Item Characteristics of Tests Constructed by Linear Programming

Frank B. Baker, Alan S. Cohen, and B. Ross Barmish
University of Wisconsin

In the present paper, linear programming was used to select items from item pools based on one-, two-, and three-parameter models so that a target test information function was reached. The primary interest was in the distributional characteristics of the items thus selected. The results suggest that the linear programming approach focuses on the "worst feature" of the target information function (i.e., the extremes of a uniform target and the maximum of a peaked target). The values of the parameters of the selected items tend to form clusters. For uniform targets, these clusters are associated with the extremes of the target range, whereas for peaked targets they are associated with the maximum of the target. Selecting items from an item pool by linear programming appears to be a useful addition to the test constructor's repertoire. However, additional refinement may be needed to obtain a specific distribution of item parameters for a given test. *Index terms: Item response theory, Item selection, Linear programming, Target information function.*

Under item response theory (IRT), one step in test construction is to specify a target information curve corresponding to the particular goal of testing. For example, a screening test might employ a narrow highly peaked curve, while a broad-range test might employ a uniform target. Given a target information curve, the task faced by the test constructor is to select items from a precalibrated item pool such that the sum of the item information functions closely approximates the target information curve over the ability (θ) range of interest. In addition, it would be preferable to select as few items as possible to reduce the amount of time needed for testing.

Lord (1980, p. 23) suggested a "cut and fill" heuristic for accomplishing this selection task. The problem with this heuristic, even when using a computer, is that the number of possible item combinations to try can be exceedingly large even for modest-size item pools. In addition, there are no viable rules for selecting items in a given context. As a result, the process has the potential for being very laborious or computationally intractable. Furthermore, the number of items selected under such an approach is not guaranteed to be a minimum.

In a recent paper, Theunissen (1985) used a branch-and-bound algorithm of integer programming to perform this selection task. In this approach, a system of linear equations is used to determine the subset of items in the item pool that can be used to match or exceed the target information curve. Using

APPLIED PSYCHOLOGICAL MEASUREMENT
Vol. 12, No. 2, June 1988, pp. 189–199
© Copyright 1988 Applied Psychological Measurement Inc.
0146-6216/88/020189-11$1.80

several item characteristic curve (ICC) models and several item pool sizes, he was able to approximate a number of different targets. In a subsequent paper (Theunissen, 1986), the mathematical programming approach was extended to include a wide variety of constraints over and above target information function reachability. However, few empirical results were reported for these extensions.

Mathematical programming represents a major addition to the tools available to the test constructor because it provides an analytical rather than an ad hoc procedure for item selection under IRT. Given the recency of this approach, there is limited empirical evidence regarding the application of the mathematical programming approach to the test construction process. The bulk of the existing empirical results have focused on the minimum number of items needed to reach or exceed a target information function. Thus the goal of the present paper was to extend these results by focusing on the item composition of the tests created by this approach. The results obtained from item pools defined under the three logistic ICC models were compared for a number of different target information functions. Particular attention is paid to the properties of the algorithmic selection procedure and the distributional characteristics of the parameters of the items selected from the item pool.

The Linear Programming Approach

It is well known that the computational demands of integer programming are very high and hence only small item pools are feasible. Consequently, item selection techniques yielding results approximating the integer programming solutions are needed if the general approach is to be of practical value. Boomsma (1986) evaluated a number of mathematical programming algorithms and heuristic item selection methods using various target information curves and an item pool based on the Rasch model. He concluded that the simplex algorithm of linear programming was the ''best'' in terms of the minimization of the number of items selected and the amount of computer time required.

A problem with the linear programming solution is that although fractional weights can be assigned to some items, in an actual test the number of items must be an integer. Arguing along the lines given by Dantzig (1963), Boomsma (1986) derived a theorem stating that the number of items obtaining fractional weights cannot exceed m, the number of points at which the target curve is specified. Therefore, under the linear programming approach the bulk of the items will receive integer weights and the linear programming solution should be a good approximation to the integer programming solution. Motivated by Boomsma's results, the present study employed the simplex algorithm of linear programming for item selection.

