Measurement and Research Department Reports 96-4

Parallel Test Construction Using Classical Item Parameters

P.F. Sanders
A.J. Verschoor

Cito

Measurement and Research Department Reports 96-4

Parallel Test Construction Using Classical Item Parameters

P.F. Sanders

A.J. Verschoor

Cito
Arnhem, 1996

Abstract

In recent years, a number of mathematical programming models for test construction have been presented. Adema and Van der Linden (1989) developed models for maximizing test reliability. Models for the construction of parallel tests were developed by Van der Linden and Boekkooi-Timminga (1988). In this paper, a minimization and maximization model for parallel test construction under constraints are presented. The purpose of the minimization model is to construct weakly and strongly parallel tests of minimum length. The maximization model can be used to construct weakly and strongly parallel tests with maximum test reliability.

Index terms: test construction, weakly and strongly parallel tests, classical test theory, mathematical programming, greedy algorithms.

Introduction

Gulliksen (1950) opens his chapter on item selection for single tests with the following sentence: 'Basically, item analysis is concerned with the problem of selecting items for a test, so that the resulting test will have certain specified characteristics' (p. 363). These characteristics, the mean, variance, reliability, and validity of the test, require each item to have three parameters: (1) the item difficulty: the proportion of persons answering an item correctly; (2) the reliability index: the point-biserial correlation between item and total score multiplied by the item standard deviation; (3) the validity index: the point-biserial correlation between item and criterion score multiplied by the item standard deviation.

Gulliksen proposed item selection procedures for maximizing the validity and the reliability of a test. As validity indices are almost never available, maximization of the reliability of the test is usually pursued. With a fixed number of items, test reliability can be increased by either making the average item variance smaller or the average item reliability index larger as can be inferred from Equation 1 below. The item selection procedure proposed by Gulliksen (p. 379) for maximizing the reliability of the test is to plot the item variance as the ordinate and the reliability index as the abscissa and to select the items from the lower right-hand corner of the graph.

As an alternative to Gulliksen's graphical procedure, Adema and Van der Linden (1989) developed two mathematical programming models for maximizing test reliability subject to a fixed number of items and various practical constraints. Cronbach's coefficient alpha, a lower bound to the reliability of a test, was employed as the measure for test reliability. This coefficient is defined as

$$\alpha = n(n-1)^{-1}\left[1 - \left[\sum_{i=1}^{n}\sigma_i^2\right] \Big/ \left[\sum_{i=1}^{n}\sigma_i\rho_{ix}\right]^2\right], \qquad (1)$$

where n is the number of items in the test, σ_i^2 is the variance of item i, and ρ_{ix} is the point-biserial correlation between item i and the test score. Given an item bank of I items, the (second) model developed by Adema and Van der Linden is formulated as

$$\text{maximize} \quad \sum_{i=1}^{I}\rho_{ix}\,x_i, \qquad (2)$$

$$\text{subject to} \quad \sum_{i=1}^{I}x_i = n, \qquad (3)$$

and $$\sum_{i=1}^{I} r_i x_i \leq 35n, \qquad i = 1, \ldots, I, \tag{4}$$

$$x_i \in \{0, 1\}, \qquad i = 1, \ldots, I. \tag{5}$$

The common feature of mathematical programming models is that they all involve optimization, that is, the maximization or minimization of a quantity designated as an objective function. Although the purpose of the foregoing test construction model is to maximize reliability, reliability as such is not a component of the model. The model selects items with the highest item-test correlations since test reliability depends more on item discrimination than on item difficulty (Ebel, 1967, p. 127). The decision variables x_i employed by the model are used to indicate that item i is in the test, $(x_i = 1)$, or that item i is not in the test, $(x_i = 0)$. If the decision variable can only take the values 0 or 1, the mathematical programming model is called a binary or zero-one programming model. Note that the item selection process in the present model is governed by only two constraints. The constraint in Equation 3 specifies that the length of the test be equal to n. The constraint in Equation 4 is an example of a practical constraint in which the test constructor states that it takes r_i seconds to respond to item i but that the administration time of the test should not exceed $35n$ seconds. This constraint implies that items with a relatively short response time need to be selected. In this test construction model, the specification of the number of items will result in the selection of items that give the highest possible test reliability.

