Optimal Test Design With Rule-Based Item Generation

Hanneke Geerlings¹, Wim J. van der Linden², and Cees A. W. Glas¹

Abstract

Optimal test-design methods are applied to rule-based item generation. Three different cases of automated test design are presented: (a) test assembly from a pool of pregenerated, calibrated items; (b) test generation on the fly from a pool of calibrated item families; and (c) test generation on the fly directly from calibrated features defining the item families. The last two cases do not assume any item calibration under a regular response theory model; instead, entire item families or critical features of them are assumed to be calibrated using a hierarchical response model developed for rule-based item generation. The test-design models maximize an expected version of the Fisher information in the test and control critical attributes of the test forms through explicit constraints. Results from a study with simulated response data highlight both the effects of within-family item-parameter variability and the severity of the constraint sets in the test-design models on their optimal solutions.

Keywords

Fisher information, hierarchical modeling, item response theory, optimal test design, rule-based item generation

The field of educational and psychological testing has shown a recent trend toward automation, for instance, in the form of new technology for automated item generation (Irvine & Kyllonen, 2002), automated test assembly (van der Linden, 2005), computerized adaptive testing (van der Linden & Glas, 2010), and automated scoring and feedback (Xi, 2010). As for automated item generation, if the cognitive processes involved in solving the test items are known, rules can be defined that create the possibility of real-time automated item generation during test taking (Irvine & Kyllonen, 2002). In addition to a potential increase in cost-effectiveness for large-scale testing programs, the use of explicit item-construction rules in automated item generation also furthers the content validity of tests.

The main goal of this article is to look ahead and explore the integration of rule-based item generation and automated test assembly. This integration is expected to be the next step in the trend toward automation of testing. Successful integration prevents the more
cumbersome process of having a computer generate an entire inventory of new test items in a few seconds followed by days of the manual work required to design and assemble the desired test forms.

However, for this goal to be feasible, several issues need to be addressed. One important issue is how to deal with the fact that the psychometric properties of rule-based generated items are not necessarily a priori known. The solution adopted in this research is calibration of entire families of rule-based generated items that share critical features but differ in minor, less relevant aspects, using hierarchical item response theory (IRT) modeling. More specifically, in addition to a regular response model, a different second-level model is adopted for each family representing its distribution of item parameters, which will be taken to be multivariate normal.

For this setup, item calibration can be replaced by “family calibration,” that is, estimation of the vectors of mean item parameters and covariance matrices for the families using only samples of items from each family (Glas & van der Linden, 2003). Once this has been done, any new item generated from a family does not need to be calibrated; instead, its known distribution can be used to design, assemble, and score test forms. In addition, an extension of the hierarchical model will be used, with a linear structure on the mean difficulties of the items in each family representing the critical features (or radicals) of the families used to generate items (Geerlings, Glas, & van der Linden, 2011). This extension allows for the effects of the radicals on the item families to be estimated and used to precalibrate the families.

The fact that only the family parameters are known requires an adjustment of regular optimality criteria for the design or assembly of test forms. For the case of adaptive testing from a pool with item families, this issue was already addressed by Glas and van der Linden (2003). They used a Bayesian approach with a minimum expected posterior variance criterion for the selection of the items, where the expectation was taken over the family distributions of the items to allow for the remaining uncertainty about their parameter values. In this research, the same idea was followed, but Fisher’s information integrated over the family distributions was used. Thus, instead of the regular item-information function, a “family-information function” for each of the families was used.

Obviously, the success of family calibration and the use of family-information functions depend on the degree of within-family item-parameter variability across all families. The former has already been investigated by Geerlings et al. (2011) and Glas, van der Linden, and Geerlings (2010). One of the goals of the current research was to assess the effect of within-family item-parameter variability on optimal test design and assembly.

In the next sections, first rule-based item generation, a model for the calibration of item families, and the family-information function used are reviewed. Then, optimal test design for three different cases of item generation is discussed: (a) test assembly from a pool of pregenerated, individually calibrated items; (b) test generation on the fly from pools with calibrated item families; and (c) test generation on the fly using calibrated radicals that define the item families. The first case is straightforward; the only difference with current automated test-assembly practices resides in the way the items are obtained. The case serves as a baseline to evaluate the more innovative next two cases, which do not assume any items that have already been generated but specify what item families should be sampled or what item-generation rules should be used to generate the test forms. Using simulated data, the effects of within-family uncertainty about the item parameters and the presence of different numbers of content constraints on the assembled test forms will be evaluated. The article concludes with some practical observations regarding the use of optimal test design for rule-based item generation.
Rule-Based Item Generation

An important distinction is between item features that influence the difficulty of the items and those that have only negligible effects. The former have been called *radicals* (Irvine, 2002); their systematic use can help to ensure the content validity of the items. The latter are known as *incidentals*; they only create surface variation and do not have any systematic effect on item difficulty (Irvine, 2002). Hence, they will be treated as random effects. Generating items with identical radicals but different incidentals is often referred to as “item cloning.” Items cloned from the same combination of radicals are thus expected to have similar psychometric properties but a different “look” (Bejar et al., 2003; Bejar & Yocom, 1991). For example, in the context of statistical word problems, the computations required to solve the problem could be treated as radicals but the topics used in the context stories are assumed to result in incidental variation only (Zeuch, 2011, chap. 7).

Hypotheses regarding the distinction of item features as either radicals or incidentals are typically developed through a cognitive analysis of the item domain. In the literature, such hypotheses have often been tested by coding existing items on the presence or absence of the specified radicals and performing, for example, a regression analysis on IRT-based difficulty parameters (Gorin & Embretson, 2006). Alternatively, the items have been analyzed with Fischer’s linear logistic test model (Fischer, 1973). The size of the estimated effects or the amount of variance explained by them is then taken as measures of the validity of the hypotheses. A drawback of this approach is the potential difficulty to objectively code the presence of the radicals in the items, which may manifest itself by low interrater reliabilities (Gorin & Embretson, 2006).