Within the framework of this paper, a precalibrated pool of N items, based on a particular ICC model, is assumed to exist. Given this, the test constructor then defines a target information curve by specifying m points θ_k ($k = 1, 2, \ldots, m$) along the θ scale and the target amount of test information $I(\theta_k)$ at each point. It is then necessary to compute the amount of information at each of the m θ levels for all N items in the item pool. The linear programming technique minimizes the function

$$I_z = X_1 + X_2 + X_3 + \ldots X_N \tag{1}$$

subject to the constraints

$$\begin{array}{l}
I(\theta_{1,1})\,X_1 + I(\theta_{1,2})X_2 + \ldots I(\theta_{1,N})\,X_N \geqslant I(\theta_1) \\
I(\theta_{2,1})\,X_1 + I(\theta_{2,2})X_2 + \ldots I(\theta_{2,N})\,X_N \geqslant I(\theta_2) \\
\cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \\
\cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \\
\cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \\
I(\theta_{m,1})X_1 + I(\theta_{m,2})X_2 + \ldots I(\theta_{m,N})X_N \geqslant I(\theta_m) \quad ,
\end{array} \tag{2}$$

where $0 \leqslant X_i \leqslant 1$ are the item weights in the solution, and $I(\theta_{k,i})$ is the amount of information for item i

at θ_k. Linear programming introduces surplus variables, Y_k, on the left sides of the constraints to eliminate the inequalities. The simplex algorithm is then used to determine the item weights that will define the minimum set of items needed to match or exceed the target information curve (see Feiring, 1986, for a tutorial presentation of the simplex algorithm). The items not included in the selected set receive a weight of 0, and those used receive a weight greater than 0 and less than or equal to 1.

Design of the Study

Following Theunissen (1985) and Boomsma (1986), item pools based on the three logistic ICC models were established. To accomplish this, a simulated pool of 500 items based on a three-parameter logistic model was generated using program GENIRV (Baker, 1978). The obtained item difficulties were normally distributed with a mean of $-.03$ and standard deviation of 1.14. The item discriminations were normally distributed with a mean of 1.18 and standard deviation of .41 in a logistic ogive metric. The c parameter was normally distributed with a mean of .23 and standard deviation of .041 and was truncated at 0 and .35.

Two additional item pools were derived from this three-parameter pool. The first of these was obtained by using the 500 item difficulties from the original pool to define an item pool under the Rasch model. The second of the additional pools was generated by using the 500 pairs of item difficulty and discrimination values from the original pool to define an item pool under a two-parameter logistic model. While these three item pools shared the numerical values and distributions of their common parameters, the obtained results are not directly comparable because the item information functions are different under the three models.

To investigate the role of the width of the target information curves, they were defined for three ranges on the θ scale: $(-1,1)$, $(-3,3)$, and $(.5,1.5)$. The range of .5 to 1.5 was used to determine the effect of a target that was noncentral relative to the mean difficulty of the item pool. In all three cases, 11 equally spaced points were used within the θ range specified. Two different forms of the target curves, uniform and peaked, were used, corresponding respectively to the goals of a broad-range test and a screening test. The desired standard errors of the θ estimates were set at .30, .25, and .20, corresponding to target information levels of $I = 10$, 16, and 25 respectively.

The uniform targets employed information levels of either $I = 10$ or $I = 16$. The peaked targets were constructed in such a way that their form approximated the shape of a normal curve with a span of 5 standard deviations matched to the width of the target. Each target had a maximum $I = 25$ at mid-range. Thus, the shorter the range of the target, the more sharply peaked was the target curve. The total number of combinations of target form, level, and range with the three item pools provides a reasonably wide set of situations to be examined.