For the construction of parallel tests, Gulliksen developed the matched random subtests procedure (1950, p. 207 ff.) in which each item is represented by a point on a scatter diagram, the abscissa of which represents item difficulty and the ordinate, the item-test correlation. Items are first simultaneously matched on these item parameters and then one item of each matched pair, triple, etc., is randomly assigned to a test. Van der Linden and Boekkooi-Timminga (1988) developed a zero-one programming version of Gulliksen's two-step method. For the construction of two parallel tests, their model is formulated as

minimize $$\sum_{i=1}^{I-1} \sum_{j=i+1}^{I} \delta_{ij} x_{ij}, \tag{6}$$

subject to $$\sum_{i=1}^{j-1} x_{ij} + \sum_{i=j+1}^{I} x_{ji} = 1, \qquad j = 1, \ldots, I, \tag{7}$$

$$x_{ij} \in \{0,1\},\ i = 1,\ldots,I-1;\ j = i+1,\ldots,I. \tag{8}$$

In Equation 6, δ_{ij} is shorthand for $\delta_{ij} = [(\pi_i - \pi_j)^2 + (\rho_{ix} - \rho_{jx})^2]^{1/2}$ with π_i and π_j being the item difficulties of item i and item j. The binary decision variables x_{ij} denote whether i and j are pairs, that is,

$$x_{ij} = \begin{cases} 1 \text{ if } i \text{ and } j \text{ are a pair} \\ 0 \text{ otherwise} \end{cases} \qquad i = 1,\ldots,I-1;\ j = i+1,\ldots,I.$$

Van der Linden and Boekkooi-Timminga replace the first step of Gulliksen's graphical method by the objective function in Equation 6, that is, minimizing the sum of the within-pair Euclidean distances, and by the constraints in Equation 7, which guarantees that each item is part of exactly one pair of items. Since all the items in the item bank have to be assigned to the two tests, the solution set will result in $I/2$ pairs of items. Instead of randomly assigning items from pairs to tests, they show how the two tests can be matched further, for example, on their means and variances.

Gulliksen (p. 210) also proposed his matched random subtests procedure to solve the problem of constructing a second test to match a test already in use. The same problem was addressed by Armstrong, Jones, and Wu (1992) using binary programming techniques. Matching parallel tests to a pre-existing so-called seed test was defined by them as a bicriteria problem (p. 277). The first criterion is matching the items in the item bank to the items on the seed test as closely as possible. The second criterion is creating parallel tests with the smallest deviation from the seed test. Three classical item distances, including the distance defined in Equation 6, and two item response distances were employed. Their solution for this optimization problem consists of decomposing it into two network-flow problems. Armstrong, Jones, and Wang (1994) also developed a network-flow model for the problem of parallel test construction using classical test theory. The objective function of their model is to maximize reliability.

In the first section of this paper, two alternative models for parallel test construction, the minimization model and maximization model, are presented. The algorithms that were developed for these models and the computational results for various test construction problems are presented in the following two sections.

Minimization Model

The purpose of the minimization model is to construct parallel tests of minimum length under constraints. The minimization model for the construction of parallel tests, $t = 1, \ldots, T$, is formulated as

minimize n, (objective function) (9)

subject to

$$n = \sum_{i=1}^{I} x_{it}, \; t = 1, \ldots, T, \qquad \text{(definition of test length)} \qquad (10)$$

$$\sum_{t=1}^{T} x_{it} \leq 1, \; i = 1, \ldots I, \qquad \text{(no overlap between tests)} \qquad (11)$$

$$\alpha_t \geq \alpha^{\ell}, \; t = 1, \ldots, T, \qquad \text{(reliability constraints)} \qquad (12)$$

$$p^{\ell} n \leq \sum_{i=1}^{I} p_i x_{it} \leq p^{u} n, \; t = 1, \ldots, T, \qquad \text{(mean number-right score)} \qquad (13)$$

$$C_m^{\ell} n \leq \sum_{i=1}^{I} C_{mi} x_{it} \leq C_m^{u} n, \; t = 1, \ldots, T, \; m = 1, \ldots, M. \qquad \text{(category constraints)} \qquad (14)$$

