LSAC RESEARCH REPORT SERIES

- **Stochastic Programming for Individualized Test Assembly With Mixture Response Time Models**

Bernard P. Veldkamp
Marianna Avetisyan
University of Twente, Enschede, the Netherlands

- **Law School Admission Council**
Research Report 15-01
March 2015

A Publication of the Law School Admission Council

The Law School Admission Council (LSAC) is a nonprofit corporation that provides unique, state-of-the-art products and services to ease the admission process for law schools and their applicants worldwide. Currently, 222 law schools in the United States, Canada, and Australia are members of the Council and benefit from LSAC's services. All law schools approved by the American Bar Association are LSAC members. Canadian law schools recognized by a provincial or territorial law society or government agency are also members. Accredited law schools outside of the United States and Canada are eligible for membership at the discretion of the LSAC Board of Trustees; Melbourne Law School, the University of Melbourne is the first LSAC-member law school outside of North America. Many nonmember schools also take advantage of LSAC's services. For all users, LSAC strives to provide the highest quality of products, services, and customer service.

Founded in 1947, the Council is best known for administering the Law School Admission Test (LSAT®), with about 100,000 tests administered annually at testing centers worldwide. LSAC also processes academic credentials for an average of 60,000 law school applicants annually, provides essential software and information for admission offices and applicants, conducts educational conferences for law school professionals and prelaw advisors, sponsors and publishes research, funds diversity and other outreach grant programs, and publishes LSAT preparation books and law school guides, among many other services. LSAC electronic applications account for 98 percent of all applications to ABA-approved law schools.

This study is published and distributed by LSAC. The opinions and conclusions contained in this report are those of the author(s) and do not necessarily reflect the position or policy of LSAC.

Table of Contents

Executive Summary 1

Introduction 1

Mixture Models 3
- RT Modeling 4
- Mixture RT Models 4
- Implications of Mixture RT Modeling 6

Test Assembly 7
- RT Constraints 9
- Constraints for Mixture RT Models 9

Stochastic Programming 11

Simulation Study 13
- Item Bank 13
- Settings of the Study 15

Conclusion and Discussion 16

References 18

Executive Summary

Many standardized tests are now administered via computer rather than paper and pencil. The computer-based delivery mode brings with it certain advantages, such as the ability to record not only the test taker's response to each item (i.e., question), but also the amount of time the test taker spends considering and answering each item. The analysis of response times (RTs) is still a developing area of research.

Early RT research assumed that a test taker would show consistent RTs over the course of a test. Such models may be unrealistic for various reasons—some items require more time than others to answer, a warm-up effect may cause a test taker to respond more quickly after completing the early items, fatigue may cause a test taker to slow down toward the end of a test, or as time runs out the test taker may quickly guess the answers to the last items on a test. To take these variable RTs into account, mixture RT models have recently been investigated.

Until now, mixture RT models have only been applied for post hoc analyses. This research expands the use of these models by exploring their application in the context of test assembly. Various strategies were applied and the strengths and weaknesses of each described. In general, it was concluded that the application of mixture RT models should prove especially useful for tests with a heterogeneous testing population.

Introduction

Computerized test administration is becoming more and more popular in education measurement. One of the advantages is that the actual response behavior of test takers can be recorded in log files. Next to the actual response, log files also provide information about response times (RTs), response strategies, the order in which the items were answered, revised answers, and the use of auxiliary materials. Generally, it remains quite a task to extract useful information from log files (He & von Davier, 2014; Timmers, Walraven, & Veldkamp, 2014). Log files contain raw data about mouse clicks and key strokes that requires interpretation. RTs, however, are rather straightforward to extract; when only one item is presented at a time, log files provide accurate RT information at the item level. The usefulness of RTs has been well demonstrated in the literature (e.g., Hornke, 1997, Masters & Keeves, 1999, Schnipke & Scrams, 1997).

RTs have been used for various kinds of analyses. First, they provide information about the average speed of working, the speededness of a test toward the end, warming-up effects, and fatigue (Ackerman & Kanfer, 2009; Evans & Reilly, 1973; Lawrence, 1993; van der Linden, 2011). For example, long RTs at the beginning of a test, often in combination with relatively many mistakes, may be an indication of a warming-up effect. Short RTs toward the end of a test, often in combination with a high number of mistakes, are an indication that the test may be speeded. Long RTs and relatively many mistakes toward the end of the test may indicate fatigue. Recently, Lee and Jia (2014) applied RTs to analyze test-taking behavior in large-scale assessments.

Second, RTs have been used for item analysis. Fan, Wang, Chang, and Douglas (2012) introduced information per time unit as a new index for item selection in computerized adaptive testing (CAT). When the total RT for a test is restricted, selecting items based on maximum Fisher information might not be the most efficient approach. Imagine an item bank where the most informative item provides 5% more information than any other item but is three times more time-consuming to answer. In that case, it might pay to administer a larger number of items with shorter RTs that together provide more information than the most informative item. RTs also reveal which items are more sensitive to working speed than others (Marianti, Fox, Avetisyan, & Veldkamp, 2014). In order to prevent differential speededness in high-stakes testing situations, this sensitivity of items to working speed is very useful in the test development process. Recently, Finkelman, Kim, Weissman, and Cook (2014) published a paper on item selection for cognitive diagnostic models and CAT in which RTs were taken into account.

Finally, RTs have been used to identify aberrant response behavior such as cheating (van der Linden & van Krimpen-Stoop, 2003). Unexpected correct answers to relatively difficult items combined with very short RTs are generally seen as a strong indication of cheating. Van der Linden and Guo (2008) mention three reasons why RTs are a strong source of information about aberrant response behavior: (a) they are very suitable for statistical testing because they are continuous variables; (b) in CAT, it remains possible to distinguish likely from unlikely RT patterns; and (c) even when test takers try to simulate realistic RTs, it is almost impossible for them to find out what a typical RT pattern would be at their ability level. Posterior predictive RT distributions (van der Linden & Guo, 2008) or modified versions of the standardized likelihood-based person-fit statistic l_0 (Levine & Rubin, 1979) can be applied to detect aberrant RTs (Marianti et al., 2014).

Various RT models have been presented in the literature. The first category of RT models focuses on the RTs without taking the correctness of the response into account (e.g., Maris, 1993; Schnipke & Scrams, 1997). The second category focuses on both RTs and accuracy. Van der Linden (2006, 2007) introduced a hierarchical framework for modeling both speed and accuracy concurrently. In this framework, a normal ogive model is formulated for dealing with the responses, a lognormal model is chosen for the RTs, and a bivariate normal distribution is chosen to model the joint distribution of both person and item parameters.

