

Running head: ITEM POOL CONSTRUCTION USING MIP/MIQP

Implementing the Mixed Integer Quadratic Programming
for Constructing Item Pools for Computerized Adaptive Testing Programs

Kyung T. Han

Lawrence M. Rudner

Graduate Management Admission Council®

Correspondence may be sent to:

Kyung T. Han

Graduate Management Admission Council

1600 Tysons Blvd. Suite#1400, McLean, VA 22102

khan@gmac.com

(Phone) 1-703-245-4363

(Fax) 1-703-749-0169

The views and opinions expressed in this article are those of the authors and do not necessarily reflect those of the Graduate Management Admission Council®.

Abstract

This study used mixed integer programming (MIP) and mixed integer quadratic programming (MIQP) to construct multiple highly equivalent item pools simultaneously. Three different MIP/MIQP models were implemented and evaluated using real CAT item pool data with 23 different content areas and a goal of equal information functions across pools and within each content area. The study addresses two important practical questions. First, how many evaluation points should the objective functions of the MIP/MIQP models use when the target is not $N(0,1)$? Second, how to structure the solver when an item bank is gigantic? The study found that all three MIP/MIQP models effectively constructed highly parallel item pools and content bins when five evaluation points were used..

Implementing the Mixed Integer Quadratic Programming for Constructing Item Pools for Computerized Adaptive Testing Programs

Introduction

For long-term quality control of computerized adaptive testing (CAT) programs, it is crucial to construct and maintain quality item pools that are consistent over time in terms of their psychometric properties and their match to the ability distributions of the test takers. Regardless of the adaptive algorithm, pool consistency is a necessary condition for consistency in the conditional standard error for the test takers.

Construction of multiple parallel item pools is often challenging, however, because of the number of factors to be considered (e.g., bank information, content balancing, exposure rate, response time, etc.) and the limited number of items available in the item bank. In applied settings, the goal is often to have consistency in information functions across pools and content is balanced in terms of the number of items within each content area in each pool. Item pool construction is usually done manually using sampling techniques. However, constructing multiple parallel item pools meeting all the pool specifications by hand is often very labor intensive especially when the number of item pools to be constructed and/or the number of items for each pool is large and when exposure control is one of the constraints.

This study investigated the feasibility of using mixed integer programming (MIP) and mixed integer quadratic programming (MIQP) to construct highly equivalent multiple item pools that meet content, exposure, and psychometric constraints, including the goal of having each content area have the equivalent information functions across pools. Three models and four different evaluation points in the objective functions were explored. Critical outcome variables were consistency of information functions and computational time.

Mixed Integer Programming Models

Linear Programming (LP) models have been widely used in various industries (from delivery services to financial institutions) to optimize resources while maximizing the outcomes. In the educational measurement field, applications of automated test assembly (ATA) started to adopt LP models for optimal test design since 1980's. For ATA, the most common setting of LP is to introduce as many 0-1 binary variables as the number of available items in the item bank, and, then, to let the solver find the best combination of the binary variables that results in the maximum (or minimum depending on a problem) objective value to find an optimal test design (Theunissen, 1985). When there are multiple test forms to be constructed at the same time, the LP often takes the form of Network-Flow programming, in which an array of integer variables

$(i \times j)$ are to be determined to optimize the flows between i supply nodes and j demand nodes (van der Linden, 1998). Since the decision variables of this kind of LP models are integer, it is generally called mixed integer programming (MIP) model across industries, and so will be referred in this paper.

Item pool construction is technically not very different from test assembly. Just as MIP models for ATA can be used to assemble the best sets of tests out of a given item pool, the item pool can also be systematically assembled using MIP models to produce optimal sets of item pools. Ariel, van der Linden, and Veldkamp (2004) used a MIP model to optimally divide an item bank into multiple operational pools with similar content distributions. Van der Linden, Ariel and Veldcamp (2006) discussed forming pools to meet the ability distributions of the targeted test takers while meeting content constraints. However, meeting content constraints does not necessarily mean that the test questions within a content area will have similar information functions across pools.

In CAT administration, which can be viewed as a special form of ATA, the one common item selection criterion is the maximized Fisher information (MFI). This can be modeled mathematically as a MIP problem with an objective and a set of constraints. One basic model for optimally selecting n items from a bank of J questions for an individual test is:

$$\text{Maximize } \sum_{j=1}^J I(j|\hat{\theta})x_j \quad (1)$$

$$\text{subject to } \sum_{j=1}^J x_j = n, \text{ (test length)} \quad (2)$$

where $I(j|\hat{\theta})$ is the expected item information for item j at the interim theta estimate, and x_j is a 0-1 variable representing the exclusion or inclusion of item j . In practice, of course, there are additional constraints.

While appropriate for developing individual tests, an objective function with maximization is often not the most efficient approach for controlling the psychometric properties of item pools. For pool construction it is more common to have targeted (or planned) psychometric properties and the objective function is usually of a minimization problem seeking to minimize the difference between a target and an item pool constructed from the data. Another difference between MIP models for CAT ATA and item pool construction is that the target for pool information function (PIF) is established mainly based on an expected distribution of examinees' proficiency, which is not a point estimate on the theta scale. Thus, the MIP model for CAT ATA above can be modified for item pool construction by evaluating the information function conditioned on θ :

$$\text{Minimize } \int_{\theta=-\infty}^{\infty} [\sum_{j=1}^J I(j|\theta)x_j - \tau_{\theta}]d\theta \quad (3)$$

subject to

$$\sum_{j=1}^J x_j = n, \text{ (item pool size)} \quad (4)$$

$$\int_{\theta=-\infty}^{\infty} \sum_{j=1}^J (I(\theta)x_j - \tau_{\theta})d\theta \geq 0, \text{ (range of variables)} \quad (5)$$

where τ_{θ} is a target PIF at θ .

When multiple item pools are constructed at once, the MIP model above can be modified as following by summing across pools in the objective function and adding an additional constraint:

$$\text{Minimize } \sum_{p=1}^P \int_{\theta=-\infty}^{\infty} [\sum_{j=1}^J I(j|\theta)x_{pj} - \tau_{\theta}]d\theta \quad (6)$$

subject to

$$\sum_{j=1}^J x_j = n, \text{ (item pool size)} \quad (7)$$

$$\int_{\theta=-\infty}^{\infty} \sum_{j=1}^J (I(\theta)x_j - \tau_{\theta})d\theta \geq 0, \text{ (range of variables)} \quad (8)$$

$$\sum_{p=1}^P x_{pj} \leq m \text{ for all } j, \text{ (item usage across pools)} \quad (9)$$

where $p=1,2,3, \dots, P$ with P being the number of item pools to be constructed at the same time and m is the maximum usage for each item across pools.