The objective function in Equation 9 of the minimization model is the minimization of the length of the test, that is, the number of items. The constraints or restrictions in Equation 10 define test length as the number of items in the tests and also imply that all tests have the same length. The restrictions in Equation 11 specify that each item may be selected only once and that no overlap is allowed. The restrictions in Equation 12 specify a common lower bound, α^{ℓ}, for the reliability of the tests. The restrictions in Equation 13 allow the mean number-right score of the tests to vary in a prespecified range. For dichotomously scored items, the test mean is equal to the sum of the item difficulties, p_i. Since the number of items is not known in advance, the mean should be specified as a proportion of the maximum score that can be obtained. Content classifications or other taxonomic groups are formulated by using categories. For each item and each classification category, a constant is defined as

$$C_{mi} = \begin{cases} 1 \text{ if item } i \text{ in category } m \\ 0 \text{ if item } i \text{ not in category } m, \end{cases}$$

indicating whether ($C_{mi} = 1$) or not ($C_{mi} = 0$) item i belongs to category m. The constraints in Equation 14 define the specified division among the categories, where C_m^{ℓ} is the minimum fraction required for category m, and C_m^{u} is the maximum fraction required for category m.

Maximization Model

The purpose of the maximization model is to construct parallel tests with maximum test reliability and a test length specified by the test constructor. The maximization model for parallel test construction is formulated as

maximize ω, (objective function) (15)

subject to $\omega \leq \alpha_t$, $t = 1, \ldots, T$, (reliability constraint) (16)

$$\sum_{i=1}^{I} x_{it} = n, \; t = 1, \ldots T. \qquad \text{(equal test length)} \qquad (17)$$

The objective function in Equation 15 together with the constraint in Equation 16 of the maximization model ensure the maximization of the test with the lowest reliability. Note that the model does not specify a restriction to guarantee that all tests have the same reliability. The constraints in Equation 17 specify that all tests should have the same length. The maximization model will be extended by constraints similar to the constraints specified in the minimization model.

Greedy Algorithms

Two greedy algorithms, Quick and Strong, were developed to construct parallel tests with either the minimization or the maximization model. It is characteristic of greedy algorithms for test construction that they are iterative algorithms and that during every iteration, the 'best' item is assigned to a test and that this assignment is irreversible. For the kind of test construction problems discussed here with few and not very

complex restrictions, greedy algorithms have the advantage that computational time is negligible. Quick was developed for the construction of weakly parallel tests, that is, tests that have identical means, variances, and reliability coefficients. Strong was developed for the construction of strongly parallel tests, that is, tests that have identical item parameters for corresponding items and thus also identical test characteristics.
Every iteration of the Quick algorithm consists of three steps. In Step 1, it is decided to which test an item will be assigned to. If none of the tests to be constructed contains an item, a test will be selected at random to be the 'first' test. If each test contains only one item, the test with the item having the lowest discrimination index, r_{ix}, will be selected. If each test contains more than one item, the test with the lowest reliability coefficient, α, will be selected. In Step 2, an item will be assigned to the test resulting from the first step. This item has to have two features. First, the item should comply with the restrictions specified in Equations 13a and 14a:

$$p^{\ell} \cdot \sum_{i=1}^{I} x_{it} \leq \sum_{i=1}^{I} p_i x_{it} \leq p^{u} \cdot \sum_{i=1}^{I} x_{it}, \quad i = 1, \ldots I, \quad t = 1, \ldots T, \tag{13a}$$

$$C_m^{\ell} \cdot \sum_{i=1}^{I} x_{it} \leq \sum_{i=1}^{I} C_{m_i} x_{it} \leq C_m^{u} \cdot \sum_{i=1}^{I} x_{it}, \quad m = 1, \ldots M. \tag{14a}$$

Second, the assignment of the item to the test should result in a test having the highest α. In the case of an empty test, the item with the highest discrimination index will be assigned to the test. In the case of tests with one item, the test having the item with the lowest discrimination index will be assigned the item that yields the test with the highest α. Note that this item is not always the item with the highest discrimination index. In the case of tests with more than one item, the test having the lowest α will be assigned the item that yields the test with the highest α. In Step 3, the algorithm stops if all tests have an α larger than α^{ℓ}, in the case of the minimization model, or if all tests have n items, in the case of the maximization model, or if no item can be found in the second step. Otherwise, the algorithm continues with Step 1.