One of the assumptions in van der Linden's model is that test takers work at uniform speed during test administration (van der Linden, 2009). In practice, this assumption might not hold. Marianti et al. (2014), Molenaar and De Boeck (2014), and Fox (2014) proposed using a mixture of response models or a more dynamic RT model to describe the RT behavior of test takers. They assumed that the working speed of a test taker varies during the test, and that it is related to test-taking strategy. Marianti et al. (2014) presented an example where some test takers showed aberrant response behavior, such as cheating. Molenaar and De Boeck (2014) studied the case where test takers used different more or less efficient strategies for solving the items, alternating among various strategies during the test (see Chen & De Boeck, 2014; Partchev & De Boeck, 2012). Finally, Fox (2014) introduced a dynamic model that accounted for the behavior

where test takers increased or decreased their working speed during test administration.

Even though the development of these mixture or dynamic RT models is still in its infancy, these models seem to fit the data quite well. The next question is how to apply these models in operational settings (e.g., in individualized test assembly such as multistage testing or CAT, or when individual linear test forms are assembled on the fly for every candidate, without knowing a candidate's proficiency in advance). Fan et al. (2012), van der Linden, Scrams, and Schnipke (1999), van der Linden (2011), and Veldkamp (2014) propose different models for taking RTs into account during test assembly. They illustrate how RTs can be used to adapt item selection to the working speed of test takers in order to prevent speededness issues toward the end of the test. All these papers assume a uniform test-taker working speed.

The present report focuses on how to assemble tests using mixture RT models. First, the more general RT models are introduced in more detail. Then, a test assembly model is presented. Even though the methodology is applicable to CAT, we will focus on the assembly of linear test forms first, especially since the assembly of CATs can be seen as solving a series of linear test assembly problems when the shadow test approach (van der Linden & Reese, 1998) is applied. Stochastic programming techniques are introduced for solving this test assembly model. In a numerical example, application of the method is illustrated and evaluated. Recommendations about its use are given. Finally, a generalization of these techniques to more general test assembly problems is discussed.

Mixture Models

Mixture models have been applied in both educational and psychological measurement to account for different response behavior by various groups of test takers in the population (Hancock & Samuelsen, 2008). These groups are also referred to as latent classes, since the class to which a test taker belongs cannot be observed directly. Test takers who behave more like each other than like other test takers in the population are categorized into classes, which are identified via statistical methods. Several examples can be found in the literature. Von Davier and Yamamoto (2004) applied mixture modeling to account for different school types in a mathematics assessment. Mixture item response theory (IRT) models were applied for explanatory differential item functioning analysis by Cohen and Bolt (2005) and by Cho and Cohen (2010). Egberink, Meijer, and Veldkamp (2010) applied mixture IRT modeling to a conscientiousness scale and found that this construct is qualitatively different for different groups of test takers, which influenced their response style. The software packages Mplus (Muthén & Muthén, 2012) and Latent Gold (Vermunt & Magidson, 2013) are generally applied for the analysis of mixture models.

RT Modeling

In this report, the lognormal RT model of van der Linden (2006) is applied. This means that the logarithm of the RTs is assumed to be normally distributed. Before we focus on mixture RT modeling, this RT model is described and algorithms for estimating the model are mentioned. In the lognormal model, the RT distribution can be characterized by the working speed parameter ς_p, a time-intensity parameter λ_i, and a time-discrimination parameter ϕ_i. For a given person p, the working speed is assumed to be constant during the test. The time-intensity parameter is a measure of the time needed to complete the item, and the time-discrimination parameter is a measure of the sensitivity of the item to differences in working speed between test takers. Since the response behavior might vary due to distraction, tiredness, or other causes, and because these deviations are assumed to be independent of working speed, a normally distributed measurement error component is added to the model, with mean equal to zero and variance equal to σ_i^2. When the observed RTs of person p to item i are denoted by T_{ip}, the lognormal RT model can be formulated as

$$p(t_{ip} \mid \varsigma_p, \phi_i, \lambda_i, \sigma_i^2) = \frac{1}{t_{ip}\sqrt{2\pi\sigma_i^2}} \exp\left[-\frac{1}{2\sigma_i^2}\left(\ln t_{ij} - \phi_i(\lambda_i - \varsigma_p)\right)^2 \right]. \quad (1)$$

Following Marianti et al. (2014), this RT model deviates slightly from the model in van der Linden (2006), since a time-discrimination parameter has been introduced. Van der Linden (2007) proposed a hierarchical framework for modeling both response correctness and RTs concurrently. A normal ogive model can be applied for modeling correctness and a lognormal model for RTs. To model the joint distribution of both person and item parameters, a bivariate normal distribution is assumed. Bayesian estimation procedures can be applied to estimate the model. Technical details about the Markov chain Monte Carlo (MCMC) methods, the specifications of prior distributions for the parameters, and the full conditional distributions of the model parameters can be found in Fox, Klein Entink, and van der Linden (2007), Klein Entink, Fox, and van der Linden (2009), and van der Linden (2007).

Mixture RT Models

In mixture RT models, a multicomponent distribution can be defined to account for differences in the RT behavior of various classes of test takers. Several examples of mixture RT modeling can be found in the literature.

Between-Subjects Latent Class Model

Marianti et al. (2014) describe a distinction between a class of test takers who behave according to the RT model and a class of test takers with aberrant behavior. For this mixture RT model, it holds that the response behavior of the test takers can be

classified into a number of latent classes A_0, …, A_K, where classes A_k, $k = 1,...,K$, are mutually exclusive and exhaustive. The probability of a person p's membership in a latent class A_k is known and denoted as $P(p \in A_k)$, a person can only be a member of one class, and the actual membership is a priori unknown.

According to the specifications of the mixture distribution, the RTs can be modeled as

$$p(T_{ip} = t_{ip}) = \sum_k p\left(t_{ip} \left| \zeta_p^{(k)}, \lambda_i^{(k)}, \phi_i^{(k)}, \sigma_i^{(k)2}, A_k \right.\right) P(t_{ip} \in A_k), \quad (2)$$

where for each class k a lognormal RT model (see Equation (1)) is estimated, and $\zeta_p^{(k)}$, $\lambda_i^{(k)}$, $\phi_i^{(k)}$, and $\sigma_i^{(k)}$ are the respective person and item parameters for the RT model of latent class k. It is even possible that different model formulations can be used for the various classes. For example, in Marianti et al. (2014), a lognormal RT model is used to describe the behavior of latent class A_0, because the class represents regular behavior, and the RT model for latent class A_1 is a generic probability model, since it describes all possible aberrant RT behaviors.