Results

The BLPX87 computer program (Eastern Software Products, 1985) for the IBM PC/AT microcomputer, equipped with an 80287 co-processor chip and an 8 MHz clock, was used in the present study. The simplex algorithm involved 500 variables, 11 constraints, and 11 surplus variables. Given an item pool, a FORTRAN program was used to compute the amount of item information at the 11 θ levels for each of the 500 items in the item pool and to store the values on disk. This required about two minutes of computer time. The 500×11 matrix was then read from disk by the BLPX87 program, which required about 13 minutes. The actual selection of items by the simplex algorithm is a function of the number of items selected, and required an amount of time ranging from 2 minutes 23 seconds (for 23 items) to 17 minutes (for 192 items). Thus, the entire process can be accomplished in a reasonable amount of time on a microcomputer.

Table 1 contains the summary data for the 27 tests resulting from the item selection procedures conducted. The results reported are based on inclusion of the fractionally weighted items. This was done because the simplex algorithm used these items in order to just exceed the target information curve. However, the contribution of these items to the reported results was not fractionally weighted. As a result, the obtained test information curves were slightly higher than those based on the fractionally weighted items. Coupled with the discreteness of the values of the item discrimination indices, this also had a minor role in increasing the values of the root mean squares (RMSs) of the differences between the obtained and target information functions. It was felt that retaining the items more closely approximated what would be done in practice. However, other rules for dealing with fractionally weighted items could have been employed. The numbers of fractionally weighted items are reported for each test; they generally constituted a small fraction of the number of items in a given test.

When the one-parameter item pool was used to match a uniform target, the range of the target curve had a large impact on the number of items selected. At a target information level of $I = 10$, the range $(-1,1)$ required 52 items while the range $(-3,3)$ required 104. The noncentral target, which spanned only one unit on the θ scale, required somewhat fewer items (45) than did the $(-1,1)$ range. Increasing the information level of the uniform target from $I = 10$ to $I = 16$ increased the number of items selected to 83, 192, and 70 for the three target ranges respectively. Under the Rasch model, the maximum amount of information was $I = .25$ for all items. Hence, for a sufficiently large item pool, the number of items selected is proportional to the level of the uniform target. The RMS of the difference between the obtained and the uniform target information curves was the largest, 9.21, when the range was $(-3,3)$ for $I = 16$. The noncentral range at target levels of $I = 10$ and $I = 16$ required the least number of items and yielded the smallest RMS values, .76 and .83.

When a peaked target was used, the target was more sharply peaked for smaller widths of the target curve, but the maximum amount of information was the same for all widths. For this reason, all three peaked target ranges required virtually the same number of items (102, 102, 103). When the range was $(-1,1)$ the RMS was 13.47; for $(-3,3)$ the RMS was 3.38. Inspection of these two tests revealed that in both cases the same 102 items were selected. The noncentral peaked target required 103 items and yielded RMS = 15.58; the items generally differed from the other two tests.

When the two-parameter item pool was used to match a uniform target, the range of the target curve again had a large impact on the number of items selected. The range $(-1,1)$ with $I = 10$ required 23 items, while the range $(-3,3)$ needed 71 items to match or exceed the target. At a level of 16, the numbers of items required were 37 and 151 respectively for these ranges. The noncentral uniform target required very few items to match or exceed the target, 17 for $I = 10$ and 28 for $I = 16$.

The RMSs were largest when the target range was $(-3,3)$: 3.80 and 9.47 for $I = 10$ and $I = 16$, respectively. The smallest RMS (1.43) was found for the $(-1,1)$ range at $I = 10$. When a peaked target was used, all three target ranges required a relatively small number of items (37, 45, and 35). However, the RMSs differed considerably. For the target ranges of $(-1,1)$, $(-3,3)$, and $(.5,1.5)$, RMS values were 10.52, 3.38, and 14.32, respectively. This pattern was similar to that found with the one-parameter item pool.

