributes being measured were randomly determined. This procedure resulted in 135 items measuring attribute 1, 115 measuring attribute 2, 108 measuring attribute 3, 115 measuring attribute 4, and 127 measuring attribute 5 across the two pools.

The two item pools were different in that one consisted of more highly discriminating items than the other. In both cases, items were assumed to follow the reduced RUM, with specific item parameters generated at random. For both pools, every π_i^* was randomly generated from the $U[0.75, 0.95]$ distribution. All r_{ik}^* values were randomly generated from $U[0.40, 0.85]$ for the pool with high discrimination and $U[0.65, 0.92]$ for the pool with low discrimination. These ranges of r_{ik}^* are similar to those obtained in analyses by Jang (2005, 2006) and Roussos, Hartz, and Stout (2003).

Prior distributions on $\boldsymbol{\alpha}$. First, we investigated the situation where the Bayesian prior on $\boldsymbol{\alpha}$ was uninformative, that is, where each of the $2^5 = 32$ possible $\boldsymbol{\alpha}$ vectors had $P(\boldsymbol{\alpha}) = 1/32$. The second prior distribution corresponded to the case where examinees' latent abilities followed the multivariate standard normal distribution; these latent abilities were then compared to specified cut points in order to determine mastery or nonmastery on each attribute. The tetrachoric correlation of each pair of attributes was set to .5, and the cut points were set so that the percentage of masters on each attribute was 45, 50, 55, 60, and 65 for attributes 1, 2, 3, 4, and 5, respectively. This prior distribution was considered more realistic than the uninformative prior: in educational measurement, latent abilities are usually considered to be positively

correlated (Henson & Douglas, 2005), and the percentage of masters is seldom exactly 50% for all attributes.

Constraints. Two different levels of constraints were specified. The first level imposed no constraints and was viewed as a baseline condition. The second level was more realistic, imposing two types of constraints: a lower bound on the number of times that each attribute must be measured, and a proper balance across the answer key. Specifically, it was required that the 40 selected items measured each attribute at least 15 times; a similar constraint on content had been studied by Henson and Douglas (2005). This constraint ensured that all content areas would be adequately represented, and thus the reporting of subscores would be defensible. Each item was also randomly assigned a correct answer of A, B, C, or D; candidate item sets were considered feasible only if each answer choice was represented between 8 and 12 times. Such balance ensures that capable examinees will not be distracted by a perceived bias toward one answer choice over another.

It should be noted at this juncture that the use of more complicated substantive constraints can be easily incorporated into the GA. For example, constraints on more fine-grained content areas could be introduced within the measurement of each attribute, as well as more sophisticated constraints having to do with student cognitive modeling (such as different strategies leading to the use of different Q-matrices).

Other simulation details. One of the GA's benefits is that it is flexible to the goal: Unlike the CDI, it tailors itself to the choice of fitness function. To test this characteristic, the GA was run for each of the fitness functions F_1, F_2, and F_3 and compared with the CDI in simulation results. That is, the fitness function was considered an input to the GA, so that the GA was tailored to the prescribed function when selecting its items. Each GA solution was only evaluated with respect to the fitness function to which it was tailored. In all cases, $S = 3$ copies of the CDI solution were chosen as initial parents. Setting $S = 3$ had been successful in a previous psychometric application of GAs (Zhang & Stout, 1999) and was expected to provide adequate variation in the children without creating an intractable computational burden.

As described in the subsection entitled "Special Considerations for CDMs," the GA utilizes preliminary simulations in order to conduct item selection. For each prior distribution and item pool, 20,000 "training" simulees were tested, with 20,000 $P(\boldsymbol{\alpha})$ simulees exhibiting a true ability vector of $\boldsymbol{\alpha}$. There were thus four training data sets, from which both the constrained and unconstrained item selections were made. The GA was stopped if the fittest solution in the system remained the same for 50 iterations, or if 500 total iterations were run, whichever came first. A FORTRAN 6.1.0 program was specially written to carry out all steps of the analysis.

To compare the methods, a common set of simulees was needed. The training data sets were not ideal for this purpose because the GA had selected its items based on these data sets; hence, the GA would be likely to "capitalize on chance" if the training sets were also used in evaluation. In other words, comparison of the methods using training data would constitute an unfair advantage for the GA. To avoid this problem, four new sets of 20,000 simulees were produced, again with true $\boldsymbol{\alpha}$ values generated proportional to their prior probabilities, and these test sets were used to evaluate the

TABLE 1
Average Number of Classification Errors, by Condition and Method

Constraints	Prior	Item Discrimination	CDI	GA	GA% Gain
Yes	Uniform	High	.402	.342	15.0%
Yes	Uniform	Low	.879	.836	4.9%
Yes	Corr = .5	High	.286	.256	10.5%
Yes	Corr = .5	Low	.617	.608	1.4%
No	Uniform	High	.418	.335	19.9%
No	Uniform	Low	.879	.837	4.8%
No	Corr = .5	High	.292	.258	11.6%
No	Corr = .5	Low	.620	.611	1.3%