Within-Subjects Latent Class Model

Molenaar and De Boeck (2014) chose a rather different approach. First, they assumed that the lognormal RT model in (1) can be used to describe the RTs. Then, they distinguished between fast and slow response behavior, where a test taker is allowed to alternate between fast and slow response behavior depending on whether the test taker behaves according to his or her fast ability $\theta_p^{(f)}$ or slow ability $\theta_p^{(s)}$. For each of the abilities, a separate measurement model can be defined:

$$\text{logit}\left[P_i(X_{ip} = 1 | \theta_p^{(s)}\right] = a_i^{(s)}\left(\theta_p^{(s)} - b_i^{(s)}\right), \quad (3)$$

and

$$\text{logit}\left[P_i(X_{ip} = 1 | \theta_p^{(f)}\right] = a_i^{(f)}\left(\theta_p^{(f)} - b_i^{(f)}\right), \quad (4)$$

where $a_i^{(s)}$ and $a_i^{(f)}$ denote the respective discrimination parameters, and $b_i^{(s)}$ and $b_i^{(f)}$ denote the respective difficulty parameters. The probability of a correct answer can now be modeled as

$$P\left(X_{ip} = 1 | \theta_p^{(s)}, \theta_p^{(f)}\right) = \pi_{ip} P\left(X_{ip} = 1 | \theta_p^{(s)}\right) + (1 - \pi_{ip}) P\left(X_{ip} = 1 | \theta_p^{(f)}\right), \quad (5)$$

where π_{ip} denotes the probability that the test taker p answers item i using slow response behavior. In this model, the probabilities π_{ip} are made dependent on the RTs, while accounting for the main effects of items and persons on the RT distribution. In contrast, in Marianti et al. (2014), test takers can only be a member of one class, each individual response is assigned to one of two classes, and class membership is a priori unknown. Using the Block Design subtest from the Wechsler Intelligence Scale for Children IV (WISC-IV; Wechsler, 2003), Molenaar and De Boeck (2014) were able to explain the observed differences between the fast and slow responses of individual test takers.

Dynamic Factor Model

The Dynamic Factor Model for stochastic speed processes is described in Fox (2014). In this model, a test is assumed to consist of a number of blocks of items, each having its own average block working speed. Items in a block can be consecutive or spread out over the test. The block working speeds are assumed to follow a time trend. In this way, RT models can account for variable speed processes. For example, test takers may increase their working speed during a test when they run out of time toward the end of the test, or they may decrease it and work more slowly toward the end of the test due to fatigue. The main advantage of the Dynamic Factor Model is its flexibility in dealing with different kinds of nonstationary RT behavior of the test takers.

Implications of Mixture RT Modeling

In each of the examples above, researchers proposed using a mixture RT model to deal with unobserved heterogeneity in the RTs of the test takers. The mixture RT models allowed for the investigation of groups of test takers who showed nonstationary RT behavior over the course of a test. Application of mixture RT models to real datasets not only led to more precise measurement, but also enabled the researchers to interpret the observed RTs, which increased the validity of the test.

Until now, mixture RT models were only applied for post hoc analyses. A next step would be to use them during test development or during test administration. For example, when pretesting or previous experience has revealed that a significant number of test takers have shown nonstationary RT behavior due to warming-up effects, speededness toward the end of the test, or fatigue, this might be taken into account in test development. Another example relates to CAT, where the application of mixture RT models might reveal aberrant test-taker behavior. If cheating is suspected, immediate actions might be taken before the remaining items are administered (e.g., selecting items from a secret back-up pool of previously unadministered items that have been put aside for such situations).

The next section of this report concerns how to assemble tests when mixture RT models have been used to calibrate an item bank. First a general model for test assembly is presented. The modifications that are needed to apply mixture RT models are then described and the implications for automated test assembly (ATA) discussed.

Test Assembly

In ATA, 0-1 linear programming (LP) methods are generally applied for item selection. Van der Linden (2005) presented a general 0-1 LP model for the assembly of linear test forms. In this model, test specifications are modeled as constraints, and the objective function represents one of the attributes of the test that must be optimized in test development. The constraints of the model can be categorized as categorical, quantitative, and logical. Categorical constraints concern attributes of the items that categorize the item bank, such as content classification of the items or item type. Quantitative constraints concern attributes that have quantitative values, such as word count; RT constraints fall under this class of specifications. Finally, logical constraints deal with dependences between pairs or groups of items, such as sets containing enemy pairs, where one item provides clues for solving the other item, or item sets where multiple items are related to the same stimulus. The most common objective functions are maximization of test information, minimization of the deviation between the test information function and a prespecified target, and minimization of the number of items.

Let

$i = 1,...,I$ be an index for the items in the bank

x_i represent whether an item is selected or not

$I_i(\theta_p)$ be the amount of information provided by item i for person p with ability level θ_p

$c = 1,...,C$ be an index for the categories

b_c be the maximum number of items that can be selected for category c

q_i denote the contribution of item i to quantitative attribute $q = 1,...,Q$

b_q be the upper bound for attribute q

$e = 1,...,E$ be an index for the various logical constraints

b_e be the maximum number of items to be selected for this group. For example, for an enemy constraint, this number is equal to one; for item sets, it is equal to the maximum number of items that can be selected from an item set.

The model can now be formulated as:

$$\max \sum_{i=1}^{I} I_i(\theta_p) x_i \tag{6}$$

subject to

$$\sum_{i \in C} x_i \leq b_c, \quad c = 1,...,C, \tag{7}$$

$$\sum_{i=1}^{I} q_i x_i \leq b_q, \quad q = 1, ..., Q, \tag{8}$$

$$\sum_{i \in e} x_i \leq b_e, \quad e = 1, ..., E, \tag{9}$$

$$x_i \in \{0,1\}. \tag{10}$$

This can be seen as a general formulation of a test assembly model, since any minimization objective function can be reformulated as a maximization objective function. Besides, any lower bound can be reformulated as an upper bound by adding minus signs to both sides of the constraints. Finally, equality constraints can also be formulated as upper bound constraints, since $'='$ implies that both $'\leq'$ and $'\geq'$ hold. This general test assembly model can easily be modified and extended to be applicable for the assembly of multistage tests, CATs with constraints, tests measuring multiple traits, mastery tests, or even test batteries. For an overview of test assembly models, see van der Linden (2005).