Since the minimization of the objective function is limited by the constraint, this model will be referred to as Single Bound Model (SBM) in this paper. SBM is mathematically simple and very straightforward, and there are many LP/MIP solvers that can handle such a model. The downside of SBM is that the chance of encountering infeasibility issues during the solving process is often fairly high if τ is not set to very low value especially when the values of J and m are small and/or the values of T and P are large.

When a chance of running into infeasibility issues is expected to be moderate to severe with SBM, a different approach to modeling an MIP is often suggested. Infeasibility issues usually occur because of unrealistic settings for constraints. In practice, however, it is often impossible for practitioners to see if the constraints are unrealistic given the item bank data before they actually attempt to solve the MIP model. Thus, it can sometimes be very useful to

have an objective function that optimizes the constraints that restrict the difference between the target and actual PIFs. This approach is called the minimax approach (van der Linden, 2005). In the minimax approach, the Equations (6) and (8) of the SBM above are replaced with

$$\text{Minimize } \delta \quad (10)$$

subject to

$$\int_{\theta=-\infty}^{\infty} \sum_{j=1}^J I_j(j|\theta)x_{pj} d\theta \leq (\tau_\theta + \delta), \text{ for each } p \quad (11)$$

$$\int_{\theta=-\infty}^{\infty} \sum_{j=1}^J I_j(j|\theta)x_{pj} d\theta \geq (\tau_\theta - \delta), \text{ for each } p \quad (12)$$

$$\delta \geq 0. \quad (13)$$

In contrast to the SBM, the MIP model applying the minimax approach allows having actual PIF lower than the target PIF, and this change significantly relieves possible infeasibility issues during the solving process. Unlike SBM, the differences between the target and actual PIFs are controlled by the bands around the target, width of which is 2δ . Thus, this model will be referred to as the Minimized Band Model (MBM) in this paper.

There are several issues with MBM, however. First, finding the minimized δ (the best solution) could take much more time than solving the SBM because the MBM requires the solver to explore a much larger problem space with more constraints and also because of the nature of MBM that keeps the solver from effectively reducing the problem space using mathematical strategies. Another issue with the MBM is the fact that the band width across θ is decided by a single δ . When τ is carefully determined throughout θ and the item bank is large and good enough to support it, it should not be a problem in many cases. However, if τ is unrealistically specified at a certain point on θ so that the difference between the actual PIF and τ is unusually large at that θ level, it will result in δ which is unnecessarily too large for the other θ levels. Therefore, with MBM, finding the minimized δ does not always guarantee the minimized difference between the actual PIF and τ throughout θ .

To overcome the problems associated with SBM and MBM, a new objective function is proposed in this study. In this approach, Equation (6) of the SBM is replaced with a quadratic term as

$$\text{Minimize } \sum_{p=1}^P \int_{\theta=-\infty}^{\infty} [\sum_{j=1}^J I_j(j|\theta)x_{pj} - \tau_\theta]^2 d\theta \quad (14)$$

subject to

$$\sum_{j=1}^J x_j = n, \text{ (item pool size)} \quad (15)$$

$$\sum_{p=1}^P x_{pj} \leq m \text{ for all } j \text{ (item usage across pools).} \quad (16)$$

It should be noted that Equation (8) has been dropped from SBM because Equation (14) is already always above zero because the objective function is no more linear but quadratic. Switching Equations (6) and (8) with Equation (14) may seem a small change, but this modification fundamentally changes the optimization setup in terms of conceptual and technical definitions. Comparing to SBM, the elimination of the variable range constraint (Equation (8)) helps a chance of encountering infeasibility issues reduced by far. Also, unlike MBM, the difference between the actual PIF and τ is always minimized throughout θ . Thus, this model will be referred to as the Minimized Squared Difference Model (MSDM) in this paper. Although MSDM has several advantages over SBM and MBM, it has rarely used in the field because most of the LP solvers could not handle such a quadratic programming models. Only a very few among the most advanced solvers recently developed can handle mixed integer quadratic programming (MIQP) problems under very limited conditions.

The conceptual illustrations of SBM, MBM, and MSDM are shown in Figure 1. The reader should note the differences in the possible shapes of the constructed pool information functions.

Purpose of Study

The primary goal of this study is to compare the performance of the three different MIP/MIQP models (SBM, MBM, and MSDM) in constructing multiple parallel item pools. To implement those MIP/MIQP models, two important questions need to be answered first.

Unlike the CAT ATA, where the objective value (Equation (1)) is evaluated at one point (for example, at $\hat{\theta}$), the objective functions (Equations (6), (10), and (14)) for item pool constructions are based on a continuous scale of θ with integrals. To reduce the intensity of mathematical computation and to make the objective function recognizable for the solver, it is necessary to change the objective functions to ones with summations of discrete objective values. Thus, for example, the objective function for SBM (Equation (6)) is changed to

$$\text{Minimize } \sum_{p=1}^P \sum_{t=1}^T \left[\sum_{j=1}^J I(j|\theta_t) x_{pj} - \tau_t \right] \quad (17)$$

where $t=1,2,3, \dots, T$ with T being the number of evaluation points (EP) on the θ scale, θ_t is the θ value at EP t , and τ_t is a target PIF at θ_t . The same applies to MBM and MSDM. An important question, now, is how many EPs should be at what locations on the θ scale. Too few EPs may not enough to tightly control the actual PIF while too many EPs may unnecessarily dramatically increase the processing time of a solver. Van der Linden (2005, p 106) suggests that with a $N(0,1)$ expected theta distribution, three or four EPs specified at $(-1.0, 0.0, +1.0)$ or $(-1.5, 0.5, 0.5, 1.5)$

will yield excellent results. In practice, however, thetas do not remain $N(0,1)$ and target difficulty is not $N(0,1)$. This study aims to determine the optimal number and locations of EPs for data that is not $N(0,1)$.

Secondly, managing the size of MIP/MIQP problems to keep it under the usable computer resource limit is also a critical part of the solving process. Unlike typical LP problems, MIP problems with 0-1 variables (for example, x in Equations (17)) heavily rely on the branch-and-bound (BnB) algorithm (Land & Doig, 1960) to implement the iterative tree search. The size of a problem tree is defined by the number of possible combinations of the 0-1 variables. Thus, for example, if an ATA problem was to assemble a test form consisting of 30 items selected from an item pool with 500 items, the size of the problem tree would be

$$\binom{500}{30} = \frac{500!}{30!(500-30)!} \cong 1.445e^{48}. \quad (18)$$

Although $1.445E+48$ is a huge problem size, many solvers can effectively reduce the problem size by eliminating infeasible solutions (according to the constraints) and skipping (or cutting) not promising solutions using various mathematical strategies. However, when it comes down to the item pool construction, the size of a problem tree becomes just unrealistic. For example, if twelve parallel item pools (500 items for each pool) were to be constructed from an item bank with 10,000 items, the tree size would be

$$\binom{10000}{500 \times 12} = \frac{(10000)!}{6000!(10000-6000)!} \cong 5.794e^{2920}. \quad (19)$$

Many solvers cannot even setup an MIP problem with such an enormous tree size. A few advanced, high-performance solvers can theoretically manage the computer resource (memory and storage) to handle such a large MIP/MIQP problem, but it is still unknown if even the most advanced solver can finish the solving process in realistic time range without technical issues. Thus, in this study, the item pool construction is done in two different ways. In the first way, a whole item bank is modeled using MIP/MIQP. In the second way, the item bank is divided into subgroups, and then the MIP/MIQP is performed. By comparing the two cases, the performance of the solver when the problem tree is astronomically large is evaluated.