When the three-parameter item pool was used to match a uniform target curve, the range of the target curve had a dramatic impact on the number of items selected. At $I = 10$, the target range $(-1,1)$ required 38 items. Inspection of the obtained information curves for the $(-3,3)$ range, at $I = 10$ and $I = 16$, showed that the selection algorithm was unable to match or exceed the target at the upper and lower ends of the target range. Due to the unavailability of items near the extremes of the range, it kept selecting items until the item pool was exhausted. In contrast, the noncentral uniform target required only

Table 1
Summary Statistics for Item Selection

Analysis Set	Model	Form	Target Level	Range	No. of Items Selected	No. of Fractional Weights	RMS	Maximum Difference	$\bar{b}$	S_b	$\bar{a}$	S_a	$\bar{c}$	S_c
1	1	u	10	-1,1	52	2	1.92	2.78	.00	.30				
2	1	u	10	-3,3	104	2	2.61	3.79	.01	2.00				
3	1	u	10	.5,1.5	45	2	.75	.98	.95	.43				
4	1	u	16	-1,1	83	2	3.06	4.43	.00	.27				
5	1	u	16	-3,3	192	2	9.21	12.43	.02	1.68				
6	1	u	16	.5,1.5	70	2	.83	1.15	.97	.36				
7	1	p	25	-1,1	102	1	13.47	19.19	-.04	.27				
8	1	p	25	-3,3	102	1	3.38	4.82	-.04	.27				
9	1	p	25	.5,1.5	103	1	15.58	23.15	.90	.33				
10	2	u	10	-1,1	23	2	1.43	1.90	-.10	.93	1.86	.39		
11	2	u	10	-3,3	71	4	3.80	8.10	-.04	2.01	1.39	.28		
12	2	u	10	.5,1.5	17	2	1.86	2.57	.80	.70	1.74	.42		
13	2	u	16	-1,1	37	2	3.22	4.30	-.06	.84	1.84	.32		
14	2	u	16	-3,3	151	2	9.47	14.38	-.05	1.76	1.23	.29		
15	2	u	16	.5,1.5	28	2	2.97	4.00	.90	.60	1.73	.34		
16	2	p	25	-1,1	37	1	10.52	15.46	.04	.46	1.72	.30		
17	2	p	25	-3,3	45	5	1.08	1.88	-.03	.79	1.82	.29		
18	2	p	25	.5,1.5	35	1	14.32	22.37	.79	.52	1.75	.31		
19	3	u	10	-1,1	38	2	1.41	1.93	-.25	.86	1.78	.32	.21	.04
20	3	u	10	-3,3	500	0	43.40	70.04	-.03	1.14	1.18	.41	.23	.04
21	3	u	10	.5,1.5	26	2	1.81	2.58	.71	.56	1.77	.35	.22	.04
22	3	u	16	-1,1	65	2	2.50	3.26	-.30	.85	1.72	.27	.22	.04
23	3	u	16	-3,3	500	0	38.52	64.04	-.03	1.14	1.18	.41	.23	.04
24	3	u	16	.5,1.5	45	2	2.63	7.71	.77	.49	1.70	.29	.22	.04
25	3	p	25	-1,1	66	1	10.56	15.68	-.15	.46	1.65	.27	.23	.05
26	3	p	25	-3,3	78	5	.42	.90	-.31	.83	1.73	.23	.22	.04
27	3	p	25	.5,1.5	61	1	14.28	21.42	.68	.45	1.67	.27	.23	.04

26 and 45 items at $I = 10$ and $I = 16$, respectively. With the exception of the two tests involving the $(-3,3)$ range, the RMS ranged from 1.41 to 2.63. When a peaked target was used in conjunction with a three-parameter item pool, all three target ranges required a similar number of items (66, 78, and 61) and the observed RMS exhibited a pattern similar to that of the previous two item pools. For the ranges $(-1,1)$ and $(-3,3)$, the RMS values were 10.56 and .424, respectively. Again, the obtained information curves were a better match to the broader peaked target curve.

Graphs of the obtained and the target information curves provide some insight into the location of the maximum differences reported in Table 1. Figure 1a is based on the one-parameter item pool and a target information curve with $I = 16$ over the range $(-3,3)$ (analysis set 5). To meet this target, 192 items were selected by the simplex algorithm. It is the worst of the 6 fits to a uniform target for this item pool. The obtained information curve only matches the target at the extremes of the target range; elsewhere it is above the target. The maximum difference between the curves was 12.43.