TABLE 2
Maximum Attribute Error Rate, by Condition and Method

Constraints	Prior	Item Discrimination	CDI	GA	GA% Gain
Yes	Uniform	High	.118	.072	39.0%
Yes	Uniform	Low	.226	.182	19.6%
Yes	Corr = .5	High	.084	.057	32.4%
Yes	Corr = .5	Low	.159	.125	21.1%
No	Uniform	High	.139	.071	48.5%
No	Uniform	Low	.241	.176	27.2%
No	Corr = .5	High	.096	.056	41.9%
No	Corr = .5	Low	.166	.132	20.7%

methods. Simulees were classified based on the items selected by each ATA method, using the $\hat{\boldsymbol{\alpha}}$ estimates minimizing expression 4. The true $\boldsymbol{\alpha}$ values were compared to these $\hat{\boldsymbol{\alpha}}$ estimates, yielding observed values of F_1, F_2, and F_3.

Results Comparing the Methods

Table 1 gives the average number of classification errors for each condition and ATA procedure, based on the test sets. This outcome measure corresponds to the observed values of F_1. Table 1 indicates that the GA resulted in an improvement over CDI on all test sets. Over the eight conditions studied, the GA's percentage reduction of average errors ranged from 1.3% to 19.9% as compared to the CDI; the greatest reductions were found in the highly discriminating item pool. As in Henson and Douglas (2005), the CDI's average number of errors actually decreased under conditions with constraints, due to a greater balance of attributes measured. In training data, the GA always exhibited better performance without constraints than with them (results not shown); as Table 1 indicates, this trend sometimes reversed in the test data.

Next, we compared the methods based on their observed values of F_2. Here, the GA always exhibited a substantial advantage over the CDI, with percentage reduction

TABLE 3
Sum of the Absolute Distances Between Realized and Target Error Rates, by Condition and Method

Constraints	Prior	Item Discrimination	Target Error Rate Per Attribute	CDI	GA	GA% Gain
Yes	Uniform	High	.10	.135	.009	93.2%
Yes	Uniform	Low	.20	.173	.015	91.4%
Yes	Corr = .5	High	.10	.214	.011	95.1%
Yes	Corr = .5	Low	.15	.151	.015	89.8%
No	Uniform	High	.10	.179	.010	94.2%
No	Uniform	Low	.20	.230	.005	97.7%
No	Corr = .5	High	.10	.208	.011	94.8%
No	Corr = .5	Low	.15	.175	.004	97.9%

of maximum error rate ranging from 19.6% to 48.5%. Again, the observed reductions tended to be larger for the highly discriminating item pool. See Table 2 for a complete list of maximum error rates.

Table 3 gives the sum of the absolute distances between realized and target error rates (i.e., observed values of F_3) for each condition and method. Not surprisingly, the GA performed much better than selection based on maximum CDI, which was not tailored to the targets. The GA's percentage reduction, compared to CDI, was always at least 89.8% over all eight conditions; the highest reduction was 97.9%.

The above simulations assume that the $P(\boldsymbol{\alpha})$ values are known. When an informative prior is utilized, this assumption should generally benefit methods, such as the GA, that incorporate the $P(\boldsymbol{\alpha})$ into the item selection process. Thus, one limitation of the simulation study is that it does not investigate the GA's robustness to a misspecified prior. Nevertheless, the conditions involving a uniform prior can be utilized to make a "fair" comparison between the CDI and GA. In these cases the GA did not benefit from informative $P(\boldsymbol{\alpha})$ values, but it achieved better results than the CDI for all three fitness functions considered.

How Good Is the GA's Solution?

The previous subsection assessed the GA's performance on test data. Attention now turns to its fundamental ability to locate a high-quality solution from the training data, over which the search is actually conducted. The following analysis is designed to evaluate the extent to which the GA is adept at finding good approximations to the optimal solution, as opposed to a local optimum far from the global optimum.

An ideal evaluation of the GA solution would involve systematically calculating the fitness of all possible $\binom{300}{40}$ item sets for each condition, then comparing the GA solution's fitness compared to that of the optimum. This method is computationally intractable, however, due to the fact that this is an astronomically high number of item sets. Therefore, three other approaches were taken to analyze the GA's ability to find a high-quality solution within the training data. The three approaches were:

(a) comparison of the GA solution with the distribution of values observed in 4,000 randomly determined item sets, (b) comparison of GA results when varying the initial parents, and (c) comparison of the GA solution with all possible $\binom{I}{N}$ item sets, with I and N chosen to be computationally tractable. None of these short investigations represents an irrefutable confirmation that the GA solution is always close to optimal; nevertheless, they give some evidence of the GA's performance with respect to optimality.