RT Constraints

Specifications related to RTs can be formulated to ensure that test takers can finish the test within the allotted time slot. Since working speed varies across test takers, and since some test takers have a tendency to postpone responding to a question and to wait for some special insight to occur when they don't know the correct answer, most testing agencies apply specifications such as (a) 100% of the test takers must be able to respond to 90% of the items, or (b) 85% of the test takers must be able to respond to all of the items. Besides, testing agencies might want to prevent differential speededness whereby, because of differences in working speed, some test takers might run out of time while others can really demonstrate their ability and finish the test without time pressure. Differential speededness may be an issue, for example, in CAT or multistage testing, where more capable test takers have to respond to more difficult, and often more time-intensive, items (van der Linden, 2006).

In test assembly, RT constraints are modeled in terms of expected RTs. For the lognormal RT model, the expected RT for item i of a test taker with working speed ς_p is equal to

$$E(t_{ip} \mid \varsigma_p) = \exp\left(\lambda_i - \varsigma_p + \frac{1}{2\phi_i^2} \right). \tag{11}$$

Let $E[T_{ip}]$ denote the expected RT for item i and test taker p, and let $T_{\max}$ be an upper bound for the available time. A generic formulation for RT constraints would be:

$$\sum_{i=1}^{I} E[T_{ip}]x_i \leq T_{\max}, \quad \forall p. \tag{12}$$

In this formulation, the RT constraint holds for $\forall p$ (i.e., for all test takers). By varying either the percentage of test takers or the percentage of items for which this constraint holds, this generic constraint can be applied to model most of the RT specifications encountered in practice.

In CAT or multistage testing, information about working speed can be gathered during test administration. After administering a number of items in CAT or one of the stages in multistage testing, van der Linden (2006) showed that for a lognormal RT model where the time-discrimination parameters are assumed to be equal, the working speed of the test taker can be estimated using the following formula:

$$\varsigma_p = \frac{\sum_{i \in R_g} \phi_i^2 \left(\lambda_i - \log t_{ip}\right)}{\sum_{i \in R_g} \phi_i^2}, \tag{13}$$

where R_g denotes the set of g items that have been administered so far. For administering the remaining items in the test, the expected RTs $E[T_{ip}]$ can be calculated more precisely, based on the estimated working speed $E(t_{ip} \mid \hat{\varsigma}_p)$, and the generic RT constraint can be modified to account for the time used for the first g items:

$$\sum_{i \in R_g} t_{ip} + \sum_{i \in I \setminus R_g} E[T_{ip}]x_i \leq T_{\max}, \tag{14}$$

where the second summation is over those items that have not be selected in the first g iterations of the CAT. See also van der Linden et al. (1999).

Constraints for Mixture RT Models

When mixture RT models are applied, formulation of RT constraints becomes slightly more complicated. Instead of one lognormal RT model that holds for all test takers, a mixture of models must be taken into account. The expected RT $E[T_{ip}]$ can now be calculated as:

$$E(t_{ip} \mid \varsigma_p^{(1)}, \ldots, \varsigma_p^{(K)}) = \sum_k \exp\left(\lambda_i^{(k)} - \varsigma_p^{(k)} + \frac{1}{2\phi_i^{(k)2}}\right) P\left(t_{ip} \in A_k\right). \tag{15}$$

This implies that, instead of a single RT constraint, a mixture of RT constraints is defined:

$$\sum_{i=1}^{I}\left[\sum_{k}\exp\left(\lambda_i^{(k)}-\varsigma_p^{(k)}+\frac{1}{2\phi_i^{(k)2}}\right)P\left(t_{ip}\in A_k\right)\right]x_i\le T_{\max},\quad \forall p. \tag{16}$$

Unfortunately, these probabilistic constraints cannot be handled by regular 0-1 LP methods directly. These methods have been developed to deal with deterministic objective functions and constraints, which are linear functions of the decision variables x_i. As a consequence, alternative test assembly methods must be applied.

Several strategies for dealing with probabilistic optimization problems are available (Birge & Louveaux, 1997). First, a probabilistic mixture RT constraint can be reformulated into a deterministic one by using the average RT over all classes. The resulting constraint can now be formulated as:

$$\sum_{i=1}^{I}\overline{E[T_{ip}]}x_i\le T_{\max},\quad \forall p, \tag{17}$$

where $\overline{E[T_{ip}]}$ denotes the average expected RT for person p over all RT classes A_k. A drawback of this strategy is that a violation of probabilistic constraints is accepted for part of the population. To prevent these violations, a much more conservative reformulation of the model can be applied. The probabilistic constraint can be replaced by a series of deterministic constraints:

$$\sum_{i=1}^{I}\exp\left(\lambda_i^{(k)}-\varsigma_p^{(k)}+\frac{1}{2\phi_i^{(k)2}}\right)x_i\le T_{\max},\quad \forall p,\forall k. \tag{18}$$

Unfortunately, one needs k times as many constraints in this approach. Besides, since the RT classes don't overlap, this strategy would severely overconstrain the problem. Finally, Bertsimas and Sim (2003) proposed robust optimization. They argued that maximum uncertainty only impacts a final solution of any optimization problem for a limited number of items. In this method, a model with uncertainty in it is reformulated into a series of deterministic optimization problems.

Veldkamp (2013) described the application of robust optimization to ATA problems. In robust optimization, the average expected RTs $\overline{E[T_{ip}]}$, the maximum expected RT over all classes $E[T_{ip}]^{\max}$, and the differences d_i between them must be calculated first for each item. Then, the items must be ranked based on their contribution to the objective function (6). Let Γ be the number of items for which uncertainty affects the solution. For most test assembly problems, Γ can be set equal to 40% of the test length. Next, a series of l optimization problems, where $l=1,...,\text{test length}$ must be solved. In these problems, the following RT constraint is imposed:

$$\sum_{i=1}^{I}\overline{E[T_{ip}]}x_i+\left[\sum_{i=1}^{l}\left(d_i-d_l^*\right)x_i+\Gamma d_l^*\right]\le T_{\max},\quad \forall p, \tag{19}$$

where $d_l^* = \min_{i<l}\{d_i\}$. Finally, the best solution of these l problems is chosen. Even though the uncertainties in the model are taken into account during optimization, a solution that is too conservative is prevented. The only drawback of this strategy is that a series of l optimization problems must be solved instead of one. For a detailed description and an analysis of the performance, the reader is referred to Bertsimas and Sim (2003).

What all of these strategies have in common is that they reformulate the model such that standard 0-1 LP software can be applied for solving the problem. However, reformulating the model comes at a cost. The final solution either violates the constraints, is far too conservative, or is far more time-consuming to obtain. In the next section, stochastic programming is introduced for dealing with mixture RT constraints.