Every iteration of the Strong algorithm consists of four steps. Step 1 and Step 2 are the same as the first two steps of the Quick algorithm. In Step 3, for every test other than the test selected in the second step, an item is assigned which comes from the same taxonomic group as the item assigned to the test selected in the second step and which has item parameters which minimize the (sum of the) Euclidian distances within-pairs, within-triples, etc. In Step 4, the algorithm stops if all tests have an α larger than α^{ℓ},

or if all the tests have n items, or if the algorithm fails to assign an item to every test in the second or third step. Otherwise, the algorithm continues with the first step. An illustration of the construction of two tests with the Strong algorithm is presented in Table 2.

The following should be noted about the test construction models and the algorithms. Firstly, except for the stopping criterion, the algorithms for the minimization and maximization model do not differ. Secondly, Quick and Strong do not differ if a single test is constructed. Finally, the algorithms do not impose an upper-bound restriction on the reliability coefficient or restrictions on the standard deviation. Experience with the algorithms has shown that only rarely are the required restrictions, for example, 'equal' standard deviations, not met.

Computational Results

Generation of the Item Bank

An item bank of 500 dichotomous test items was generated with the two-parameter logistic model as the underlying item response model. Item parameters a_i and b_i were drawn from the distributions $U(0.7,1.5)$ and $N(-0.3,1.2)$, respectively. To estimate the classical item difficulties and item discriminations, 10,000 examinees ($\theta \sim N(0,1)$) were generated to answer the items. Item discriminations were first computed as item-test correlations, where the complete item bank was considered as a test, and then corrected by the correction formula developed by Henrysson (1963) to make them invariant to test length. Items were randomly assigned to four taxonomic groups (A, B, C, and D), resulting in approximately 125 items in each group.

Generation of Parallel Tests for the Minimization Model

For all tests, a lower bound of .90 was specified for the reliability constraints in Equation 12. For the mean number-right score constraints in Equation 13, three 'proportional means' were specified for the generation of two parallel tests and one for six parallel tests. (Note that in the tables with results, the symbol '$\approx$' is used to denote that bounds have been specified.) For all tests, an equal distribution of items over each of the four taxonomic groups was specified for the constraints in Equation 14. The characteristics of the tests generated by Quick and Strong are presented in Table 1.

Table 1

Characteristics of Generated Tests by Algorithms Quick and Strong for the Minimization Model

	Two Parallel Tests by Quick					
	Prop. Mean ≈ 0.50		Prop. Mean ≈ 0.60		Prop. Mean ≈ 0.70	
Test number	1A	1B	2A	2B	3A	3B
n	32	32	31	31	37	37
KR20	.90	.90	.90	.90	.90	.90
Mean	16.01	16.00	18.72	18.59	25.95	25.85
Sd	7.77	7.80	7.47	7.50	7.76	7.91
	Two Parallel Tests by Strong					
	Prop. Mean ≈ 0.50		Prop. Mean ≈ 0.60		Prop. Mean ≈ 0.70	
Test number	1A	1B	2A	2B	3A	3B
n	32	32	31	31	37	37
KR20	.90	.90	.90	.90	.90	.90
Mean	16.01	16.01	18.70	18.69	25.91	25.92
Sd	7.81	7.82	7.46	7.46	7.85	7.85
	Six Parallel Tests by Quick (Prop. Mean ≈ 0.60)					
Test number	1	2	3	4	5	6
n	42	42	42	42	42	42
KR20	.90	.90	.90	.90	.90	.90
Mean	25.19	25.28	25.15	25.42	25.22	25.33
Sd	8.84	8.79	8.86	8.89	8.89	8.83
A	10	10	10	10	10	10
B	11	11	11	11	11	11
C	11	10	10	11	10	10
D	10	11	11	10	11	11
	Six Parallel Tests by Strong (Prop. Mean ≈ 0.60)					
Test number	1	2	3	4	5	6
n	43	43	43	43	43	43
KR20	.90	.90	.90	.90	.90	.90
Mean	25.95	25.85	25.92	25.89	25.80	26.03
Sd	8.97	9.00	9.00	8.96	9.01	8.95
A	10	10	10	10	10	10
B	11	11	11	11	11	11
C	11	11	11	11	11	11
D	11	11	11	11	11	11