A more effective approach would be to test the hypotheses directly on data collected on items generated according to the hypothetical radicals and incidentals. This predictive, model-checking approach is possible using, for example, the hierarchical model discussed in the next section. The approach forces the developers of rule-based item-generation software to explicitly define their radicals and incidentals and show how they are to be translated into computer algorithms.

Item generators based on the use of radicals and incidentals have been developed both for the generation of open-ended items (e.g., algebra word problems, Arendasy & Sommer, 2007; statistical word problems, Boer Rookhuiszen, 2011; Theune, Boer Rookhuiszen, op den Akker, & Geerlings, 2011) as well as multiple-choice items (e.g., figural matrices, Arendasy, 2005; directions and distances items, Dennis, Handley, Bradon, Evans, & Newstead, 2002; literacy items, Dennis et al., 2002). In general, whereas multiple-choice items have the advantage of simple scoring, the need to generate distractors for them as well may complicate automated generation. The effects of distractor features on the difficulty of an item are generally difficult to predict, and this topic certainly deserves more research.

If certain combinations of radicals and incidentals result in invalid items, the item generator can be supplied with constraints to exclude such combinations. Also, constraints can be added to avoid the presence of cognitive processes in item solving that are not construct related (Arendasy & Sommer, 2007).

For more information on rule-based item generation, the reader is referred to the edited volumes by Irvine and Kyllonen (2002) and Gierl and Haladyna (2012).

Modeling Rule-Based Item Generation

The joint use of radicals and incidentals in item generation results in families of items, with between-family variation caused by combinations of radicals and within-family variation by the incidentals. The hierarchical structure of items grouped into families can be explicitly taken into
account through multilevel response modeling. Let $n = 1, \ldots, N$ be the persons, $f = 1, \ldots, F$ the item families, and $i_f = 1, \ldots, I_f$ the items from family $f$. Furthermore, let $U_{i_n}$ be the dichotomous variable for the response of person $n$ to item $i_f$. The model for this case proposed by Glas and van der Linden (2003), henceforth denoted as the item cloning model (ICM), extends the three-parameter logistic (3PL) model with a hierarchical structure on the item parameters for each family. A polytomous version of the model was developed by Johnson and Sinharay (2005).

In this article, for numerical reasons, the three-parameter normal-ogive (3PNO) model is used as first-level model:

$$
p(U_{i_n} = 1 | a_{i_f}, b_{i_f}, \gamma_{i_f}) = \gamma_{i_f} + \left(1 - \gamma_{i_f}\right) \Phi(\alpha_{i_f} \left[ \theta_n - b_{i_f} \right]),
$$

where $a_{i_f}$, $b_{i_f}$, and $\gamma_{i_f}$ are the discrimination, difficulty, and guessing parameters for item $i$ in family $f$, respectively, $\theta_n$ is the person parameter, and $\Phi(.)$ is the cumulative normal distribution function. The two-parameter normal-ogive (2PNO) model arises as the first-level model when $\gamma_{i_f} = 0$ for $i_f = 1, \ldots, I_f$. Generally, the choice of the normal ogive instead of the logistic link function, in combination with a reparameterization of the model has the advantage of easy implementation of the Gibbs sampler for the ICM extended with a linear structure on the family difficulty parameters used in the empirical examples below.

The item parameters are transformed as

$$
\xi_{i_f} = \left(a_{i_f}, b_{i_f}, \logit \gamma_{i_f}\right),
$$

with $\mu_f$, and $\Sigma_f$ family parameters representing the vector of the means of the item parameters for family $f$ and their variability about these means, respectively. $c_{i_f}$ is used to denote the transformed guessing parameter. If the covariance matrices $\Sigma_f$ are approximately equal, the assumption of a common covariance matrix $\Sigma$ instead of family-specific matrices is convenient. This case may arise, for example, when all families are generated by the same set of incidentals, and there exists no interaction between the effects of radicals and incidentals.

A linear structure on the mean difficulty parameters $\mu_{b_f}$, $f = 1, \ldots, F$, can be used to model the fact that family membership is determined by the radicals (Geerlings et al., 2011). Let $r = 1, \ldots, R$ be the radicals used to generate the families. The proposed linear structure is

$$
\mu_{b_f} = \sum_{r=1}^{R} d_{fr} \beta_r,
$$

where $\beta_r$ denotes the effect of radical $r$ on the mean family difficulty $\mu_{b_f}$ and $d_{fr}$ is a design variable denoting how often radical $r$ should be used to construct an item from family $f$. Thus, Equation 4 represents the mean family difficulty as a weighted sum of the effects of the radicals used to generate items from it.

Observe that the use of Equation 4 involves the specification of an $F \times R$ design matrix $D = (d_{fr})$. In the literature, such a matrix is also referred to as a Q-matrix (Fischer, 1973; Tatsuoka, 1983).
The basic model restricted by Equation 4 will be labeled the linear item cloning model (LICM). Furthermore, the model with family-specific covariance matrices will be labeled the LICM-F, whereas the model with a common covariance matrix is referred to as the LICM-C.

A Gibbs sampling algorithm to estimate the parameters of a reparameterization of the models in a Bayesian manner was presented in Geerlings et al. (2011, Appendix). Unlike regular item calibration, where the item parameters are treated as fixed effects, estimation of the family parameters involves random samples of items from the families. This means that any constraints on specific combinations of radicals and incidentals in the item generator should also be imposed when generating a calibration sample of items. Geerlings et al. presented a parameter recovery study that shows the precision with which the family parameters are estimated for specific numbers of families, items per family, and persons taking the items.