Once those two research questions are answered, the three MIP/MIQP models are compared and evaluated. A comprehensive discussion on the findings of the study will draw important guidelines for item pool construction using MIP/MIQP techniques.

Method

Pool Specification

For the study, 12,000 quantitative items were randomly selected from GMAT[®] exam item bank. This study pictured a scenario¹ in which twelve item pools had to be constructed to meet the following requirements to

- (a) Each item pool consists of items from 23 mutually exclusive content areas,
- (b) Number of items for each content area is equal to a pre-specified value, n_k , (k being an index for each content area),
- (c) An item cannot be included in more than two of the twelve pools, and
- (d) An item cannot be included in more than one of four consecutive pools.

The number of available items for each content area (N_k) in the item bank ranged between 233 and 938, and the pre-specified number of items for each content area (n_k) per item pool ranged between 27 and 40.

Models

The four aforementioned pool requirements were added as constraints into SBM, MBM, and MSDM. To add the content area component, C_{jk} , a matrix of 0-1 constants that indicates the content area of each item was added to the models, in which

$$C_{jk} = \begin{cases} 1 & \text{if item } j \text{ belongs to content area } k \\ 0 & \text{otherwise.} \end{cases} \quad (20)$$

Thus, for example, for SBM, the final model could be expressed as,

$$\text{Minimize } \sum_{k=1}^K \sum_{p=1}^P \sum_{t=1}^T \left[\sum_{j=1}^J I(j|\theta_t) x_{pj} C_{jk} - \tau_{kt} \right] \quad (21)$$

subject to

$$\sum_{j=1}^J x_{pj} = n, \text{ for each } p \text{ (pool size),} \quad (22)$$

$$\sum_{j=1}^J x_{pj} C_{jk} = n_k, \text{ for each } k \text{ and } p \text{ (content areas),} \quad (23)$$

$$\sum_{j=1}^J I(j|\theta_t) x_{pj} C_{jk} - \tau_{kt} \geq 0, \text{ for each } k, p, \text{ and } t \text{ (information targets),} \quad (24)$$

$$\sum_{p=1}^P x_{pj} \leq 2, \text{ for each } j \text{ (maximum item use),} \quad (25)$$

¹ This is a hypothetical scenario, and the pool specifications are different from the operational GMAT[®] exam.

$$\sum_p^{(p+3)\leq P} x_{pj} \leq 1, \text{ for each } j \text{ (consecutive use constraint).} \quad (26)$$

To find the optimal number and location of the evaluation points (EPs) on the θ scale, four different combinations of number and location of EPs were attempted: 1 EP ($\theta=0$), 3 EPs ($\theta=-1, 0$, and 1), 3 EPs ($\theta=-2, 0$, and 2), and 5 EPs ($\theta=-2, -1, 0, 1$, and 2). Thus, for our analysis we have $P=12$ pools to be created, $J=12,000$ items in the bank, $K=23$ content areas, and we will evaluate $T=1, 3$ and 5 EPs.

As a baseline, pools were constructed manually using sampling techniques. In this manual construction, the mean and standard deviation (SD) of a - and b -parameters of each pool were matched to the other item pools and the maximum information was targeted at about $\theta=0.85$. The manually constructed item pools were compared with the item pools constructed using the MIP/MIQP optimization.

Implementation

As MIP/MIQP models, SBM, MBM, and MSDM were built using the optimization modeling software, *AIMMS 3.10FR2* 64-bit edition (see <http://www.aimms.com>). Within AIMMS, one of the most advanced solvers, *CPLEX 12.1* (see <http://www.cplex.com>), which could handle both MIP and MIQP, was used. To make the EP conditions comparable to each other, each of the objective functions was divided by the number of EPs. The absolute optimality tolerance was set to 0.547, 0.045, and 0.024 for SBM, MBM, and MSDM, respectively, to result in the equivalent level of item pool consistency in terms of the mean difference between PIF and τ across the MIP/MIQP models. To expedite the iterative process of the BnB algorithm, the cutoff criteria for SBM, MBM, and MSDM were set to about twice as large as the corresponding absolute optimality tolerances based on the results from multiple preliminary runs. The best-estimate strategy was used in the node selection to start from a node after all integer infeasibilities were removed. The strong branching, in which the variable selections were based on partially solving a number of subproblems to see which branch was the most promising, was also used because this approach was very effective on large MIP/MIQP problems. To utilize the all available computer resources, the solver was set to run in the parallel thread mode using all of available CPU cores.

As mentioned earlier, two different ways of implementation were conducted. First, the item bank ($N=12,000$) was divided into the 23 subgroups (i.e., item bins) by content areas, each of which was mutually exclusive to the others. Then, the solving process was performed for each of the 23 subgroups (i.e., 23 separate runs of the solver). In the second way, the solver was set to

determine all 144,000 binary variables (the $p \times j$ matrix of x 's, where $P=12$ and $J=12,000$) to construct 12 item pools with all 23 content area at once.

The computer system used for the solving process was a virtual machine built on Microsoft® Windows Server 2003® 64-bit Edition with a dedicated Intel® Xeon® CPU with four cores running at 2.80 GHz. The computer had 16 GB of physical memory on it. It should be noted, as a practical tip, that using a 64-bit operating system is almost necessary to avoid system crashes associated with the memory management when the size of a MIP/MIQP problem is very large and the parallel thread running is required.

Result

The pool information functions of the 12 manually constructed item pools are shown in Figure 2. Although the PIFs across the 12 item pools slightly differed in where they peaked, the overall shapes of the PIFs were quite similar. When we looked at the information functions at the item bin level (i.e., subgroups by content area), however, the information functions across the pools within each bin are very inconsistent. For example, in Figure 3, the bin information functions (BIFs) for Bins #6 and #9 differed by about 200% between some pools where they peaked, and for Bins #16 and #17, the overall shapes of the BIFs are very different from one pool to another. In sum, the item pool construction using the sampling technique matching the mean and SD of item parameter values across item pools might result in the modest level of consistency in PIF at the pool level, but at the lower level (i.e., item bin or item content area), the quality of the item bins could be very inconsistent across item pools (or over time). In terms of the quality control of item pools over item, the item pools constructed manually in this result may be still seen as acceptable to test developers in the real world depending on the test developers' intention. However, it is undeniable that the inconsistency in the psychometric properties among the item pools when the pools were manually constructed was far from what was ideal, and there was a great deal of room for improvement.