For brevity, F_1 was considered to be the only fitness function of interest in all investigations. Also, only one condition was analyzed in each investigation, namely the condition with highly discriminating items, a uniform prior on $\boldsymbol{\alpha}$, and no constraints. The choice of a uniform prior and no constraints was made because of the simplicity and interpretability of these features; the choice of the highly discriminating item pool was arbitrary. Due to the complex relationship between item parameters and error rates, as well as the complexity of GA methodology itself, it is difficult to predict which conditions will yield similar results to those obtained here. In general, the relative quality of a GA's solution may be expected to depend upon the size of the item pool, the number of items to be selected, the degree to which the optimal solution is superior to other candidate solutions, and the proximity of the initial solutions to the optimal solution (as well as their proximity to local optima). Therefore, caution should be observed when generalizing the following results to other conditions.

Comparison with randomly determined item sets. In this investigation, 4,000 item sets out of the possible number were chosen at random and compared to the GA solution (as well as the CDI) based on performance over the training data. Of the 4,000 randomly determined item sets, the best set had an average number of classification errors of .539. The mean of the distribution for the 4,000 values was .675, and the standard deviation was .043. Both ATA methods resulted in values (.411 for CDI, .329 for GA) that were substantially better than any of the randomly selected sets. Note that the values presented here are different from their counterparts in Table 1, which were the results of test data rather than training data.

Comparison of GA results when varying the initial parents. In all previous applications of the GA considered herein, the same initial parents were used: three copies of the CDI solution. A natural question is whether the GA's solution would converge to the same fitness value if initial parents other than these were input, and if not, how close the two values would be. Therefore, the GA was rerun using three randomly chosen item sets as initial parents. If varying the initial parents resulted in markedly different fitness values, it would cast doubt upon the GA's ability to consistently find high-quality solutions. Such a result did not occur in this investigation, however; comparing the results for the two sets of initial parents, the associated difference in fitness was in the third decimal place (.329 using the CDI as initial parents, .331 using random initial parents). The fitness value of the CDI was .411; those of the random initial parents were .612, .690, and .727. Thus, the fitness values of the GA solutions were much closer to each other than to their starting values, suggesting that the difference in quality due to varying the initial parents is negligible.

Comparison of the GA solution with all possible item sets. As explained above, computing the fitness of all possible item sets is intractable. However, it is possible to compute the fitness of all possible item sets in a smaller item pool, and then determine the GA solution's rank with respect to this smaller pool. For this subsection, a "subpool" of 20 items was selected at random from the highly discriminating pool. The GA was then run on this subpool; its task was to select the 10 subpool items yielding the smallest average number of classification errors over the training data. Three copies of the CDI solution served as initial parents for the GA.

Both the CDI and GA found top-tier solutions within the subpool. The CDI solution had an average of 1.35 classification errors per simulee; this value ranked 11th out of the $\binom{20}{10} = 184{,}756$ possible item sets. The GA solution had an average of 1.30 classification errors per simulee, ranking first out of all possible sets. Although optimal selection within this small subpool does not guarantee optimal selection for the complete 300-item pool, it is an encouraging result nonetheless.

Summary and Discussion

Local search algorithms, though relatively new to the psychometric literature, have been successfully used in several ATA applications (Veldkamp, 1999; Verschoor, 2007). In this article, a new GA was introduced to select items for CDM. This GA circumvents potential problems arising from nonexistence of Fisher information in CDMs by utilizing fitness functions that directly relate to classification accuracy. The three particular functions defined were average number of classification errors, maximum error rate, and distance to a set of target error rates.

Several theoretical and practical benefits of using the GA are as follows:

1. In contrast to methods that select items based on individual merit, such as the CDI, the GA evaluates each item set as a unit. Such holistic selection guarantees that the chosen items complement one another. Although linear programming methods also treat item sets holistically, they are unable to directly optimize Bayesian error rates alongside CDMs, as does the GA considered in this article.
2. The GA is flexible to the practitioner's goal: it can be applied whether the fitness function is the average number of classification errors, the maximum error rate, proximity to target error rates, or any other numerical criterion.
3. The GA takes initial solutions as parents, then finds a solution that performs as well as or better than those parents. Thus, any solution can potentially be improved upon by inputting it to the GA as a parent.

We have given evidence that the GA is not sensitive to its initial parents. However, using the CDI as an initial parent carries two advantages over random starting points. First, because the GA's solution cannot be worse than its initial parents, using the CDI guarantees that the GA will never be outperformed by it. Second, because the CDI solution is generally closer to optimal than randomly determined tests (Henson & Douglas, 2005), its input as an initial parent tends to reduce the GA's computational time. Thus, our proposed procedure may be considered a sequential method where the practitioner first finds the CDI solution, then refines this solution via the GA.