Stochastic Programming

Stochastic linear programming deals with problems with random constraints (Klein Haneveld & van der Vlerk, 2006):

$$\max \sum_{i=1}^{I} c_i x_i \tag{20}$$

such that

$$\sum_{i=1}^{I} T(\omega)_i x_i \leq h(\omega), \tag{21}$$

$$\sum_{i=1}^{I} a_{ij} x_i \leq b_j, \tag{22}$$

$$x_i \in \{0,1\}, \quad i = 1,...,I, \tag{23}$$

where the actual value of ω (where ω might refer to, for example, the RT class to which the test taker belongs) is unknown. Only probabilistic information about ω is available; that is, we assume that the distribution of ω is given. This model must be interpreted in the following way. In the first stage, we must decide on the first-stage variables x_i, without any information about the realization of ω available. However, this solution will often be infeasible with respect to the second-stage specifications in (21) (Klein Haneveld & van der Vlerk, 1999). In our simulation study, settings from an operational computerized high-stakes test are used, where for each test taker a new linear test form is assembled from an item bank. These test forms have to meet a set of test specifications. For this test, a mixture RT model as described in (2) can be applied, and the total testing time is restricted. When the test is assembled, the RT classes to which the test taker belongs is unknown. So although ω is unknown, we do have a

distribution of class membership for the whole population of test takers. In the first stage, we have to decide on variables x_i, denoting whether or not item i is selected for the test, without any information about the RT class membership of the individual test taker. *Note*: Since the total test time is identical for all test takers, ω only plays a role in the left-hand side of (21) for the example in the simulation study.

Two classical strategies for dealing with stochastic programming problems are available:

1. Penalty costs are assigned to violations of the constraints in (21). In this strategy, the objective function of the model in (20) is extended with a penalty function consisting of an expected violation of the probabilistic constraint multiplied by a cost parameter. Recourse actions are then taken to compensate for the infeasibility. Therefore, such strategies are also referred to as *recourse models* (Birge & Louveaux, 1997).
2. The second strategy is to specify a model with *chance constraints* (Birge & Louveaux, 1997). In these constraints, the probability that any of the constraints in (21) is violated is restricted:

$$P\left(\sum_{i=1}^{I} T(\omega)_i x_i > h(\omega)\right) \leq \alpha, \tag{24}$$

where α limits the probability of a violation. Imposing these constraints implies that a solution for the problem in (20)–(23) is accepted only if it is not too risky.

The second strategy seems more appropriate to apply to mixture RT modeling. As was also mentioned in the introduction section, RT specifications are often formulated relative to the population. Generally, the maximum amount of time to finish a test is limited for practical reasons. Within this time limit, a prespecified percentage of test takers must be able to complete the whole test. At the individual level, this means that the probability of not finishing the test in time must be limited, which is exactly what chance constraints intend to model.

The introduction of chance constraints to an optimization model introduces some technical difficulties. Chance constraints are nonconvex in general, especially when ω is discrete in nature, which is the case in mixture RT modeling where ω refers to class membership, and classes are assumed to be mutually exclusive and exhaustive. Therefore, Klein Haneveld and van der Vlerk (2002) proposed modeling them as *integrated chance constraints,* where the uncertainty in ω is integrated out of the constraint*:*

$$E\left[\sum_{i=1}^{I} T(\omega)_i x_i - h(\omega) \geq 0\right] \leq \beta, \tag{25}$$

where β represents the largest acceptable expected violation, and it is specified a priori by the test developer. Given that RT distributions are known for the various classes,

values for β can be derived. For cases where the RT distribution for one of the classes is unspecified (Marianti et al., 2014), a decision must be made as to whether and how a bound should be imposed.

Simulation Study

A simulation study was carried out to illustrate the use of stochastic programming, and to test whether and when stochastic programming would be beneficial in ATA.

Item Bank

For this example, real items from a Basic Safety Exam were used to generate the item bank. This exam consists of 40 knowledge items, and it is obligatory for all personnel of petrochemical factories in the Netherlands. Annually, thousands of test takers participate in this exam. The exam is administered both on paper and digitally, so detailed information about actual RTs is available. In the digital environment, an individual linear test form is assembled for each candidate using stratified random sampling from an item bank of 1,700 items. The R package LNIRT (Marianti, 2015) was used to estimate the RT parameters. The MIRT package (Glas, 2010) was used to calibrate the IRT parameters, with a two-parameter logistic model. We were not allowed to publish the real item parameters for this test, but based on the real item parameter estimates and their distributions, we simulated a pool of 640 items. The test is administered in over ten different languages, and each test taker can take the test in his or her first language. The item parameters of these versions are more or less comparable (in the exam they are assumed to be equal for all languages), but some of the languages are more time-intensive than others. Therefore we simulated a mixture RT model where we distinguished between a fast (85% of the population) and a slow (15% of the population) working speed. We used the same item parameters for both classes, $a \in [0.29, 1.12]$ and $b \in [-2.60, -0.48]$; and a different RT parameter for fast response behavior, $\phi_f \in [0.66, 1.42]$ and $\lambda_f \in [3.02, 3.69]$, and for slow response behavior, $\phi_s \in [0.67, 1.41]$ and $\lambda_s \in [3.4, 3.91]$. Figure 1 shows the distribution of both the item and the RT parameters of the simulated item bank.