As already indicated, this model framework can be used to test hypotheses regarding the effects of the radicals and incidentals used in the item-generation rules. In this regard, it is important to check the general validity of the first-level response models (unidimensionality, monotonicity, and local independence) as well as the second-level models (LICM: linearity; no residual variance in the prediction of the family difficulties by the radicals; multivariate normality of the family distributions; LICM-C: homogeneity of within-family item-parameter variability across families). Besides, the hypotheses on the impact of the radicals on the item families can be tested checking the validity of versions of the model with alternative design matrices. For statistics to test such hypotheses and empirical examples, see Geerlings (2012, chap. 4).

**Family-Information Function**

Generally, item-information functions are defined as

\[ I_{i_f}(\theta) = -E_u \left[ \frac{\partial^2}{\partial \theta^2} \ln p \left( u|\theta, \xi_{i_f} \right) \right] \]

(see, for example, van der Linden & Pashley, 2010, Section 1.2.1). This type of information function can be used to assemble test forms from a pool with items pregenerated by the computer and individually calibrated upon their generation.

In the two other cases of test design, the parameters of the individual items are unknown but the item families have been calibrated; that is, their second-level parameters \( \mu_f \) and \( \Sigma_f \) have been estimated or have been computed from estimates of \( \beta \) and \( \Sigma \). For these cases, family-information functions can be used, which are defined as the expected information about \( \theta \) in the response to a random item from family \( f \). The function is obtained integrating Equation 5 over the item parameters, as

\[ I_f(\theta) = -E_{i_f} \left[ \frac{\partial^2}{\partial \theta^2} \ln \int p \left( u|\theta, \xi_{i_f} \right) p \left( \xi_{i_f}|\mu_f, \Sigma_f \right) d\xi_{i_f} \right]. \]

A convenient approximation of the integral can be obtained by Monte Carlo integration. Details of the approximation are given in Appendix A. Besides, Appendix B shows the results from a small simulation study investigating the number of iterations needed to obtain a required precision.
Three Cases of Automated Test Design

The approach to automated test design is based on mixed integer programming (MIP; for a general introduction, see van der Linden, 2005). Three different cases are distinguished, differing in flexibility of test generation.

The first case is assembly of test forms from pregenerated, calibrated item pools. In this case, the computer has already been used to generate all the items in the pool and each of them has been calibrated prior to operational testing. Whereas a stand-alone response model as the 3PNO model could be used to estimate the item parameters, the authors suggest the use of the (L)ICM, because, in a Bayesian fashion, it allows the item parameters’ estimates to borrow strength from their family means, leading to more precise estimates. The test forms are assembled from the pool using the regular item-information functions calculated from the estimated item parameters. The functions are used to define the objective function in the MIP model for the assembly of the tests. The content constraints in the model are derived from the radicals and incidentals that define the item families.

The second case is test generation on the fly from a pool of calibrated item families. In this case, the family parameters \( m \) and \( S \) of the (L)ICM are assumed to be estimated. The test forms are not assembled from a pool of existing items but designed by the MIP model. The model has an objective function based on the family-information functions. Its content constraints are derived from the radicals describing the families. Its solution indicates how many items to generate from which families. The items are then generated by random application of incidentals. Given the potentially large numbers of different items that can be generated from the families, this fully automated type of test generation on the fly offers the advantage of giving each test taker a unique test. In doing so, test security problems due to item overexposure are avoided. Note, however, that family overexposure may still be an issue. The likelihood of family disclosure depends on the power of the incidentals, that is, the variability of the surface features they are able to create.

The third case goes one step further and involves test generation on the fly directly from calibrated radicals defining the item families. This case is possible for the model with a linear structure on the mean difficulty parameters and a common covariance matrix, in combination with the assumptions of common family means of the discrimination and guessing parameters. Once these common parameters and the effect parameters for the radicals, \( \beta_r, r = 1, \ldots, R \), have been estimated, they enable a prediction of the distribution of the item parameters for each family. Otherwise, the test design and assembly process is entirely identical to the previous case: The MIP model has an objective function based on the family-information functions along with content constraints on the radicals. The actual test form is generated by applying the combinations of radicals corresponding to the selected families and randomly applying the incidentals. As an alternative to the assumption of common family means for the guessing and discrimination parameters, a model as in Equation 4 could be used to predict them as a linear combination of radical effects.

The next sections discuss each of the three cases in more detail. For each case, the type of MIP model that can be used is discussed.

Test Assembly From a Pregenerated Item Pool

Assembling a test from a Pregenerated Item Pool with families of items with calibrated parameters \( \xi_f = (\xi_{ij}) \) can be seen as a series of decisions of whether or not family \( f \) should be represented in the test by including item \( ij \). The problem can be formalized using 0-1 decision variables for all families and items in the pool, \( x = (x_1, \ldots, x_f, \ldots, x_F, x_1, \ldots, x_f, \ldots, x_IF) \), defined as...
The decision variables are used to formulate the objective function and constraints in the optimization model presented below. The objective function maximizes the total information in the test subject to a set of weights $R_p$ at $\theta_p$, $p = 1, \ldots, P$, that represent the relative shape of the target for the information function. The constraints represent all other test specifications. Constraints can be imposed both on the radicals and incidentals. For the former, constraints with the design variables $d_f$ in Equation 4 can be used. For example, for the statistical word problems discussed earlier, these variables could serve as indicators of whether or not probability formula $r$ needs to be applied to solve an item from family $f$. For the latter, constraints with a second type of design variables $t_{i_c}$ can be used, with $c = 1, \ldots, C$ denoting the incidentals available for the families. For example, for the statistical word problems discussed earlier, the design variables $t_{i_c}$ could serve as indicator variables for the context story used to generate the problem for item $i_f$.