Figure 1b is based on the two-parameter item pool and a uniform target information curve with $I = 16$ over the range $(-1,1)$ (analysis set 10). The simplex algorithm selected 37 items. Again, the extremes of the obtained curve match the target; the remainder of this curve exceeds the target, with a maximum difference of 4.3.

Figure 1c is based on the three-parameter item pool and a peaked target over the range (.5,1.5) (analysis set 27). The obtained information curve was not a good match to the peaked target (RMS = 14.28). However, the obtained information curve matched the target at the peak of 25. As the θ level departed from that of the peak, the difference between the two curves increased to a maximum of 21.24. Across the three item pools, all but one of the nine information curves for the peaked targets had this characteristic. The exception was the test selected from the three-parameter item pool for a target range of $(-3,3)$. In this case, an excellent match to the target, RMS = .42, was obtained.

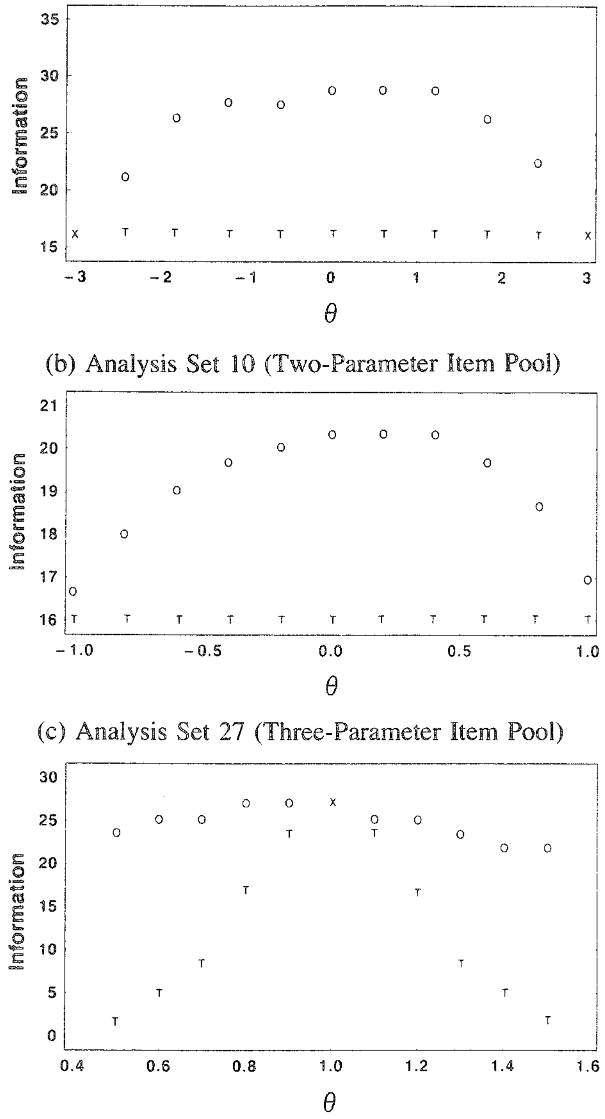
The means and standard deviations of the item parameters of the items selected for each test were reported in Table 1. The mean difficulty of the selected items was roughly the same as the midpoint of the range of the target curve. In the case of the three-parameter item pool, the mean difficulty was slightly below the midpoint value. The magnitudes of the standard deviations of the difficulties tended to be a function of the range of the target curve. However, the peaked targets tended to have, on average, the smallest standard deviations for the same target ranges.

As would be expected, the mean item discrimination parameters of items selected for the two- and three-parameter cases tended to be quite high. The standard deviations of the discrimination parameters ranged from .28 to .42. The magnitude of the standard deviations of the discrimination indices did not seem to be related to the form of the target curve. When the three-parameter item pool was used, the mean of the c parameter for the selected items varied from .21 to .23. Eight of the standard deviations of the c parameters had a value of .04; the other had a value of .05.