Table 2

Difficulty (p_i) and Discrimination Values (r_{ix}) for the Items in Test 2A and Test 2B

Test 2A			Test 2B		
Item	p_i	r_{ix}	Item	p_i	r_{ix}
365	.61	.51	448	.61	.50
204	.69	.49	310	.68	.50
402	.55	.50	410	.55	.50
464	.58	.49	182	.57	.50
434	.63	.48	167	.63	.49
468	.70	.49	385	.70	.48
376	.53	.46	239	.55	.46
196	.51	.49	383	.52	.49
74	.69	.49	281	.70	.48
43	.65	.48	55	.66	.48
343	.47	.48	67	.46	.50
194	.67	.48	352	.67	.48
318	.56	.47	367	.52	.47
435	.62	.47	208	.61	.48
147	.73	.48	321	.73	.46
256	.48	.49	322	.48	.49
20	.65	.47	247	.65	.48
201	.51	.49	190	.51	.49
248	.60	.47	315	.60	.45
273	.63	.47	325	.64	.48
159	.65	.47	4	.66	.46
100	.73	.47	300	.74	.45
308	.43	.48	298	.43	.49
36	.58	.47	473	.58	.47
408	.70	.46	474	.71	.47
440	.57	.46	69	.58	.46
307	.49	.46	110	.48	.48
287	.63	.46	15	.62	.44
115	.73	.45	89	.74	.44
10	.45	.46	28	.44	.47
394	.70	.45	500	.69	.45

The results in Table 1 show that the differences between parallel tests generated by either Quick or Strong are quite small. Also note the small differences between the standard deviations which were not constrained by the model. As expected, the differences between tests generated by Strong are smaller than those generated by Quick. These differences are related to the statistical as well as the taxonomic characteristics. In order to comply with the test specifications, Strong requires one item more than Quick, for example, in generating six parallel tests. It should be noted that if Quick is used for the minimization model, tests of different length may be generated. For example, the specified reliability coefficient for test 1 will be obtained with 35 items, while 36 items are required for test 2. If this occurs and the test constructor wants the tests to be of equal length, the shortest test should be assigned one more item. In general, this results in negligible differences between the test parameters.

The test generation process performed by Strong of parallel tests 2A and 2B is illustrated in Table 2. Table 2 shows that item 365, that is, the item with the highest r_{ix} in the item bank, was the first item selected for test 2A and item 448 was the first item selected for test 2B. In the second iteration, test 2B has the lowest 'reliability', that is, one item with r_{ix} = .50. Therefore, item 310 was assigned to this test in step 2. In step 3, the item with the shortest distance to item 310, item 204, was assigned to test 2A. The other items in Table 2 are also presented in the order in which they were selected. A graphical display of Table 2, a 'Gulliksen plot', is shown in Figure 1.

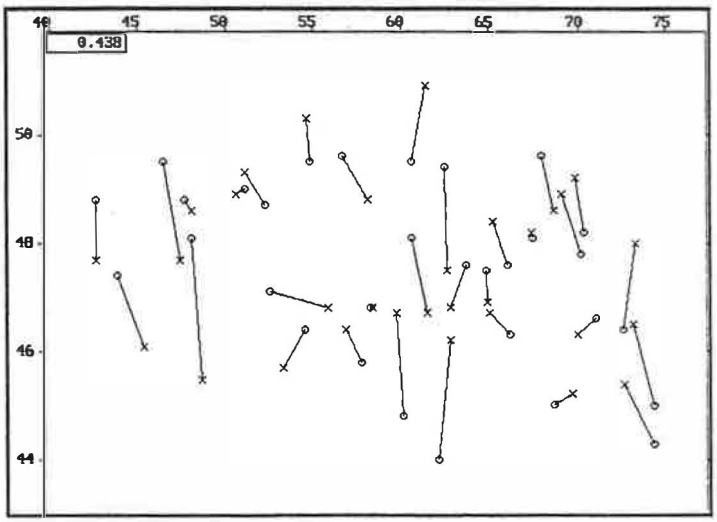

Figure 1

Gulliksen Plot of Test 2A and Test 2B (Minimization Model)