The item pool constructions using the MIP/MIQP optimization based SBM, MBM, and MSDM were conducted. The information functions of the constructed item pools at the item pool level are shown in Figure 4. When the objective functions were controlled at one evaluation point (1EP; $\theta=0$), the PIFs of the item pools were very similar to what was seen in the manual construction cases in Figure 3. When the objective functions were controlled at three evaluation points with 1 SD interval (3EP1SD; $\theta=-1, 0, \text{ and } 1$), the differences among the PIFs across the 12 pools were still as large as the 1EP condition. Another tendency of PIFs in the 3EP1SD condition was that most of the pools showed PIFs exceeding the target where $\theta > 1$. When the objective functions were controlled at three evaluation points with 2 SD interval (3EP2SD; $\theta=-2, 0, \text{ and } 2$), the MBM case resulted in the most consistent PIFs across the 12 pools. The PIFs tended to be closer to the target in the 3EP2SD condition than in the 3EP1SD condition, but, with SBM and

MSDM, many of the pools showed PIFs lower than the target where the PIFs peaked around $0 < \theta < 2$. When the objective functions were controlled at five evaluation points (5EP; $\theta = -2, -1, 0, 1, \text{ and } 2$), PIFs of the 12 pools were on the top of each other as well as right on the target throughout θ with all three MIP/MIQP models. This was a dramatic improvement over the manual pool construction (Figure 1) in terms of the quality control of item pool construction. Based on the results of the 1EP, 3EP1SD, 3EP2SD, and 5EP conditions, it would be reasonable to conclude that (1) the objective functions should be evaluated where the PIFs are expected to peak, (2) the interval of EPs should be 1 SD or less, and (3) there should be at least five EPs to effectively control the objective functions throughout the θ scale.

Although we found the information functions of the 12 pools controlled effectively at the pool level with the 5 EPs, examining if the 12 item pools were consistent at the critical item bin level (for each item content area) was still important because, as we saw in Figure 3, consistency in pool information does not necessarily translate to consistency at the content or bin level, even if the items are drawn from these mutually exclusive bins. The BIFs of the item pools constructed by SBM, MBM, and MSDM are shown in Figures 5, 6, and 7, respectively. With 3 EPs or less, BIFs often differ meaningfully from the target as well as differ substantially among the 12 pools. With 5 EPs, BIFs were right on the target across the 12 pools within each content area. There was a few cases where the 12 pools differed from each other within each bin when $\theta > 2$, but the differences observed were negligible. Overall observations on the BIFs led us to the conclusion that evaluating the objective functions on the five points (5EP) was effective to control the information at the item bin level as well as at the item pool level.

A choice of the MIP/MIQP models did not seem to make meaningful differences in terms of quality of the constructed item pools in this study in which the carefully chosen targets (τ) were well supported by the large item bank. However, when it came to the time took for the solving processes, there were substantial differences among SBM, MBM, and MSDM as shown in Table 1. When the objective functions were evaluated at one EP (1EP), the solver performed the fastest with the MBM objective. The average processing time across 23 runs (for 23 item bins, with 12 pools for each bin) was 9 seconds. In 3EP1SD and 3EP2SD conditions, the processing time increased comparing to the 1EP condition, but the MBM still resulted in the shortest processing time. Under the 5EP condition, which resulted in the near optimal pools and bins, the processing time for MBM jumped to an average of 2,868 seconds (47.8 minutes), and three out of the 23 runs could not finish within the time limit (10,000 seconds or 166.6 minutes) although the objective function values of those three cases were very close to the absolute optimality tolerance. For the 5EP condition, SBM was the model that took the shortest time to solve with an average of 921 seconds or 16 minutes.

MSDM took the longest time to solve when there were 1 or 3 EPs, but what was interesting about MSDM was that the processing time did not dramatically increase as the number of EPs increase comparing to the other models. It should be noted that the tendency in the processing time influenced by the number of EPs and a choice of MIP/MIQP models was not

directly generalizable, but the overall result of the processing time was informative enough to help us make a better choice on the MIP/MIQP models according to the number of EPs to consider.

A simultaneous solving, where all of the 23 bins (each of which had 12 pools) were to be constructed altogether, was also attempted to see if the solver could still effectively solve the MIP/MIQP models when the size of problem tree was extremely large. With SBM and MSDM, the solver falsely concluded that there was no feasible solution after the pre-solving process. When MBM was used, the solver ran until the solving process was forcedly terminated due to out of computer resource after 1,219,846 seconds (more than 2 weeks) of running (1,964,920 nodes explored). The objective function value of the best solution found so far was 13.820, which was much larger than the sum of the objective values from the 23 separate runs (0.904) in the 5EP condition.

Discussion and Conclusion

The study aimed to answer three important questions for item pool construction using the MIP/MIQP technique. The first question was how many EPs would be necessary to have to effectively control the information functions both at the item pool level and at the item bin (i.e., content area) level. In the studied condition, where the information function between -2 and 2 of the θ scale was the most interested, at least five EPs were required to control the information function effectively. The EPs do not have to be equidistance, but findings of the study suggested the intervals between EPs to be 1 SD or less. Thus, if the information functions need to be controlled for a wider range on the θ scale, more EPs would be required.

The second research question in this study was to see whether the latest solver could handle a large item pool construction at once. Under the studied condition, where 12 item pools were to be constructed from an item bank of 12,000 items, the solver could not even get to the actual solving stage for some cases. Even if the solving process was successfully initiated, the level of optimality of the solutions found within the realistic setting of processing time was hardly satisfactory. On the other hand, when the item bank could be stratified into smaller subgroups (each subgroup had less than 1,000 available items to choose from), the solver was able to find near-perfect solutions for constructing 12 item pools within an hour except for a few cases with MBM. Thus, if stratifying the item bank and item pools into mutually exclusive, smaller subgroups is possible, it is advised to do so. If stratifying the item bank is not allowed and the size of item bank is gigantic, some other approaches like van der Linden's big-shadow-test approach (2004) or sequential solving (constructing one item pool with each run) should be used to reduce the size of the problem tree.

The last research question, which was the main question of this study, was how SBM, MBM, and MSDM performed the item pool construction comparing to each other. Under the studied conditions, all of the three MIP/MIQP models turned out to be effective in item pool

construction when the number of EPs were 3 or less. However, when to insure content is consistent across pools, 5 EPs were needed. Under the 5EP condition, MBM took a longer time compared with SBM and MSDM. With 23 content areas, SBM took a total of approximately 6 hours to derive a solution and MSDM took more than 18 hours. Relative to the person-hours required for manual construction, these times are extremely fast and the time advantage of SBM might not be a major consideration.