Bivariate plots of the parameters of the items selected from the several item pools were revealing. Figure 2a shows the item difficulties for the selected items from the one-parameter item pool and the uniform target curve depicted in Figure 1a (analysis set 5). A clear bimodal pattern exists. The bulk of the 192 selected items fell in two clusters, one centered at a difficulty of roughly -2 and another at $+1.7$. In addition, a single item was found at $-.2$ and there were several isolates at the extremes. This pattern apparently results from the simplex algorithm attempting to match the upper and lower extremes of the target curve.

Figure 2b plots the discrimination parameters against the difficulty parameters for items selected from the two-parameter item pool, and corresponds to Figure 1b above (analysis set 10). Again two clusters are apparent, with one centered near a difficulty of -1 and another near a difficulty of $+1$. The interesting feature of this plot is the three items having discriminations near 1 whose difficulties are approximately equally spaced over the θ scale. It appears that these three lower-discrimination items are

Figure 1
Obtained and Target Information Curves For Three Analysis Sets
(T = Target Curve, O = Obtained Curve, X = Matched Values)


(a) Analysis Set 5 (One-Parameter Item Pool)

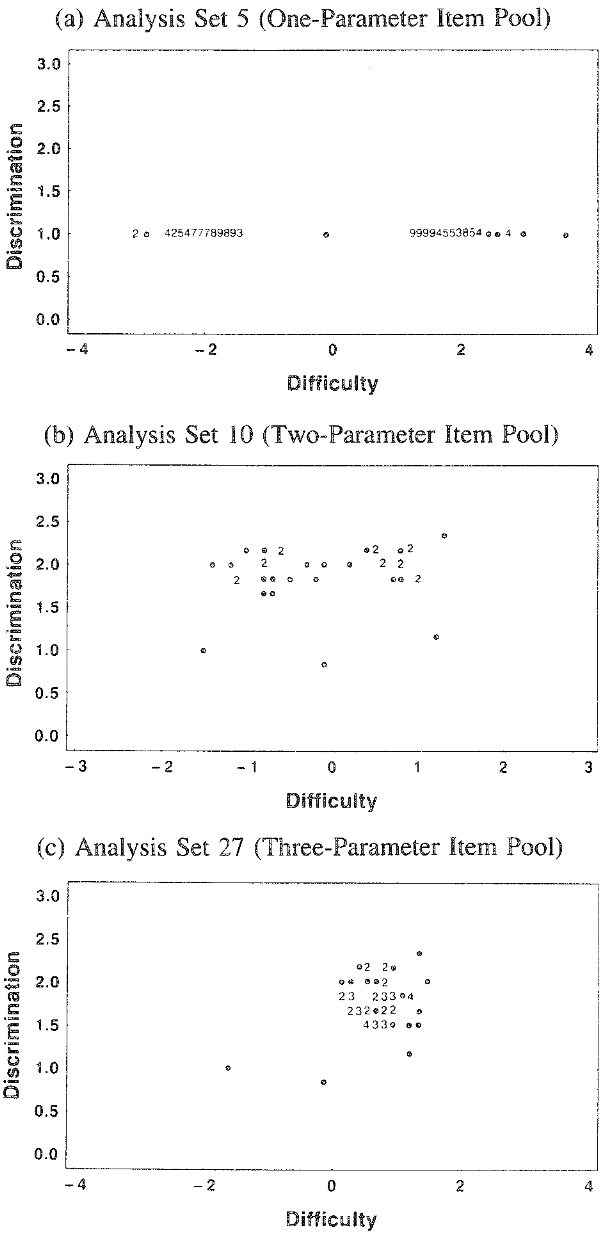
(b) Analysis Set 10 (Two-Parameter Item Pool)

(c) Analysis Set 27 (Three-Parameter Item Pool)

"filler" items that are used to raise the general level of the obtained information curve. Most of the tests obtained under the two- and three-parameter models for uniform targets had such items.