The following model can be used to assemble a test form with $k$ items from each of $l$ families with an optimal information function subject to sets of constraints on their radicals and incidentals:

Maximize $y$ (maximum information) \hspace{1cm} (7)

subject to

$$\sum_{f=1}^{F} \sum_{i_f=1}^{I_f} I_{i_f}(\theta_p)x_{i_f} \geq R_p y, \quad p = 1, \ldots, P,$$ (relative shape of target) \hspace{1cm} (8)

$$\sum_{f=1}^{F} x_f = l,$$ (number of families) \hspace{1cm} (9)

$$-kx_f + \sum_{i_f=1}^{I_f} x_{i_f} = 0, \quad f = 1, \ldots, F,$$ (number of items per family) \hspace{1cm} (10)

$$\sum_{f=1}^{F} d_f x_f \leq b_r, \quad r = 1, \ldots, R,$$ (radicals) \hspace{1cm} (11)

$$\sum_{f=1}^{F} \sum_{i_f=1}^{I_f} t_{i,c} x_{i_f} \leq b_c, \quad c = 1, \ldots, C,$$ (incidentals) \hspace{1cm} (12)

$$x_f = (0, 1), \quad f = 1, \ldots, F,$$ (variables for families) \hspace{1cm} (13)

$$x_{i_f} = (0, 1), \quad i_f = 1, \ldots, I_f, \ldots, I_F,$$ (variables for items) \hspace{1cm} (14)

$$y = (0, \infty),$$ (auxiliary variable) \hspace{1cm} (15)

with $I_{i_f}(\theta_p)$ the information provided by item $i_f$ at ability value $\theta_p$ (Equation 5). The objective is of the maximin type; the real-valued variable $y$, which is maximized in Equation 7, is a
common factor in the lower bounds to the sum of the item-information functions in Equation 8 (for details, see van der Linden, 2005, Section 5.1.4). Observe that \( R_p = 1.0 \) for \( p = 1, \ldots, P \)
represents the case of a uniform target for the test-information function. The equalities in Equation 9 and Equation 10 constrain the numbers of families and items per family to be selected in the test. The latter also ensures that items are selected from a family if and only if the family is selected. (If \( x_f = 0 \), the equation only holds if \( x_{if} = 0 \) for all \( i_f = 1, \ldots, I_f \), but if \( x_f = 1 \), it holds only if the sum of all \( x_{if} \) equals \( k \).) The constraints in Equation 11 and Equation 12 impose bounds \( b_r \) and \( b_c \) on the total number of times radicals and incidentals are represented in the test, respectively.

The equations only represent the core of an optimization model for test assembly from a pre-generated pool of items. In a real-world application, several other types of constraints may have to be added; for examples, see van der Linden (2005). A solution to the optimization problem in Equations 7 to 15 is a string of values for the 0-1 decision variables that maximizes its objective function. It can be found using a solver of the branch-and-bound type (Williams, 1993, Section 6.2). In the later examples, the MIP solver in CPLEX 9.0 (ILOG, 2003) was used.

**Test Generation on the Fly From Calibrated Families**

For the second case, the problem changes from the assembly of a test from a Pre-generated Item Pool to one of designing a test, that is, identifying the families from which to generate the items for the test. The solution to the MIP model identifies these families. The actual items in the form are generated by randomly sampling the incidentals.

The MIP model for this type of generation on the fly contains decision variables for the families, \( x_f \), only. The selection of families is based on the family-information measure \( I_f(\theta) \) in Equation 6. The proposed model is

\[
\text{Maximize } y \quad \text{(maximum information)} \tag{16}
\]

subject to

\[
\sum_{f=1}^{F} I_f(\theta_p) x_f \geq R_p y, \quad p = 1, \ldots, P, \quad \text{(relative shape of target)} \tag{17}
\]

\[
\sum_{f=1}^{F} x_f = l, \quad \text{(number of families)} \tag{18}
\]

\[
\sum_{f=1}^{F} d_{rf} x_f \leq b_r, \quad r = 1, \ldots, R, \quad \text{(radicals)} \tag{19}
\]

\[
x_f = (0, 1), \quad f = 1, \ldots, F, \quad \text{(variables for families)} \tag{20}
\]

\[
y = (0, \infty), \quad \text{(auxiliary variable)} \tag{21}
\]

Again, the model is for the selection of \( l \) families (see Equation 18), which together provide an optimal information function with the relative shape of the target in Equation 17 and satisfy the content constraints on the radicals in Equation 19. There are no constraints on incidentals; consistently with the use of the family-information functions in Equation 17, they are randomly sampled.
Test Generation on the Fly Using Calibrated Radicals Only

This time, the decision variables $x_v$ are redefined as admissible combinations of radicals $v$:

$$x_v = \begin{cases} 
1, & \text{combination } v \text{ of radicals is selected;} \\
0, & \text{otherwise.}
\end{cases}$$

The test-design model in Equations 16 to 21 can now be used to generate items from families for the selected (admissible) combinations of radicals. In principle, this is possible even for combinations that have never been used before. The only difference resides in an extension of the hierarchical response model, which now involves prediction of the mean difficulties for each family through the estimated effects of the radicals in Equation 4.

Evaluation of Results

The evaluation exists of two different parts. First, the effect of the within-family variability of the item parameters on the family-information functions used in the last two test-design models is analyzed. This is done by studying a few examples with different values for the critical second-level parameter that reflects the variability—the family covariance matrix. Second, results are presented from different applications of the test-design models based on simulated response data. Because all true parameter values are known, the effect of using different target information functions and constraint sets on the results could be evaluated.

Effect of Within-Family Item-Parameter Variability on Family Information

The variability of the item parameters within a family has a potentially large impact on the value of the family-information function in Equation 6. The effects of within-family variability are explored for the 3PNO model as first-level model.