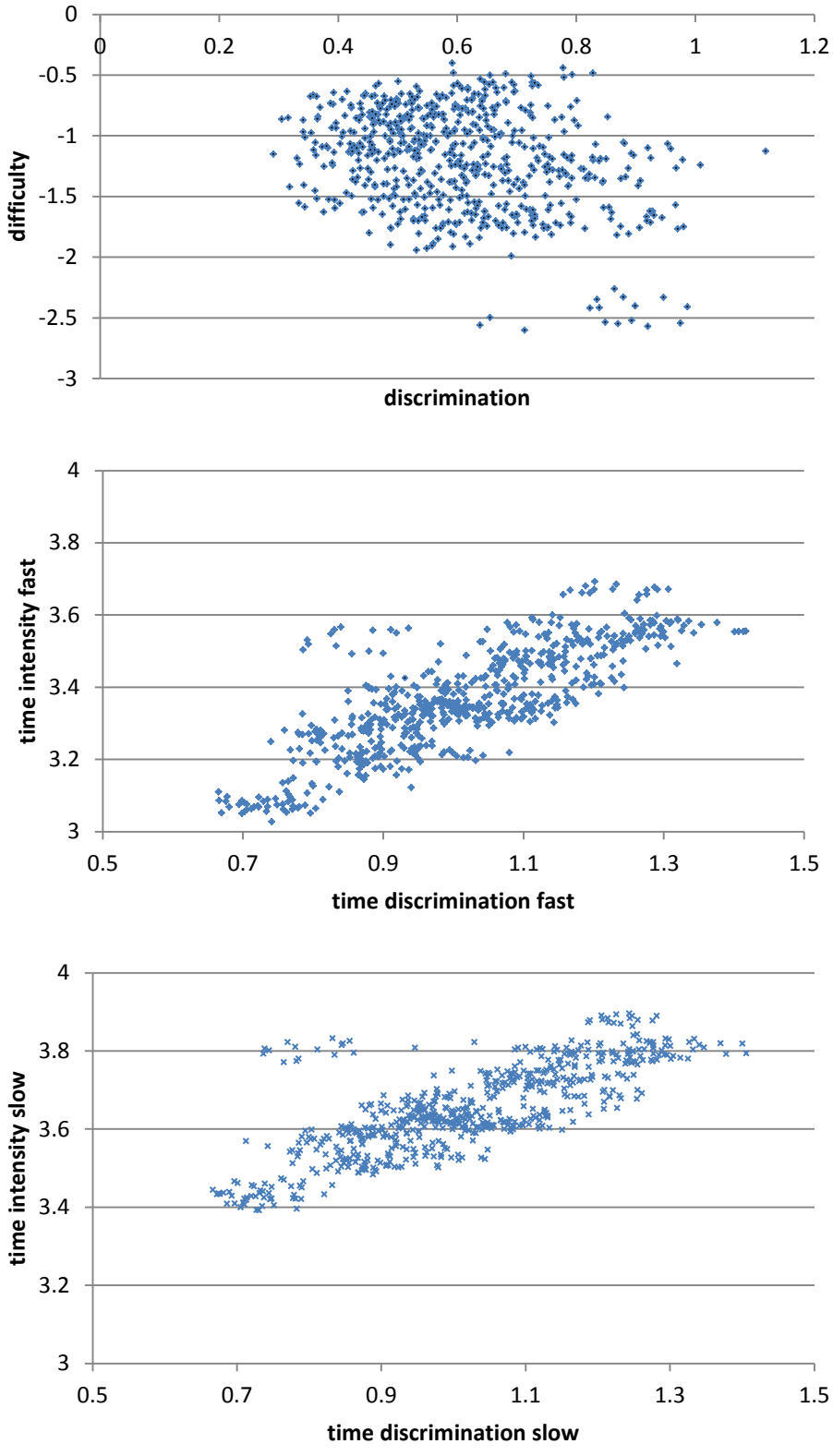

FIGURE 1. *Item and RT parameters of the simulated item bank*

Settings of the Study

In the simulation study, test length was set equal to 40 items, and the maximum expected RT was set equal to 1,200 seconds. Test information was maximized for $\theta \in \{-2, -1.5, ..., 2\}$. A maximin approach (van der Linden & Boekkooi-Timminga, 1989) was applied to formulate the test assembly model. With this approach, the objective function of the test assembly problem can be formulated as:

$$\max \min_{\theta_p \in \{-2,-1.5,...,2\}} \sum_{i=1}^{I} I_i(\theta_p) x_i. \quad (26)$$

No weighting of various ability points was used. Software R (CRAN, 2014) with the lpsolve package was used in this simulation study for all simulation conditions.

Given these settings, a test was assembled using various strategies. The RT constraint for a mixed RT model distinguishing two classes of responses was modeled either based on average RTs (see Equation (17)), via a series of deterministic constraints (see Equation (18)), as a robust optimization problem (see Equation (19)), or by using stochastic programming (see Equation (25)). Given the settings of this study and the difference in time intensity between both classes of test takers, parameter β in Equation (25) was set equal to 60. There is no actual method available for selecting this parameter. In this simulation study, we compared the average time-intensity parameters for both the fast and slow test takers, and we calculated the total RT for both classes for a test consisting of 40 average items. It turned out that slow test takers needed 400 seconds more (1,550 seconds compared to 1,150 seconds). Given a prevalence of slow test takers of 15% and taking into account that this difference is based on average time intensities instead of the actual time intensities of the selected items, we set the parameter β equal to 60. It should be mentioned that a different value could have be chosen as well.

The resulting test information functions are shown in Figure 2. For the interval $[-2, 2]$, the minimum test information was maximized. With this implementation of the maximin method, the ability value $\theta = 2$ turns out to be most critical. This is in line with our expectations, because the item bank was generated with data from an exam where the cutoff point for the pass/fail decision is close to the ability value $(\theta = -1.5)$. With respect to the various strategies, it can be seen that the second strategy, where the probabilistic constraint is replaced by a series of deterministic constraints, is much more conservative than the others. The robust ATA strategy is slightly more conservative than the strategy based on expected RTs. The stochastic programming strategy provides the most informative test in this example.

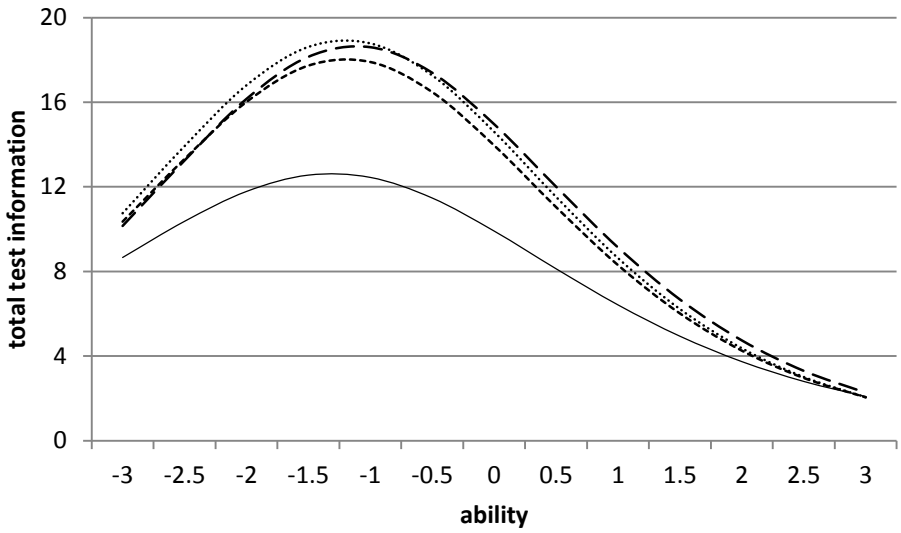

FIGURE 2. *Test information function for the various strategies: expected RTs (dotted line), series of constraints (solid line), robust ATA (short dashed line), stochastic programming (long dashed line)*

Besides comparing the strategies with respect to the test information curves, the item overlap between various test assembly strategies was also calculated. The resulting overlap is shown in Table 1.