Figure 2c plots the discrimination parameters against the difficulty parameters for the 61 items from the three-parameter item pool involved in the peaked target of Figure 1c (analysis set 27). The pairs of parameter values form a cluster centered on a difficulty of about .7 and a discrimination of about 1.7. Again three filler items are clearly present in the plot.

Figure 2
Difficulty and Discrimination Indices for Items
Selected in Three Analysis Sets
(● is Single Observation, Number is Frequency of Occurrence)



(a) Analysis Set 5 (One-Parameter Item Pool)

(b) Analysis Set 10 (Two-Parameter Item Pool)

(c) Analysis Set 27 (Three-Parameter Item Pool)

Discussion

The focus of the present paper was on the distributional characteristics of the items selected to reach a target information function. The simplex algorithm of linear programming appeared to work quite well, and the number of items selected to form a test generally fell within the range of values employed in practice. In terms of the number of items selected, the two- and three-parameter models uniformly required fewer items than did the one-parameter model, even though the RMSs were quite similar over all conditions. The apparent cause is that all items in the one-parameter pool had a discrimination of unity. For this reason, the amount of information for an item is limited and more items are required to reach a target.

Additionally, filling in the target can only be accomplished by selecting items on the basis of difficulty. In contrast, under the two- and three-parameter models, items with larger maximum amounts of information are available and can be used to fill in the target. Even though larger item pools were used in the present study, the numbers of items selected to meet the targets were very similar to those found by Boomsma (1986) and Theunissen (1985). This is in large part due to the similarity of the item pools and targets.

The observed RMSs were generally quite small, and plots showed that the obtained information function generally exceeded the target curve. Thus, the precision of the θ estimates would be greater than that specified. One component of this excess was the manner in which the fractionally weighted items were handled. Another component is the finite set of values of the item discrimination indices available in the item pool.

When uniform target curves were specified, the maximum difference between the obtained and target curves occurred in the middle portion of the θ range covered by the target. In the case of peaked targets, the opposite held. The obtained information curve matched or slightly exceeded the peak of the target; as the θ point was moved away from the peak, the discrepancy increased. When the target was sharply peaked, the observed curve was also peaked, but it could not match the rapid decrease in the target. As a result, the RMS increased, but the bulk of its value was from the tails. In the central portion, the match was generally quite good. When a peaked target covered a broad range of θ, the obtained curve followed it quite closely and the RMS was small.

For a given item pool, both the number of items selected and the magnitude of the RMS appear to be related to the width of the target curve. The widths employed were 1, 2, and 6 θ units. The number of items selected was a monotonically increasing function of the width of the target except in the case of the one-parameter item pool, when a peaked target was employed. In this case, matching the peak was the driving force and the same number of items was selected for all three widths.

The three-parameter results for the uniform target with a range of $(-3,3)$ illustrate an obvious limitation of the selection process. In this case, there was not enough item information in the item pool to match or exceed the extremes of the target curve. As a result, the simplex algorithm continued to select items in an attempt to match the tails of the target. Eventually, the item pool was simply depleted. Because of this, the author's computer program reports the total amount of information in the item pool at each of the *m* points of the target curve prior to selecting items. An initial check could be made and the analysis aborted if failure is foreseen.

Although only a few plots of the parameters of the selected items were presented above, a set of well-defined patterns existed in the overall results. In the case of the one-parameter item pool and uniform targets, a clustering of item difficulties near the extremes of the target ranges was found. This differed from the results of Boomsma (1986), where the number of items located at a θ level increased sharply as the extremes of the target range were approached. This difference in results may be due to the larger number of target points employed in the present study (11), compared to the 5 used by Boomsma.

Both the two- and three-parameter results exhibited a distinct clustering of selected items with respect to item difficulty and discrimination. Broad, uniform targets yielded two widely spaced clusters with respect to difficulty and a narrow range of discrimination within each cluster. This clustering effect is somewhat contrary to the usual test construction practice, where the test is loaded with items at the center of the target. However, the narrower the width of the target, the more central were the clusters with respect to item difficulty, but discrimination appeared unaffected by the width. With peaked targets, a single centrally located cluster was observed.