Setup of the study. Mean family discrimination was fixed at 1, mean family guessing at 0.2 (which corresponds to $\mu_c = \logit(0.2)$), and mean family difficulty was chosen to be optimal with respect to $\theta = 0$; that is, $\mu_b = -0.258$ (see Wolfe, 1981). The variances of the item parameters in the families were set equal to either 0.00 or 0.05 ($\sigma_a^2$), 0.00 or 0.50 ($\sigma_b^2$), and 0.00 or 0.20 ($\sigma_c^2$). Note that the nonzero values are large, but their relative magnitudes are not unusual. The covariances were chosen to correspond to a correlation $\rho$ of 0, .5, or –.5. The values of the family-information functions for each of the different covariance matrices were computed at a grid of 31 ability values $u = 0.0(0.2)3.0$. Their integrals were approximated using 10,000 Monte Carlo draws from the item-parameter distributions for each family (see Appendix B). The curves in Figures 1 and 2 were obtained interpolating between these values.

Results. Figure 1 presents the curves for the case of zero correlations in combination with zero variance for all item parameters (solid line) or positive variance for only one type of parameter (dashed and dotted lines). An increase in the variance in the difficulty parameters $\sigma_b^2$ led to a decrease in the family information over a large range of the ability scale. However, an increase in the variance of the discrimination parameters $\sigma_a^2$ led to a decrease only at the ability levels away from the optimal difficulty value. In general, the effects of an increase in the variance of the guessing parameters were small.

The first plot in Figure 2 is for the case of $\sigma_a^2 = 0.05$, $\sigma_b^2 = 0.50$, and $\sigma_c^2 = 0.00$ with different correlations between the item discrimination and difficulty parameters. A positive correlation resulted in an increase in information at $\theta > \mu_b$ but a decrease at $\theta < \mu_b$ (compare the curves for 0 and 0.5 correlation in the upper plot in Figure 2). This can be explained as follows: In case
of a positive correlation, larger discrimination parameters coincide with larger difficulty parameters. As the former is the main determinant of the family-information function, the positive correlation introduces a shift of the function to the right. For a negative correlation, the reverse shift can be observed.

The second plot in Figure 2 is for the case of $\sigma^2_a = 0.00$, $\sigma^2_b = 0.50$, and $\sigma^2_c = 0.20$ and different correlations between the item difficulty and guessing parameters. A positive correlation resulted in a small shift to the left for the family-information function; negative correlation in a shift to the right. These shifts can be explained by the fact that larger guessing parameters result in less information. However, the effects were quite small and are hard to discern in the figure.

The third plot in Figure 2 is for the case of $\sigma^2_a = 0.05$, $\sigma^2_b = 0.00$, and $\sigma^2_c = 0.20$ and different correlations between the item discrimination and guessing parameters. A positive correlation had a partially counterbalancing effect on the family-information function (more information for larger discrimination parameters but less information because of larger guessing parameters). However, negative correlation means larger discrimination for lower guessing parameters and thus generally higher values for the family-information function than without any covariance between them.

Note that when all variances are 0 (solid line in Figure 1), all item parameters are equal to their respective family means, and family information equals item information (Equation 5) with the family values substituted for the item parameters. If the (co)variances are unequal to zero, use of item information in this way leads to overestimation of the actual amount of information on $\theta$ in the response to a random item from the family.

**Examples Using Simulation Data**

The goal was to illustrate the use of the models for the cases of test assembly from a Pregenerated Item Pool and test generation on the fly from pools with calibrated item families. At the same time, the examples provide empirical estimates of the decrease in test information due to the use of family-information instead of item-information functions in the latter.
Item pools. A total of 32 item families was created through the use of five dichotomous radicals with a fully crossed design. All family-mean parameters used in this study are shown in Figure 2.

**Figure 2.** Family information for the 3PNO as the first-level model as a function of the correlation between the item parameters and ability level

Note: 3PNO = three-parameter normal ogive. The family-mean parameters were fixed at $\mu_a = 1$, $\mu_b = -0.258$, and $\mu_c = 0.2$. For the upper plot, $\sigma_a^2 = 0.05$, $\sigma_b^2 = 0.50$, and $\sigma_c^2 = 0.00$; for the middle plot, $\sigma_a^2 = 0.00$, $\sigma_b^2 = 0.50$, and $\sigma_c^2 = 0.20$; and for the lower plot, $\sigma_a^2 = 0.05$, $\sigma_b^2 = 0.00$, and $\sigma_c^2 = 0.20$.
Table 1. The mean difficulties of the families in this table were computed from Equation 4, with $\beta = (-2.0, 1.0, 0.3, 0.9, 0.6, 1.2)$, where the first component of this vector represents the intercept and the next components are the regression coefficients for the five radicals. The mean discrimination parameters for the families were sampled from 0.8(0.01)1.7. Likewise, the mean guessing parameters were sampled from 0.1(0.01)0.2 and subsequently transformed into the logits in Equation 2. Observe that the covariance matrix of the family-mean parameters in Table 1 can be used as a measure of the between-family variation in this study. The within-family covariance matrices $\mathbf{S}$ were chosen to be equal across families (LICM-C). Four different within-family covariance matrices were used, equal to 0.01, 0.05, 0.1, and 0.2 times the size of the between-family matrix for Table 1. In each of the 100 replications, 10 or 20 items were sampled from the family distributions.

**Setup of the study.** Two different kinds of test-assembly models were formulated: a baseline model without any of the constraints on radicals in Equation 11 (M1) and an alternative model with constraints on the distribution of the radicals in the test (M2). Both for M1 and M2, the general formulation in Equations 7 to 15 was used. In either case, the objective was maximization of the height of the test-information function while maintaining the relative shape defined by the weights $R_p, p = 1, 2, 3$, at $\theta_p = -1, 0, 1$. The weights were set equal to (1, 1, 1) (uniform relative target) or (1, 2, 1) (relative target with twice as much information at $\theta = 0$). The number of families selected was equal to $l = 10$ or 20, while the number of items selected from each family was fixed at 1. Thus, the length of the test was also equal to $l$. The second model (M2) had 10 constraints on the radicals added to it. Each of the five radicals was restricted to occur between 5 and 6 or 10 and 12 times in the test with test length $l$ equal to 10 or 20, respectively.

All MIP models were solved using CPLEX 9.0 (ILOG, 2003).