When a MIP/MIQP model is to be selected for the item pool construction, there are practical advantages of using MSDM (MIQP) over the other two models. First, MSDM usually has much less constraints than the comparable SBM, and as a result, has much less chance of encountering infeasibility issues. Secondly, MSDM looks for the solution that is the most optimal at each of the EPs, whereas the best solution by MBM is not necessarily the most optimal at each EP. Therefore, if the processing time is not the main factor to consider, MSDM should be the choice over SBM and MBM as long as the solver can handle MIQP problems.

References

- Ariel, A., Veldkamp, B. P. & van der Linden, W. J. (2004). Constructing rotating item pools for constrained adaptive testing. *Journal of Educational Measurement*, 41, 345-359.
- Land, A. H., & Doig, A. G. (1960). An automatic method of solving discrete programming problems. *Econometrica*, 28(3), 497-520.
- Theunissen, T.J.J.M. (1985). Binary programming and test design. *Psychometrika*, 50, 411-420
- van der Linden, W. J. (1998). Optimal assembly of psychological and educational tests. *Applied Psychological Measurement*, 22, 195-211.
- van der Linden, W.J. (2005). *Linear models for optimal test design*. New York: Springer.
- van der Linden, W. J., Ariel, A., & Veldkamp, B. P. (2006). Assembling a CAT item pool as a set of linear test forms. *Journal of Educational and Behavioral Statistics*, 31. 81-100.

Table 1. Processing Time and Final Objective Value for Each MIP/MIQP Model

Bin	Model 1 (SBM)								Model 2 (MBM)								Model 3 (MSDM)							
	1EP		3EP1SD		3EP2SD		5EP		1EP		3EP1SD		3EP2SD		5EP		1EP		3EP1SD		3EP2SD		5EP	
	time	Obj.	Time	Obj.	time	Obj.	time	Obj.	time	Obj.	time	Obj.	time	Obj.	time	Obj.	time	Obj.	time	Obj.	time	Obj.	time	Obj.
1	50	0.389	1032	0.518	987	0.514	960	0.499	12	0.042	218	0.037	42	0.044	1333	0.044	1206	0.021	1131	0.023	919	0.022	1393	0.017
2	130	0.536	624	0.496	910	0.534	636	0.471	10	0.023	1355	0.043	164	0.043	10000	0.057*	853	0.014	954	0.016	588	0.013	3071	0.018
3	192	0.520	1553	0.518	834	0.415	813	0.533	35	0.042	243	0.044	330	0.042	1983	0.042	1891	0.023	1481	0.019	1096	0.019	2021	0.019
4	29	0.390	119	0.521	228	0.533	137	0.479	4	0.041	71	0.043	62	0.040	524	0.044	287	0.016	187	0.019	131	0.018	347	0.020
5	55	0.517	904	0.456	198	0.446	665	0.535	3	0.024	59	0.041	33	0.047	564	0.038	118	0.023	149	0.022	64	0.021	231	0.022
6	64	0.526	231	0.534	397	0.513	1061	0.534	9	0.043	197	0.043	107	0.042	1036	0.044	823	0.024	776	0.022	512	0.017	1066	0.022
7	13	0.420	263	0.519	1332	0.503	2393	0.517	8	0.040	414	0.040	93	0.039	9698	0.044	759	0.022	647	0.019	444	0.013	872	0.024
8	50	0.565	378	0.520	988	0.490	851	0.535	5	0.035	313	0.044	28	0.044	1351	0.043	457	0.023	326	0.023	202	0.020	644	0.023
9	62	0.465	736	0.437	796	0.460	655	0.520	6	0.058	443	0.039	106	0.044	825	0.043	608	0.010	665	0.023	542	0.009	883	0.018
10	56	0.516	261	0.368	362	0.509	1265	0.523	6	0.042	743	0.041	56	0.043	10000	0.054*	695	0.015	918	0.019	563	0.023	1197	0.022
11	13	0.531	1683	0.526	1252	0.501	902	0.534	9	0.040	528	0.038	45	0.043	1642	0.044	870	0.020	806	0.017	508	0.021	1309	0.023
12	53	0.457	454	0.474	94	0.514	620	0.523	3	0.032	371	0.041	88	0.035	1136	0.044	206	0.016	201	0.020	106	0.010	285	0.024
13	83	0.523	296	0.492	460	0.370	2491	0.531	9	0.035	701	0.043	128	0.035	3858	0.044	1019	0.021	1451	0.009	843	0.013	1519	0.021
14	24	0.533	768	0.518	1531	0.481	1950	0.519	14	0.041	41	0.043	66	0.042	1411	0.044	1593	0.020	831	0.023	902	0.015	2109	0.017
15	27	0.506	780	0.515	1598	0.529	1184	0.504	16	0.027	143	0.043	138	0.044	10000	0.048*	2236	0.013	1592	0.010	1059	0.023	2418	0.016
16	28	0.486	76	0.530	131	0.527	123	0.502	4	0.027	62	0.036	28	0.043	4012	0.040	178	0.016	195	0.021	95	0.021	354	0.022
17	43	0.457	183	0.529	634	0.478	494	0.522	4	0.035	99	0.039	77	0.042	862	0.043	260	0.014	323	0.018	152	0.024	562	0.024
18	52	0.492	173	0.466	258	0.523	645	0.528	4	0.036	102	0.044	90	0.042	545	0.041	376	0.018	343	0.015	204	0.005	681	0.021
19	11	0.422	198	0.521	224	0.528	413	0.514	7	0.044	94	0.044	21	0.041	1322	0.040	659	0.023	482	0.022	492	0.022	816	0.023
20	67	0.524	368	0.513	432	0.435	582	0.450	7	0.039	167	0.044	56	0.044	943	0.044	799	0.008	927	0.022	382	0.021	1234	0.019
21	66	0.520	250	0.487	471	0.533	1268	0.485	12	0.029	145	0.044	34	0.039	1773	0.044	1111	0.017	923	0.015	623	0.015	1101	0.023
22	70	0.278	179	0.457	210	0.474	838	0.516	7	0.041	129	0.043	33	0.039	319	0.044	538	0.024	641	0.020	441	0.019	739	0.023
23	70	0.415	146	0.492	362	0.487	235	0.461	6	0.042	105	0.044	11	0.044	817	0.046	641	0.010	520	0.018	342	0.012	582	0.023
Avg.	57	0.478	507	0.496	639	0.491	921	0.510	9	0.037	293	0.042	80	0.042	2868	0.044	791	0.018	716	0.019	487	0.017	1106	0.021

Note: Two stop criteria for the solving process were used; (a) CPU time (10,000 sec) or objective value (0.547, 0.045, and 0.024 for Models 1, 2, and 3, respectively). The objective values with “*” indicate that the solver was stopped due to the time limit before finding solutions that resulted in the objective values smaller than the stopping criteria.