The horizontal axis in Figure 1 represents item difficulty and the vertical axis, item discrimination, respectively. Items assigned to Test 2A and Test 2B are indicated by crosses and open circles, respectively. The top left-hand corner contains the value of the sum of the Euclidian within-pair distances which can be used to compare the parallelism of strongly parallel tests. Note that the line farthest to the left in this figure pairs item 298 with item 308. As can be seen in Table 2, item 308 has been assigned to test 2A, while item 298 has been assigned to test 2B.

Generation of Parallel Tests for the Maximization Model

The specifications for the constraints in Equations 12, 13, and 14 used for the minimization model were also used for the maximization model. The constraints in Equation 17 were specified to be 40 items. The characteristics of the tests generated by Quick and Strong are presented in Table 3.

Table 3 shows that the differences between parallel tests generated by either Quick or Strong increase as more parallel tests are generated. The differences between tests generated by Strong are smaller than the differences between tests generated by Quick. As expected, the quality of the parallel tests decreases if more tests are generated.

Test Information and Efficiency Functions

Since the item bank was generated with the two-parameter logistic model, the framework of item response theory was used to investigate the parallelism of the tests generated by the minimization and maximization model and the algorithms Quick and Strong. In item response theory, parallel tests are defined in terms of information functions (Samejima, 1977). Tests are defined as weakly parallel tests when they measure the same ability and have the same test information functions. Strongly parallel tests are defined as tests in which corresponding items in the tests have identical item information functions and thus identical test information functions. The test information function for the two-parameter logistic model is defined as $I(\theta) = \sum_{i=1}^{n} a_i^2 P_i(\theta) Q_i(\theta)$. Test information functions can be compared by computing $RE(\theta) = I_1(\theta) / I_2(\theta)$, where $RE(\theta)$ denotes relative efficiency and $I_1(\theta)$ and $I_2(\theta)$ are the information functions for test 1 and 2, respectively.

Table 3

Characteristics of Generated Tests by Algorithms Quick and Strong for the Maximization Model

	Two Parallel Tests by Quick					
	Mean ≈ 20		Mean ≈ 24		Mean ≈ 28	
Test number	1A	1B	2A	2B	3A	3B
KR20	0.92	0.92	0.92	0.92	0.91	0.91
Mean	19.99	20.05	24.01	24.21	27.99	27.99
Sd	9.58	9.56	9.44	9.40	8.44	8.40
	Two Parallel Tests by Strong					
	Mean ≈ 20		Mean ≈ 24		Mean ≈ 28	
Test number	1A	1B	2A	2B	3A	3B
KR20	0.92	0.92	0.92	0.92	0.91	0.91
Mean	20.00	19.99	23.99	24.01	28.00	28.00
Sd	9.54	9.56	9.41	9.42	8.38	8.38
	Six Parallel Tests by Quick (Mean ≈ 24)					
Test number	1	2	3	4	5	6
KR20	0.90	0.90	0.90	0.90	0.90	0.90
Mean	24.29	24.01	24.03	24.05	24.15	24.09
Sd	8.53	8.50	8.53	8.48	8.55	8.48
	Six Parallel Tests by Strong (Mean ≈ 24)					
Test number	1	2	3	4	5	6
KR20	0.90	0.90	0.90	0.90	0.90	0.90
Mean	24.03	23.99	24.07	24.00	24.01	23.99
Sd	8.50	8.52	8.50	8.52	8.54	8.56

For the minimization model of generating two parallel tests by Strong (see Table 1), the test information functions of tests 1A, 2A, and 3A, and the relative efficiency functions (dotted lines) of these tests with respect to parallel forms 1B, 2B, and 3B, respectively, are displayed in Figure 2.