To compare the results for test assembly from a Pregenerated Item Pool and test generation on the fly from Calibrated Families, all models were used twice, once with a target based on the item-information functions in Equation 8 and a second time with a target based on the family information in Equation 17. Both types of information functions were calculated at the three ability values, $\theta = -1, 0, 1$. The family-information functions were calculated using Monte Carlo integration with a sample size of 10,000.

As the true values of the item parameters for the simulated items were known, the objective function values for the solutions produced by the models for the case of test generation on the
Figure 3. Mean value of the objective function (\(\bar{y}\)) for Models M1 and M2 with a uniform relative target and with item selection and family selection, as a function of the ratio of within to between item-parameter variability (W–B ratio)

Note: PIP = test assembly from a pregenerated item pool; CF = test generation on the fly from calibrated families; Fam Inf = family information; True Item Inf = true item-information. The upper plot is for the condition with \(I = 10\) selected families and \(I_f = 10\) items per family in the item pool, the middle plot for \(10\) families and \(20\) items per family, and the lower plot for \(20\) families and \(10\) items per family.
fly from Calibrated Families could be recalculated replacing the family-information functions by the item-information functions with these known parameters. The difference between the two results allows an assessment of the effects of test assembly based on knowledge of item families only.

**Results.** Figures 3 (uniform target at \( \theta = \{-1, 0, 1\} \)) and 4 (target with twice as much information at \( \theta = 0 \)) show the results for Models M1 and M2 in the form of the mean values of their objective functions \( \langle y \rangle \) for the cases of test assembly from a Pregenerated Item Pool and test generation on the fly from Calibrated Families, as a function of (a) the number of items per family \( I_f \), (b) the number of selected items/families \( l \), and (c) the within–between (W–B) ratio of the item-parameter variability. For the case of test generation on the fly from Calibrated Families, both the values of the objective functions with the family-information functions used in the assembly (Calibrated Families; Fam Inf) and with their family functions replaced by the true item-information functions (Calibrated Families; True Item Inf) are given.

A comparison between Pregenerated Item Pool and Calibrated Families shows that, with increasing variance of the item parameters per family, the information in a test produced by the former increased. This can be explained as follows: Increasing within-family variance means increasing the variance in the information in its items as well, from which the most informative items are selected. However, family information decreases because of the larger uncertainty about the item-parameter values—an effect already discussed in the previous section. Besides, for the case of test generation on the fly from Calibrated Families, the comparison between Fam Inf and True Item Inf shows minimal differences for small within-family variances, whereas for larger within-family variances True Item Inf was sometimes larger than Fam Inf. In other words, the results for test generation on the fly from Calibrated Families tended to be somewhat conservative for the larger within-family variances.

As expected, the addition of the extra constraints on the distribution of the radicals in M2 caused a decrease in the values of the objective functions, both for the assembly from a Pregenerated Item Pool and the generation on the fly from Calibrated Families. The effect was more pronounced for the latter, though.

Doubling the number of items per family, \( I_f \), led to a slight increase in the values of the objective function for the assembly from a pregenerated pool, but the results remained approximately the same for the case of test generation from a pool of Calibrated Families. This effect of the length of the test relative to the size of the item pool is in agreement with the results obtained by Hambleton, Jones, and Rogers (1993) in their study of the effect of parameter estimation error on item selection. When twice the number of families was selected, the absolute difference between the results for the two different cases of test assembly increased (notice the different scales for the respective plots); however, the shape of the plots remained approximately the same.

All results for the two targets for the test-information functions were comparable. In general, for the uniform target, larger values of the objective function were obtained for both cases of test assembly.

Figures 5 (uniform target at \( \theta = \{-1, 0, 1\} \)) and 6 (target with twice as much information at \( \theta = 0 \)) show the effects of increasing the within-family variance on the two cases of test assembly (this time with the 2PNO as first-level model; that is, \( \mu_y = 0 \)). For both figures, the number of families selected was \( l = 10 \), the number of items per family was equal to \( I_f = 10 \), and the model was M1. The plots in the left-hand columns show the item and family parameters in the pool, with the selected items and families highlighted in black. The plots in the right-hand columns show the corresponding information curves for test assembly from a Pregenerated Item Pool and generation on the fly from Calibrated Families. The effects of the increase in within-family variability can be seen by comparing plots in different rows. The item parameters for the top rows had a within–between family variability (W–B) ratio of 0.01; for the lower rows, the
Figure 4. Mean value of the objective function ($y$) for Models M1 and M2 with a relative target with twice as much information at $\theta = 0$ than at $\theta = -1$ and $\theta = 1$ and with item selection and family selection, as a function of the ratio of within to between item-parameter variability (W–B ratio)

Note: PIP = test assembly from a pregenerated item pool; CF = test generation on the fly from calibrated families; Fam Inf = family information; True Item Inf = true item-information. The upper plot is for the condition with $l = 10$ selected families and $I_f = 10$ items per family in the item pool, the middle plot for 10 families and 20 items per family, and the lower plot for 20 families and 10 items per family.
ratios were 0.05, 0.10, and 0.20, respectively. The information functions for test assembly from Calibrated Families decreased with increasing variability in the item parameters, whereas the results for generation from a Pregenerated Item Pool tended to approximate their targets better.

**Figure 5.** Simulated family and item parameters (left column) and family and item-information curves (right column) for within–between covariance ratios of 0.01 (upper row), 0.05 (second row), 0.10 (third row), and 0.20 (last row), for a test selected based on a uniform relative target.

Note: PIP = test assembly from a pregenerated item pool; CF = test generation on the fly from calibrated families.
Discussion

The main goal of this research was to explore the possibility of automated test generation, that is, the integration of automated item generation and test design/assembly. Three different cases of test design were discussed, with each next case offering an increase in flexibility of automated test generation but at the cost of an increasing number of model assumptions. For instance, the two cases of on-the-fly test generation have the advantages of not having to calibrate newly generated items and generating unique tests in real time for different test takers but do require the assumption of known distributions of the item parameters within families. The assumption of multivariate normality for these distributions can be facilitated by using an appropriate transformation of the item parameters, such as the logit transformation for the guessing parameter in Equation 2.