Figure 1. Three Optimization Models (excluding Constraints for Item Exposure Control)

Model	Demonstration	Main Part of Model
<p>Model 1: Single Bounded Model (SBM) Type: MIP</p>		<p>Minimize</p> $\sum_{p=1}^P \int_{-\infty}^{\infty} [\sum_{j=1}^J I(j \theta)x_{pj} - \tau_{\theta}] d\theta$ <p>subject to</p> $\sum_{j=1}^J I(j \theta)x_j - \tau_{\theta} \geq 0, \text{ for each } p.$
<p>Model 2: Minimized Band Model (MBM) Type: MIP</p>		<p>Minimize δ</p> <p>subject to</p> $\sum_{j=1}^J I_j(j \theta)x_{pj} \leq \tau_{\theta} + \delta,$ <p>and</p> $\sum_{j=1}^J I_j(j \theta)x_{pj} \geq \tau_{\theta} - \delta,$ <p>for each p across $\theta[-\infty, \infty]$.</p>
<p>Model 3: Minimized Squared Difference Model (MSDM) Type: MIQP</p>		<p>Minimize</p> $\sum_{p=1}^P \int_{-\infty}^{\infty} [\sum_{j=1}^J I(j \theta)x_{pj} - \tau_{\theta}]^2 d\theta$