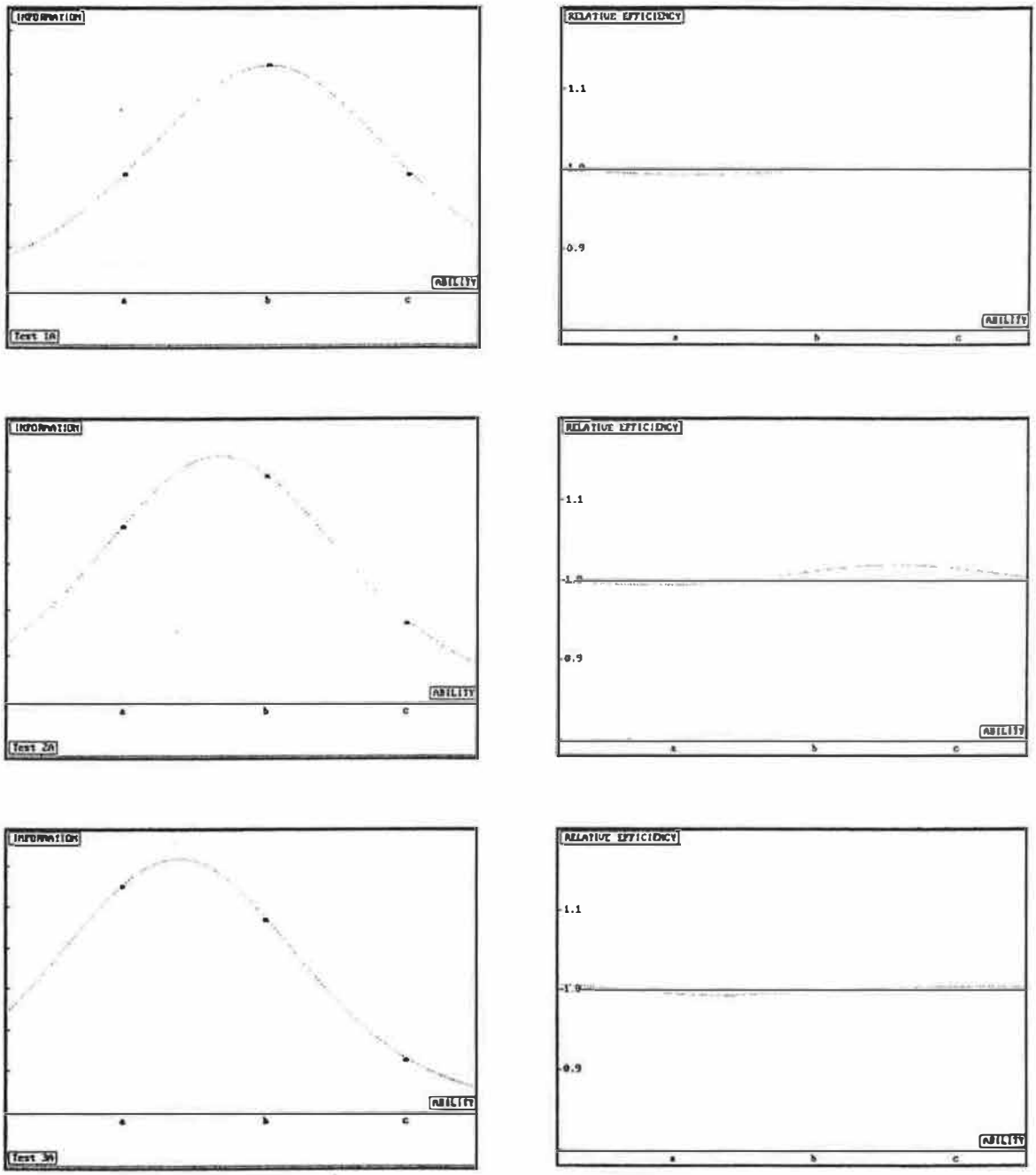

Figure 2

Test Information Functions of Test 1A, Test 2A, and Test 3A and Relative Efficiency Functions of Test 1A and Test 1B, Test 2A and Test 2B, and Test 3A and 3B (Minimization Model; Algorithm Strong)

Note that the letters 'a', 'b', and 'c' on the θ-axis of the test information functions denote $\theta = -2.25$, $\theta = 0$, and $\theta = +2.25$, respectively. Figure 2 shows that while the three tests have quite different test information functions, the relative efficiency

functions indicate that the test information functions of the corresponding parallel forms are the same.