The most ambitious case ("Test Generation on the Fly Using Calibrated Radicals Only") allows for a further reduction of item calibration efforts because of its additional assumption of a linear relationship between the radical effects and the family means.

In practical applications, the assumptions can be tested using, for example, the model fit assessment methods discussed in Geerlings (2012). The results can help to decide which case(s) of test design should be appropriate for the intended domain of test items.

Another goal of this research was to assess the degree to which within-family item-parameter variability affects the solutions to the on-the-fly test-design models. In practice, the advantages of not having to calibrate every single item have to be weighed against the loss of information in the scores of the test takers. However, other factors should play a role in the decision. For example, items from families with small item-parameter variability may be remembered more easily by test takers and are thus vulnerable to test security breaches in the form of "family disclosure." However, to mitigate the possible effects of low item-parameter variability on family disclosure, an exposure-control method could be used that, with increasing exposure of a family over time, gives lower weight to it in the test-assembly process.

The current research has a link to previous research on the effects of item-parameter uncertainty on optimal test assembly, which focused almost exclusively on uncertainty due to the use of small calibration samples (Hambleton et al., 1993; van der Linden & Glas, 2000). These effects can be described as capitalization-on-chance effects: Small calibration samples lead to larger positive errors of the discrimination parameters of some of the items, and hence to their likely selection in the test, with a less than nominal test information and underestimation of the standard error of the ability estimates as results (Tsutakawa & Johnson, 1990; Zhang, Xie, Song, & Lu, 2011). The magnitude of the capitalization-on-chance effect not only depends on the size of the calibration sample but also on the variability of the true item parameters (van der Linden & Glas, 2001) and the item bank size–test length ratio (Hambleton et al., 1993; Hambleton & Jones, 1994). However, in real-world applications, the effect of these factors is mitigated by the content constraints added to the test-assembly problem (Hambleton et al., 1993). In the current context of optimal assembly from families of items, the opposite of a capitalization-on-chance effect can be expected: Instead of selecting the most informative items from the pool, the items are randomly selected from the families that are most informative in the sense of high mean information. As the actual items in these families are distributed around these means, the true test-information functions obtained for family selection will be lower than those for optimal selection of individually calibrated items from the same pool.

Another effect of item-parameter uncertainty specific to adaptive testing is that of systematic errors in the ability parameters when the distribution of the item difficulty parameters within the pool is nonuniform (Doebler, 2012). The systematic errors occur because the nonuniformity implies an unequal probability of selecting an item with a positive or negative random error of
the difficulty parameter. With regard to within-family item-parameter variability, the effect does not occur when the two-stage item-generation procedure discussed in this article is used; that is, when the item families are optimally selected and the incidentals are randomly applied to
generate the items. In this case, the effect only occurs due to any imprecision with which the family parameters are estimated. As these parameters are usually estimated with high precision, the effect is expected to be small in most cases.

Finally, this article proposed a few alternative combinations of automated item generation and test design for domains where distinctions between item features that can be treated as radicals and incidentals are appropriate. Although automated item generators based on this distinction are already available, the authors are not aware of any earlier efforts to combine them with optimal test-design methods. To decide on the feasibility of such combinations, evaluation studies for different areas of application are needed.

Appendix A

Family Information

Family information is defined as

$$ I_f(\theta) = -E_u \left[ \frac{\partial^2}{\partial \theta^2} \ln \int p(u|\theta, \xi_j) p(\xi_j|\mu, \Sigma) d\xi_j \right]. \quad (22) $$

Let $\int \ldots dG = \int \ldots p(\xi_j|\mu, \Sigma) d\xi_j$, so that Equation 22 can be abbreviated as

$$ I_f(\theta) = -E_u \left[ \frac{\partial^2}{\partial \theta^2} \ln \int p(u|\theta, \xi_j) dG \right]. $$

Let $L_u = \int p(u|\theta, \xi_j) dG$ denote the likelihood of the hierarchical model (the item cloning model [ICM] or linear item cloning model [LICM]), and let $L_1 = \int p(u = 1|\theta, \xi_j) dG$ and $L_0 = \int p(u = 0|\theta, \xi_j) dG$ denote the probabilities of a correct and incorrect response, respectively, according to this model. Family information is then derived as

$$ I_f(\theta) = -E_u \left[ \frac{\partial^2}{\partial \theta^2} \ln L_u \right], $$

$$ = -E_u \left[ \frac{\partial}{\partial \theta} L_u \right], $$

$$ = -E_u \left[ L_u \frac{\partial^2 L_u}{\partial \theta^2} - \left( \frac{L_u'}{L_u} \right)^2 \right] = -E_u \left[ L_u'' \frac{L_u'}{L_u} - \left( \frac{L_u'}{L_u} \right)^2 \right]. \quad (23) $$

Let $P_u = p(u|\theta, \xi_j)$ be the likelihood of the first-level model, and let $P_1$ and $P_0$ be the probabilities of a correct and incorrect response according to this model, respectively. Making use of Leibniz’s rule for changing the order of integration and differentiation (e.g., Casella & Berger, 2002, Section 2.4), the first and second derivatives of $L_1$ and $L_0$ can be obtained by Monte Carlo integration over the first and second derivatives of $P_1$ and $P_0$, respectively:

$$ L_1' = \frac{\partial}{\partial \theta} \int P_1 dG = \int \frac{\partial P_1}{\partial \theta} dG, \quad L_1'' = \int \frac{\partial^2 P_1}{\partial \theta^2} dG, \quad L_0' = \int \frac{\partial P_0}{\partial \theta} dG, \quad L_0'' = \int \frac{\partial^2 P_0}{\partial \theta^2} dG. \quad (24) $$

The expectation in Equation 23 can thus be rewritten as

$$ I_f(\theta) = -E_u \left[ \frac{\partial^2}{\partial \theta^2} \left( \int P_u dG \right) \right], $$

$$ = -\left[ \int \frac{\partial^2}{\partial \theta^2} \left[ P_1 dG - \left( \int \frac{\partial}{\partial \theta} [P_1] dG \right)^2 \right] \right] - \left[ \int \frac{\partial^2}{\partial \theta^2} \left[ P_0 dG - \left( \int \frac{\partial}{\partial \theta} [P_0] dG \right)^2 \right] \right]. \quad (25) $$
The first and second derivatives of $P_1$ and $P_0$ are well known and easy to compute (see Table A1).