The "Simulation Results" section compared this sequential GA to the CDI using simulation. The conclusions to be drawn from such simulations are specific to the fitness function studied. When the defined fitness function was average number of classification errors, the GA achieved a modest-to-significant gain in accuracy as compared to the CDI. For this fitness function, the decision of whether to use the GA or CDI depends on the cost of computation relative to the cost of classification error. The CDI may be adequate if computation is expensive; on the other hand, if classification error is considered more expensive, the GA can help reduce this cost. Turning to the results for maximum error rate, the GA achieved a substantial gain relative to the CDI under all eight conditions. Therefore, if a practitioner seeks to avoid any high attribute-level error rates, the simulations herein recommend the GA as the superior method. Finally, if the goal is to hit attribute-level target error rates, then no method based on CDI is adequate, as there is no known function relating the CDI to error rates. The GA, on the other hand, was able to hit prescribed targets with high precision. Thus, whenever target error rates are sought, GA adds an important dimension beyond what can be provided using the CDI.

Clearly, the widespread use of the GA is dependent on the ability of computers to perform the requisite tasks in an adequate amount of time. Because the GA requires the use of simulated training data to assemble tests, its computational intensity is considerably greater than that of analytic procedures like the CDI. We note, however, van der Linden's (2005) statement that "the current enormous power of our computers has stimulated the interest in local search heuristics" (p. 94). As this power continues to grow, the computational expense of the GA may be expected to shrink in comparison to its contribution to enhancing classification accuracy.

Further study includes simulations using CDMs other than the reduced RUM, as well as varying the number of attributes, discrimination of item parameters, and complexity of constraints. Robustness of the GA to a misspecified prior distribution should also be examined. Furthermore, although we have provided a preliminary analysis of the GA's proximity to the optimal solution, this analysis was limited to the condition with highly discriminating items, a uniform prior on $\boldsymbol{\alpha}$, and no constraints; moreover, the comparison of the GA with all possible item sets was limited to a small subpool. Additional investigation is required to determine general conditions when the procedure finds the optimal solution and when it does not. Finally, it is important that the CDM ATA methods developed in this paper be applied in a wide variety of real-data settings. The comparison of tests assembled by computer and by test developers will likely produce important insights that will improve the CDM ATA methods. These and other extensions of this article will be addressed in future work.

Acknowledgments

The authors are indebted to Jon-Michael Brasfield and Angela Verschoor for helpful discussions of GAs.

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* (pp. 395–479). Reading, MA: Addison-Wesley.

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

de la Torre, J., & Douglas, J. A. (2004). Higher-order latent trait models for cognitive diagnosis. *Psychometrika*, *69*, 333–353.

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

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. (2008). Cognitive diagnostic attribute-level discrimination indices. *Applied Psychological Measurement*, *32*, 275–288.

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

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, Urbana-Champaign.

Jang, E. E. (2006). *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.

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.

Nichols, P. D., Chipman, S. F., & Brennan, R. L. (1995). *Cognitively diagnostic assessment*. Hillsdale, NJ: Erlbaum.

Reckase, M. D. (1985). The difficulty of test items that measure more than one ability. *Applied Psychological Measurement*, *9*, 401–412.

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). *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.

Roussos, L. A., Templin, J. L., & Henson, R. A. (2007). Skills diagnosis using IRT-based latent class models. *Journal of Educational Measurement*, *44*, 293–311.

Sinharay, S. (2006). Model diagnostics for Bayesian networks. *Journal of Educational and Behavioral Statistics*, *31*, 1–33.

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

Sun, K.-T., Chen, Y.-J., Tsai, S.-Y., & Cheng, C.-F. (2008). Creating IRT-based parallel test forms using the genetic algorithm method. *Applied Measurement in Education*, *21*, 141–161.

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

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. (1999). Multiple objective test assembly problems. *Journal of Educational Measurement*, *36*, 253–266.

Verschoor, A. J. (2007). *Genetic algorithms for automated test assembly*. Unpublished 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.

Authors

MATTHEW FINKELMAN is an Assistant Professor, Tufts University School of Dental Medicine, 75 Kneeland Street, Boston, MA 02111; matthew.finkelman@tufts.edu. His primary research interests include computerized adaptive testing, test assembly, and biostatistics.

WONSUK KIM is a Psychometrician I, Measured Progress, 100 Education Way, Dover, NH 03820; kim.wonsuk@measuredprogress.org. His primary research interests include equating and test assembly.

LOUIS A. ROUSSOS is a Psychometrician II, Measured Progress, 100 Education Way, Dover, NH 03820; roussos.louis@measuredprogress.org. His primary research interests include cognitive diagnosis models, dimensionality analysis, and differential item functioning.