For the maximization model of generating six parallel tests by Strong, the relative efficiency function of the two most dissimilar tests, test 3 and test 6, is shown by the dotted line in Figure 3.

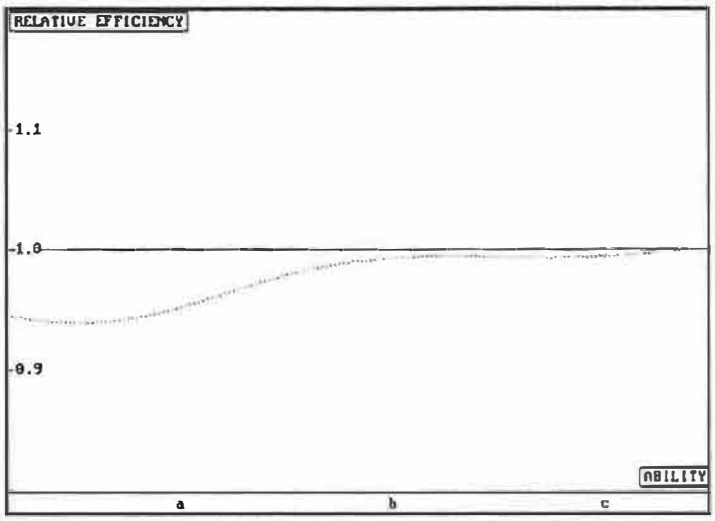

Figure 3

Relative Efficiency of Test 3 Compared to Test 6
(Maximization Model; Algorithm Strong)

It can be inferred from Figure 3 that the tests are not parallel at $\theta = -2.25$. Test 3 is functioning as if it were 5%, that is, two items, longer than test 6. In terms of item response theory the tests are locally parallel between $\theta = 0$ and $\theta = +2.25$.

Conclusions

The results presented in this article show that within the framework of classical test theory, weakly and strongly parallel tests with specified statistical characteristics and taxonomic composition can be successfully generated with the minimization and maximization model and heuristic algorithms Quick and Strong.

The minimization and maximization models for parallel test construction presented here can be considered both an integration and extension of the models discussed in the introduction. These models are not only useful in providing an interface between

calibrated item banks and test constructors not familiar with concepts from item response theory, but can also be very useful in situations where classical test theory applies.

A computer program has been developed to solve the test construction problems discussed here. The figures presented in the paper are samples of the output from the program. The construction of six weakly parallel tests took less than a second, while the construction of six strongly parallel tests took three seconds using a 486DX2 66 MHz computer. Information about the program can be obtained from the authors.

References

Adema, J.J., & van der Linden, W.J. (1989). Algorithms for computerized test construction using classical item parameters. *Journal of Educational Statistics, 14*, 279-289.

Armstrong, R.D., Jones, D.H., & Wu, I.L. (1992). An automated test development of parallel tests from a seed test. *Psychometrika, 57, 2,* 271-288.

Armstrong, R.D., Jones, D.H., & Wang, Z. (1994). Automated parallel test construction using classical test theory. *Journal of Educational Statistics, 19,* 73-90.

Ebel, R.L., (1967). The relation of item discrimination to test reliability. *Journal of Educational Measurement, 4,*125-128.

Gulliksen, H. (1950). *Theory of mental tests.* New York: Wiley. [Hillsdale: Erlbaum, 1987.]

Henrysson, S. (1963). Correction of item-total correlations in item analysis. *Psychometrika, 28,* 211-218.

Samejima, F. (1977). Weakly parallel tests in latent trait theory with some criticisms of classical test theory. *Psychometrika, 42,* 193-198.

van der Linden, W.J., & Boekkooi-Timminga, E. (1988). A zero-one programming approach to Gulliksen's matched random subtests method. *Applied Psychological Measurement, 12,* 201-209.

Recent Measurement and Research Department Reports:

96-1 H.H.F.M. Verstralen. Estimating Integer Parameters in Two IRT Models for Polytomous Items.

96-2 H.H.F.M. Verstralen. Evaluating Ability in a Korfball Game.

96-3 T.J.H.M. Eggen & G.J.J.M. Steatmans. Computerized Adaptive Testing for Classifying Examinees Into Three Categories