TABLE 1
Overlapping number of items for various test assembly strategies

Strategy	Series Constraints	Robust ATA	Stochastic
Expected RT	8	32	34
Series Constraints		10	5
Robust ATA			27

As can be seen, the overlap in items between the Series Constraints strategy (where the probabilistic constraint is replaced by two deterministic constraints) and the other strategies is very small. The test resulting from this strategy differs from all the other tests by more than 75% of the items. The difference between the other strategies is smaller. For example, tests assembled with the stochastic programming strategy and the expected RT strategy have 34 of 40 items in common. Differences between the robust ATA strategy and the stochastic programming strategy are larger: they only have 27 of 40 items in common.

Conclusion and Discussion

The purpose of this report was to introduce a new method for dealing with mixture RT constraints in ATA. Mixture RT constraints fall under the category of probabilistic constraints. During test assembly, the distributions of the item parameters are known, but the RT parameters depend on the class membership of the test taker, which is a priori unknown. Nowadays, 0-1 LP methods are generally applied for solving test

assembly problems. But these methods can only deal with deterministic constraints, where the contribution of an item to a constraint is fixed. Three strategies for reformulating deterministic alternatives to probabilistic optimization problems were introduced. A stochastic programming strategy was also described. The simulation study revealed that (a) replacing a mixture RT constraint with a series of RT constraints was far too conservative; (b) the robust ATA method performed only slightly worse than the method where the expected value of the RTs over the various classes was restricted; and (c) stochastic programming performed best for the test assembly problem in the example.

Implementing the stochastic programming method turned out to be rather straightforward when integrated chance constraints were applied. The only complicated part was that upper bounds β had to be derived for the chance constraints in (25). In our study, both response classes differed only with respect to the time intensity of the items. We now calculated the difference in average total RTs of a test of 40 items assembled from the simulated item bank for each class of response behavior. Given the prevalence of both classes, the percentage of test takers permitted to violate the RT constraint, and the skewness of the RT distribution, we were able to choose $\beta = 60$. In the case of a larger number of classes, or a greater number of differences between the classes, a different approach for selecting β will have to be applied. For example, when the mixture model described by Marianti et al. (2014) is applied, the RT behavior of the second class is unspecified. For this class of test takers, the average observed total RTs can be used as an indication. For the mixture RT model of Molenaar and De Boeck (2014), the prevalence of slow and quick response behavior within one test taker could be used to obtain information about how to weight both classes for the whole population. Finally, in the case of dynamic RT models (Fox, 2014), one must take the expectation over the distribution of response behaviors, rather than a weighted combination of classes.

One of the biggest advantages of stochastic programming is that the probabilistic nature of the constraints is really taken into account during test assembly. A disadvantage is that the models don't have the nice properties that 0-1 LP models have when it comes to convexity of the solution space. Fortunately, several approximations have been proposed in the literature, and the approximation by using integrated chance constraints (Klein Haneveld & van der Vlerk, 2006) turned out to work well.

When the strategies are compared with respect to violations of the RT constraint, it can be observed that the strategy that replaced the probabilistic constraint by a series of constraints was the only one that met the RT constraint, irrespective of the class of response behavior to which the test taker belonged. For 15% of the test takers, this strategy would be the right one. For the other 85%, Figure 2 illustrates that the amount of information obtained is far from optimal. This can be seen as the cost of applying a conservative strategy. The expected RT strategy aggregates the RT constraint at the population level. So for 85% of the population, the bound of the RT constraint is slightly tighter, such that this compensates for the expected violation of the RT constraint by the 15% of test takers that need more time. As a result of this, the resulting test might be less informative for a test taker in the class of fast test takers. When robust ATA is applied, items that differ the most in time intensity between both classes of RT behavior are more or less excluded by adding a penalty term for selecting these items. The gain

is that the resulting test is more robust against differences in speed between both classes, but the cost is that the strategy favors items with small differences in time intensity during item selection. For this reason, the resulting test might become less informative as well. Finally, stochastic programming was applied to minimize the loss in information for both the fast and slow test takers by allowing a small violation of the RT constraint. In Figure 2, it can be seen that the gain in information is only small in this example. It will be up to test developers to decide how to value the additional information relative to the probability of violating the RT constraints. One remark must be made, however, with respect to the way the RT constraints were formulated in this study. The four different strategies focused only on the uncertainty due to the mixture of classes of response behavior. Uncertainties in RTs within each class were neglected. All of the constraints were formulated for the time intensities λ_i of the items, rather than for the RTs of the test takers. In order to formulate the constraints with respect to actual RTs, a two-level structure would have to be imposed. But since the purpose of this report was to introduce stochastic programming for dealing with mixture constraints, this additional source of uncertainty was not taken into account.

The next step in our research would be to implement stochastic programming in CAT and in multistage testing. In these modes of testing, information about the response behavior of the test taker becomes available during test administration, and it can be taken into account when selecting the next item or module. The shadow test approach (van der Linden, 2005) is very suitable for dealing with all kinds of constraints, and when it is combined with a stochastic programming method for solving item selection problems, it will be able to deal with probabilistic constraints related to, for example, mixture RT models as well.

Finally, in this report we focused on mixture RT constraints, and stochastic programming turned out to be a method that is suitable for dealing with these constraints. However, application of stochastic programming can easily be generalized to test assembly problems with, for example, mixture IRT models. For these problems, different classes of item information functions must be taken into account, and a probabilistic formulation of the objective function might have to be dealt with. Moreover, the application of stochastic programming could be generalized to any model that distinguishes latent classes of test takers during test assembly. What all of these applications have in common is that groups of test takers behave differently, and the group to which a test taker belongs is unknown in advance. So whenever a test must be assembled for a heterogeneous population, stochastic programming might be considered as an alternative to the more restricted 0-1 LP methods that are currently applied.

References

Ackerman, P. L., & Kanfer, R. (2009). Test length and cognitive fatigue: An empirical examination of effects on performance and test-taker reactions. *Journal of Experimental Psychology*, *15*(2), 163–181.

Bertsimas, D., & Sim, M. (2003). Robust discrete optimization and network flows. *Mathematical Programming, 98,* 49–71.

Birge, J. R., & Louveaux, F. (1997). *Introduction to stochastic programming.* New York: Springer Verlag.

Chen, H., & De Boeck, P. (2014). *Are slow and fast ability test responses different?* Manuscript submitted for publication.