### Appendix B

**Number of Monte Carlo Iterations**

A small simulation study was performed to investigate the required number of iterations to obtain desirable precision in the computed family information. Both the 2PNO and 3PNO models were used as first-level models. For the former, the mean discrimination and difficulty parameters per family were fixed at 1 and 0, respectively; for the latter, the mean discrimination and guessing parameters were fixed at 1 and 0.2, while the mean difficulty was chosen to be optimal at $\theta = 0$; that is, $\mu_b = -0.258$ (see Wolfe, 1981). For both models, the variances and covariance of all parameters were set equal to 0.1 and 0.05, respectively. Family information was computed 100 times at $\theta = 1$, 0, and 1. The number of Monte Carlo iterations to compute the information was 1,000, 5,000, 10,000, or 20,000. The standard deviation over the replications was used as a measure of precision of the information estimates.

Table B1 presents the results from the study. The larger values for the family-information measure required more iterations to obtain the same precision; compare, for example, the differences between the standard deviations for the two-parameter and three-parameter models and those for $\theta = 0$ and $\theta = \{-1, 1\}$. In general, the precision was high. In the main simulation studies, 10,000 iterations were used to compute family information. As shown in Table A2, for

### Table A1. Probability of a Correct Response ($P_1$) and Its First ($P'_1$) and Second ($P''_1$) Derivative With Respect to $\theta$ Two Normal Ogive First-Level Models

<table>
<thead>
<tr>
<th></th>
<th>2PNO</th>
<th>3PNO</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_1$</td>
<td>$\Phi(a</td>
<td>\theta - b</td>
</tr>
<tr>
<td>$P'_1$</td>
<td>$a\phi(a</td>
<td>\theta - b</td>
</tr>
<tr>
<td>$P''_1$</td>
<td>$-a^2(\theta - b)\phi(a</td>
<td>\theta - b</td>
</tr>
</tbody>
</table>

Note: 2PNO = two-parameter normal ogive; 3PNO = three-parameter normal ogive. The first and second derivative of the probability of an incorrect response equal the respective derivatives for a correct response, with a reversed sign.

### Table B1. Standard Deviation of the Computed Family Information for the 2PNO and the 3PNO as the First-Level Model as a Function of the Number of Monte Carlo Iterations

<table>
<thead>
<tr>
<th>Number of iterations</th>
<th>$\theta_1 = -1$</th>
<th>$\theta_2 = 0$</th>
<th>$\theta_3 = 1$</th>
<th>$\theta_1 = -1$</th>
<th>$\theta_2 = 0$</th>
<th>$\theta_3 = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,000</td>
<td>0.006</td>
<td>0.011</td>
<td>0.009</td>
<td>0.003</td>
<td>0.007</td>
<td>0.005</td>
</tr>
<tr>
<td>5,000</td>
<td>0.003</td>
<td>0.005</td>
<td>0.003</td>
<td>0.002</td>
<td>0.003</td>
<td>0.002</td>
</tr>
<tr>
<td>10,000</td>
<td>0.002</td>
<td>0.003</td>
<td>0.003</td>
<td>0.001</td>
<td>0.002</td>
<td>0.001</td>
</tr>
<tr>
<td>20,000</td>
<td>0.001</td>
<td>0.002</td>
<td>0.002</td>
<td>0.001</td>
<td>0.002</td>
<td>0.001</td>
</tr>
</tbody>
</table>

Note: 2PNO = two-parameter normal ogive; 3PNO = three-parameter normal ogive.
this number the standard error of the family-information estimates for each of the models can
be expected to be smaller than 0.005.

Editor’s Note
This manuscript was reviewed and accepted under the editorship of Mark Davison.

Declaration of Conflicting Interests
The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publi-
cation of this article.

Funding
The authors disclosed receipt of the following financial support for the research, authorship, and/or publi-
cation of this article: This work was partially supported by the Deutsche Forschungsgemeinschaft (Grant
Number HO 1286/5-1).

References
of a schema-based isomorphic approach to the automatic generation of quantitative reasoning items.
*Learning and Individual Differences, 17*, 366-383.
feasibility study of on-the-fly item generation in adaptive testing. *Journal of Technology, Learning,
and Assessment, 2*, 1-29.
Dennis, I., Handley, S., Bradon, P., Evans, J., & Newstead, S. (2002). Approaches to modeling item-
generative tests. In S. H. Irvine & P. C. Kyllonen (Eds.), *Item generation for test development* (pp. 53-
Psychological Measurement, 36*, 255-270.
Psychologica, 37*, 359-374.
Geerlings, H. (2012). *Psychometric methods for automated test design* (Doctoral dissertation, University of
*Psychometrika, 76*, 337-359.
Glas, C. A. W., van der Linden, W. J., & Geerlings, H. (2010). Estimation of the parameters in an item-
cloning model for adaptive testing. In W. J. van der Linden & C. A. W. Glas (Eds.), *Elements of
adaptive testing* (pp. 289-314). New York, NY: Springer.


ILOG (2003). *CPLEX 9.0* [Computer program]. Incline Village, NV: ILOG.