Figure 2. Pool Information Functions of Twelve Manually Constructed Item Pools

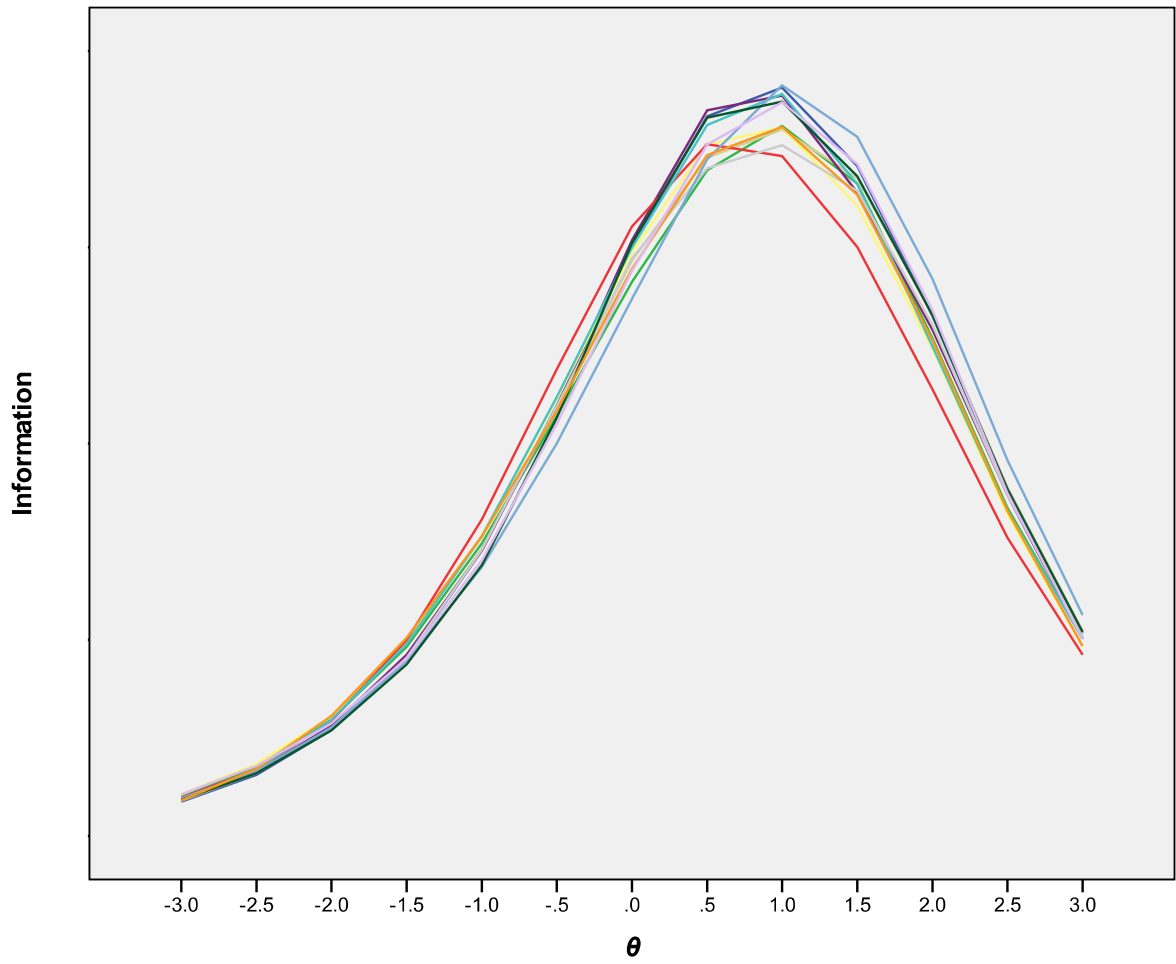


Figure 3. Bin Information Functions of Twelve Manually Constructed Item Pools

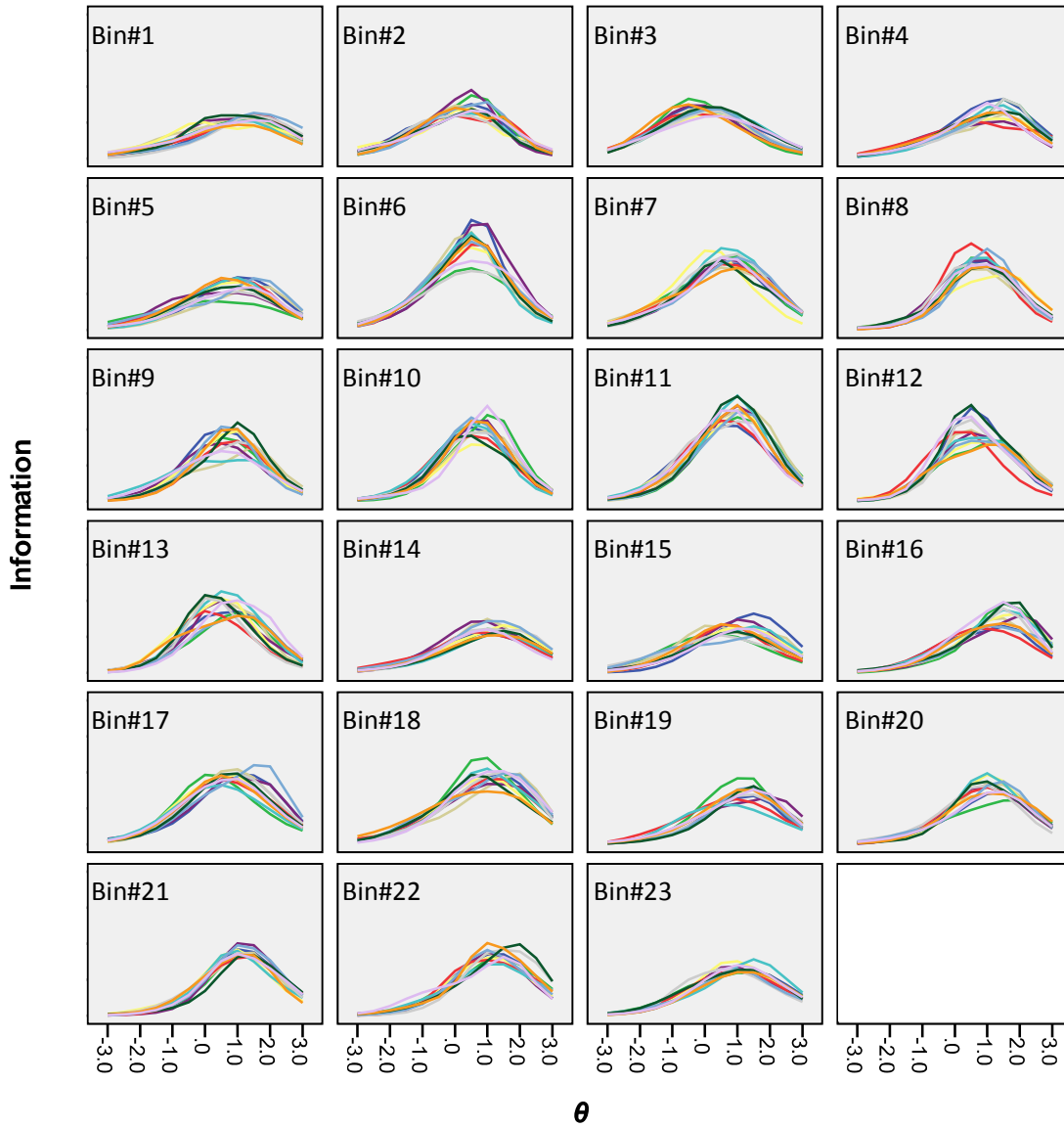


Figure 4. Information Functions of 12 Pools Constructed by Three Models at the Pool Level.

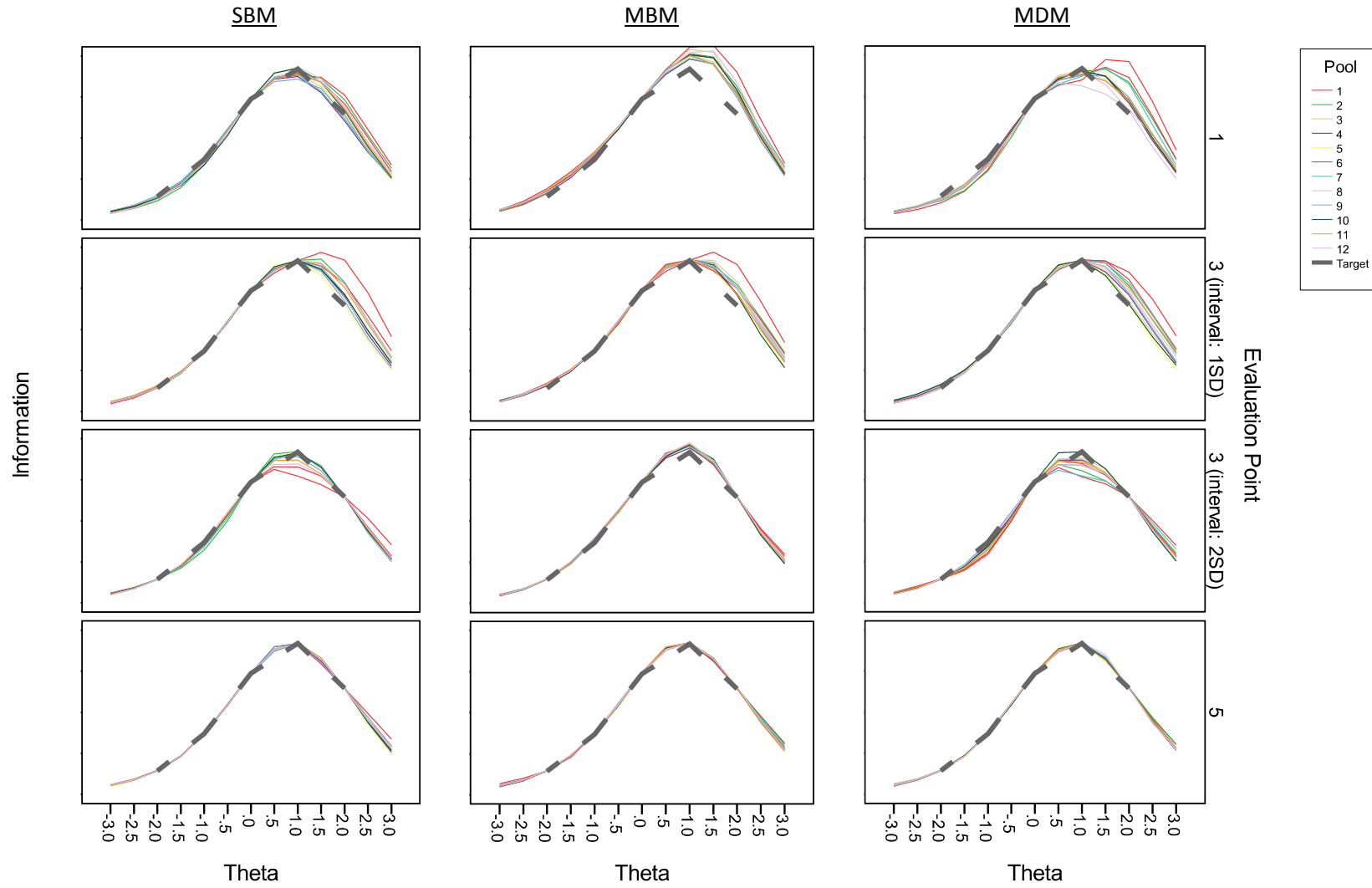


Figure 5. Bin Information Functions for 12 Pools Constructed with SBM (MIP).