Cho, S. J., & Cohen, A. S. (2010). A multilevel mixture IRT model with an application to DIF. *Journal of Educational and Behavioral Statistics, 35,* 336–370.

Cohen, A. S., & Bolt, D. M. (2005). A mixture model analysis of differential item functioning. *Journal of Educational Measurement, 42,* 133–148.

CRAN (2014). *Statistical software package R.* Retrieved from http://cran.r-project.org/.

Egberink, I. J., Meijer, R. R., & Veldkamp, B. P. (2010). Conscientiousness in the workplace: Applying mixture IRT to investigate scalability and predictive validity. *Journal of Research in Personality*, *44*, 232–244.

Evans, F., & Reilly, R. (1973). A study of speededness as a source of test bias. *Journal of Educational Measurement, 9*, 123–131.

Fan, Z., Wang, C., Chang, H. H., & Douglas, J. (2012). Utilizing response time distributions for item selection in CAT. *Journal of Educational and Behavioral Statistics, 37,* 655–670.

Finkelman, M. D., Kim, W., Weissman, A., & Cook, R. J. (2014). Cognitive diagnostic models and computerized adaptive testing: Two new item-selection methods that incorporate response times. *Journal of Computerized Adaptive Testing, 2,* 59–76.

Fox, J.-P. (2014). *Modeling differential working speed in assessment testing.* (LSAC Report Series, RR 14-05). Newtown, PA: Law School Admission Council.

Fox, J.-P., Klein Entink, R., & van der Linden, W. (2007). Modeling of responses and response times with the package CIRT. *Journal of Statistical Software, 20,* 1–14.Glas, C. A. (2010). *Preliminary manual of the software program Multidimensional Item Response Theory (MIRT).* University of Twente, Enschede, the Netherlands.

Hancock, G. R., & Samuelsen, K. M. (Eds.). (2008). *Advances in latent variable mixture models.* Charlotte, NC: Information Age Publishing.

He, Q., & von Davier, M. (2014). *Extracting sequence patterns from process data of problem solving items using n-grams* . Paper presented at the 30th Workshop on IRT and Educational Measurement, Enschede, the Netherlands, November 19–21, 2014.

Hornke, L. F. (1997). Untersuchung von Itembearbeitungszeiten beim computergestützten adaptiven Testen. *Diagnostica, 43*, 27–39.

Klein Entink, R. H., Fox, J. P., & van Der Linden, W. J. (2009). A multivariate multilevel approach to the modeling of accuracy and speed of test takers. *Psychometrika, 74,* 21–48.

Klein Haneveld, W. K., & van der Vlerk, M. H. (1999). Stochastic integer programming: General models and algorithms. *Annals of Operations Research, 85,* 39–57.

Klein Haneveld, W. K., & van der Vlerk, M. H. (2006). Integrated chance constraints: Reduced forms and an algorithm. *Computational Management Science, 3,* 245–269.

Lawrence, I. (1993). The effect of test speededness on subgroup performance. Research Report 93–49, Princeton, NJ: Educational Testing Service.

Lee, Y. H., & Jia, Y. (2014). Using response time to investigate students' test-taking behaviors in a NAEP computer-based study. *Large-Scale Assessments in Education, 2,* 1–24.

Levine, M. V., & Rubin, D. B. (1979). Measuring the appropriateness of multiple-choice test scores. *Journal of Educational and Behavioral Statistics, 4*, 269–290.

Marianti, S. (2015). *R-package LNIRT.* University of Twente.

Marianti, S., Fox, J.-P., Avetisyan, M., & Veldkamp, B. P. (2014). *Testing for aberrant behavior in response time modeling* (LSAC Report Series, RR 14-02). Newtown, PA: Law School Admission Council.

Maris, E. (1993). Adaptive and multiplicative models for gamma distributed variables, and their application as psychometric models for response times. *Psychometrika, 58,* 445–469.

Masters, G., & Keeves, J. (1999). *Advances in measurement in educational research and assessment.* Amsterdam: Elsevier Science.

Molenaar, D., & De Boeck, P. (2014). *Response mixture IRT modeling of the speed accuracy trade-off in psychometric tests.* Paper presented at the 30th Workshop on IRT and Educational Measurement, Enschede, the Netherlands, November 19–21, 2014.

Muthén, L. K., & Muthén, B. O. (2012). Mplus. *The comprehensive modelling program for applied researchers: User's guide, 5*.

Partchev, I., & De Boeck, P. (2012). Can fast and slow intelligence be differentiated? *Intelligence, 40,* 23–32.

Schnipke, D. L., & Scrams, D. J. (1997). Modeling item response times with a two-state mixture model: A new method of measuring speededness. *Journal of Educational Measurement, 34,* 213–232.

Timmers, C., Walraven, A., & Veldkamp B. P. (2014). *The effect of regulation feedback in a computer-based formative assessment on information problem solving.* Manuscript submitted for publication.

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

van der Linden, W. J. (2006). A lognormal model for response times on test items. *Journal of Educational and Behavioral Statistics, 31,* 181–204.

van der Linden, W. J. (2007). A hierarchical framework for modeling speed and accuracy on test items. *Psychometrika, 72,* 287–308.

van der Linden, W. J. (2009). Conceptual issues in response-time modeling. *Journal of Educational Measurement*, *46*(3), 247–272.

van der Linden, W. J. (2011). Test design and speededness. *Journal of Educational Measurement, 48,* 44–60.

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

van der Linden, W. J., & Guo, F. (2008). Bayesian procedures for identifying aberrant response-time patterns in adaptive testing. *Psychometrika*, *73*, 365–384.

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

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

van der Linden, W. J., & van Krimpen-Stoop, E. M. (2003). Using response times to detect aberrant responses in computerized adaptive testing. *Psychometrika, 68,* 251–265.

Veldkamp, B. P. (2013). Application of robust optimization to automated test assembly. *Annals of Operations Research, 206,* 595–610.

Veldkamp, B. P. (2014). *Some practical issues in computerized adaptive testing with response times* (LSAC Report Series, RR 14-06). Newtown, PA: Law School Admission Council.

Vermunt, J. K., & Magidson, J. (2013). *Latent Gold 5.0 Upgrade Manual.*

Von Davier, M., & Yamamoto, K. (2004). Partially observed mixtures of IRT models: An extension of the generalized partial-credit model. *Applied Psychological Measurement, 28,* 389–406.

Wechsler, D. (2003). *Wechsler intelligence scale for children–Fourth Edition (WISC-IV).* San Antonio, TX: The Psychological Corporation.