When the two- and three-parameter item pools were used, the simplex algorithm appeared to use the most discriminating items whose difficulties were within or near the range of the target. The average discrimination of the selected items was generally substantially higher than the mean discrimination of the item pools. In general, the smaller the number of items selected, the higher the average level of discrimination of the selected items. Perhaps the most interesting result was the existence of a small number of filler items. These lower-discrimination items were observed in conjunction with all targets. Their function appears to be simply one of raising the general level of the obtained information curve. In the case of the one-parameter item pool, items with outlier values of item difficulty were also selected. These appear to be assuming the role of fillers under this model. It was not possible to trace the execution of the BLPX87 computer program to ascertain at what point in the selection process the filler items were employed.

A limited comparison of the simplex and branch-and-bound algorithms was conducted. Theunissen (personal communication, July 1987) kindly provided the authors with a 300-item pool based on the Rasch model. It was reanalyzed using four of his uniform targets. In three of the cases, the same number of items was selected by the simplex algorithm and the branch-and-bound integer solution. In the fourth case the branch-and-bound algorithm selected three fewer items than did the simplex. This reduction was the same as the number of fractionally weighted items yielded by the simplex algorithm. Because the first stage of the branch-and-bound solution involves the execution of a simplex step, the integer portion of the branch-and-bound algorithm does not appear to contribute greatly to the solution. In addition, the branch-and-bound procedure requires considerably longer computer runs.

The present results suggest that the simplex algorithm of linear programming tends to attempt to cope with the ''worst'' feature of the target information curve. In the case of a uniform target, this is the extremes of the target curve. In the case of a peaked target, this is the maximum of the target. The simplex algorithm selects items from the item pool until this worst feature is matched or exceeded, and it does not attend particularly well to the rest of the target curve. It would appear that the optimal target for this approach would have a wide range with a gently rounded form.

Due to this worst-feature characteristic, it is not necessary in some cases to specify a large number of points along the θ scale. Doing so increases the amount of computing time, and a similar information function would be obtained using only the θ levels corresponding to the worst feature. Two supplemental analyses were performed using only the end points $(-3,3)$ with a target amount of information of 16 (set 5 and set 14). The numbers of items selected were 192 and 151 with total computer run times of 8 and 15 minutes, respectively. Although the same numbers of items were selected as in the 11-point analyses, there were some trivial differences in the actual items selected. Thus, employing a smaller number of points than used in the present study appears to reduce the amount of computer time required to obtain a similar solution.

On balance, selecting items from an item pool by linear programming appears to be a useful addition to a test constructor's repertoire of tools. It could be used as an initial automated item selection procedure to be followed by the usual test refinement procedures.

References

Baker, F. B. (1978). *GENIRV: A computer program for generating item responses*. Unpublished manuscript, University of Wisconsin.

Boomsma, Y. (1986). *Item selection by mathematical* *programming* (Specialistisch Bulletin NR 47). Arnhem, The Netherlands: CITO (National Institute of Educational Measurement).

Dantzig, G. B. (1963). *Linear programming and extensions*. Princeton NJ: Princeton University Press.

Eastern Software Products, Inc. (1985). *BLP88 user's guide: Linear programming with bounded variables for the IBM PC* (p. 53). Alexandria VA: Author.

Feiring, B. R. (1986). *Linear programming: An introduction*. Beverly Hills CA: Sage.

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

Theunissen, T. J. J. M. (1985). Binary programming and test design. *Psychometrika, 50,* 411–420.

Theunissen, T. J. J. M. (1986). Some applications of optimization algorithms in test design and adaptive testing. *Applied Psychological Measurement, 10,* 381–389.

Author's Address

Send requests for reprints or further information to Frank B. Baker, Department of Educational Psychology, Educational Sciences Building, University of Wisconsin, Madison WI 53706, U.S.A.