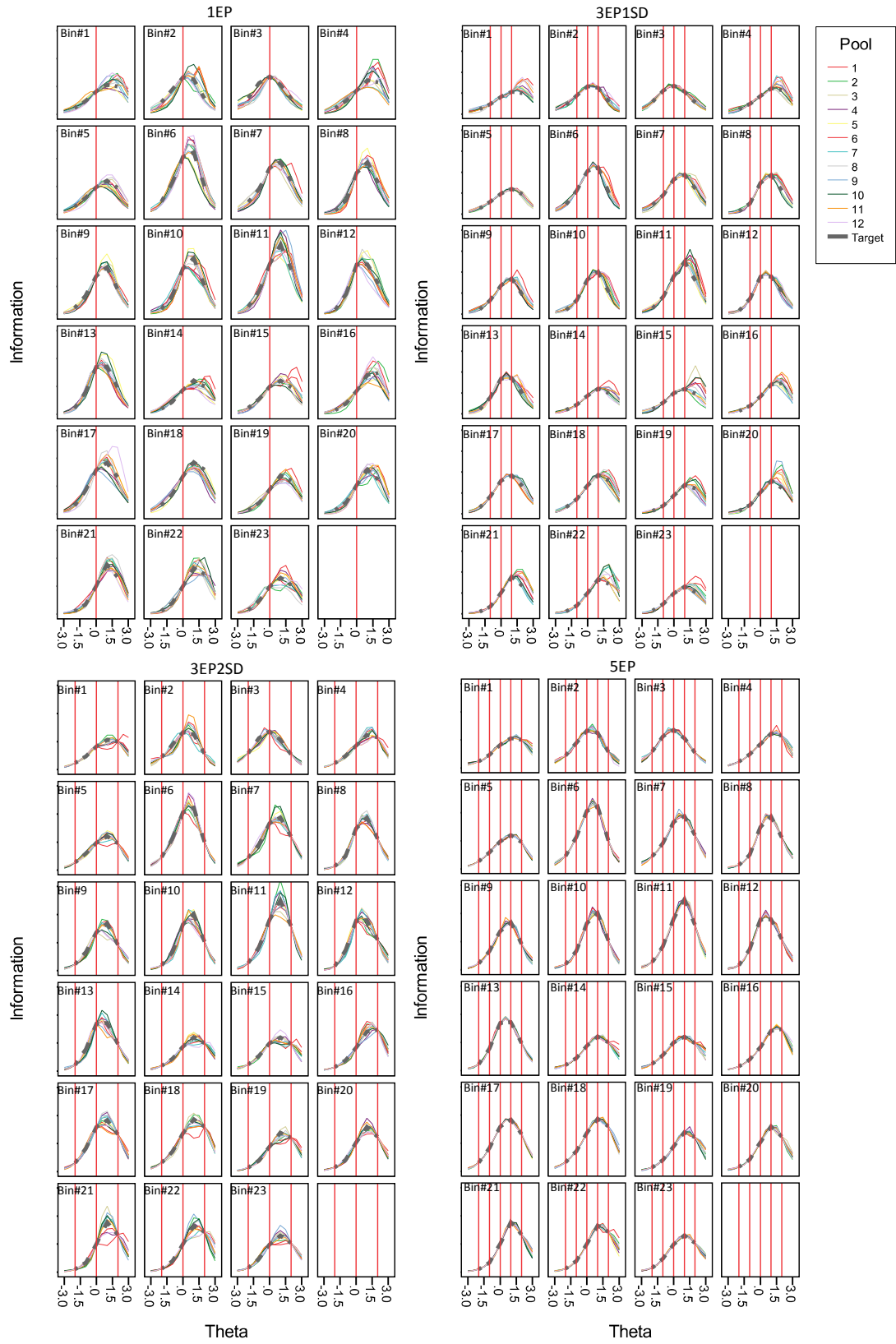


Figure 6. Bin Information Functions for 12 Pools Constructed with MBM (MIP).

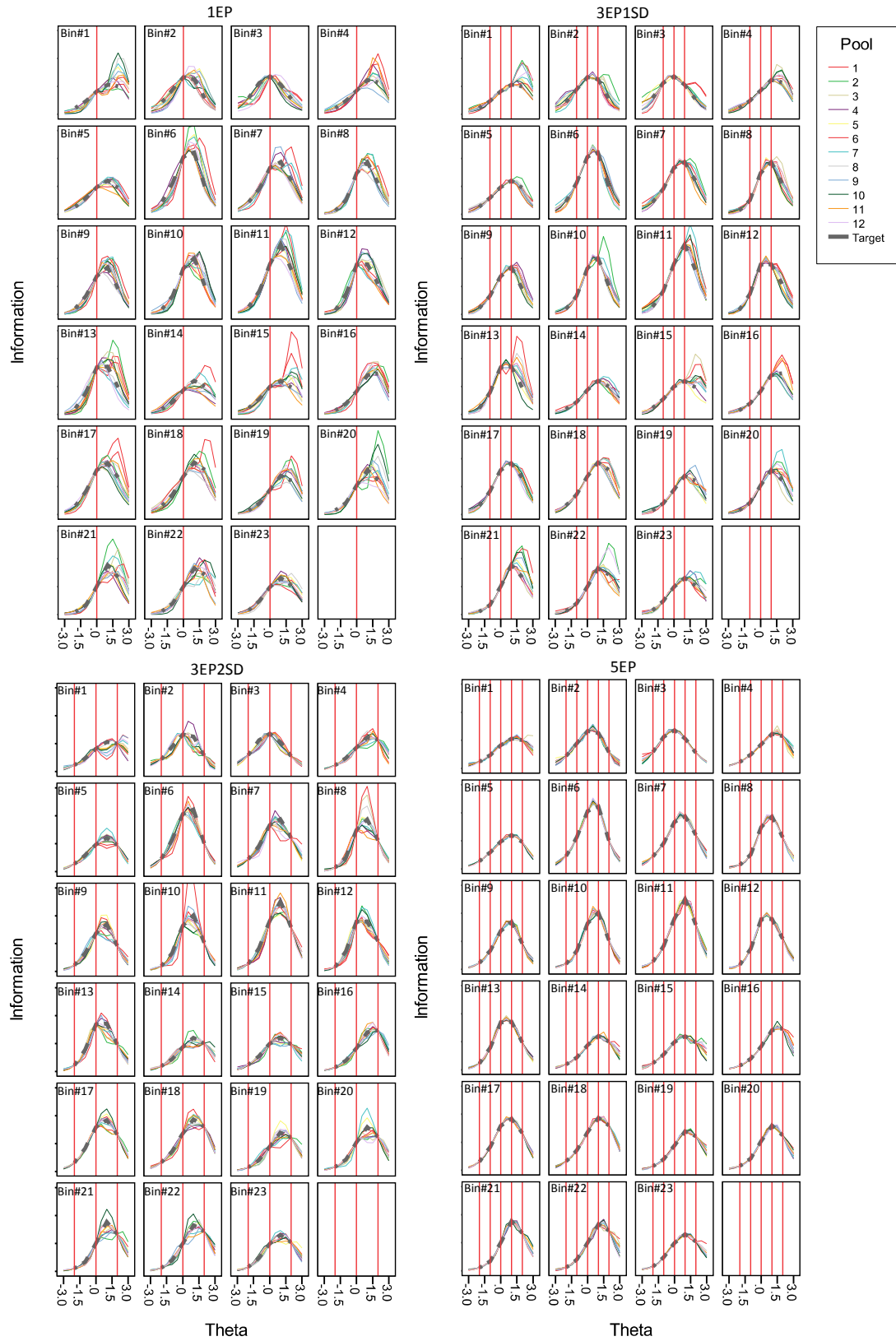


Figure 7. Bin Information Functions for 12 Pools Constructed with MSDM (MIQP).

