Journal of Computerized Adaptive Testing

Volume 6 Number 2

August 2018

Adaptive Item Selection Under Matroid Constraints

Daniel Bengs, Ulf Brefeld, and Ulf Kröhne

DOI 10.7333/1808-0602015

The *Journal of Computerized Adaptive Testing* is published by the International Association for Computerized Adaptive Testing
www.iacat.org/jcat

ISSN: 2165-6592

International Association for Computerized Adaptive Testing
IACAT
Advancing the Science and Practice of Human Assessment

Journal of Computerized Adaptive Testing

Volume 6, Number 2, August 2018

DOI 10.7333/1808-0602015

ISSN 2165-6592

Adaptive Item Selection Under Matroid Constraints

Daniel Bengs
German Institute of International Educational Research

Ulf Brefeld
Leuphana University

Ulf Kröhne
German Institute of International Educational Research

The shadow testing approach (STA; van der Linden & Reese, 1998) is considered the state of the art in constrained item selection for computerized adaptive tests. The present paper shows that certain types of constraints (e.g., bounds on categorical item attributes) induce a matroid on the item bank. This observation is used to devise item selection algorithms that are based on matroid optimization and lead to optimal tests, as the STA does. In particular, a single matroid constraint can be treated optimally by an efficient greedy algorithm that selects the most informative item preserving the integrity of the constraints. A simulation study shows that for applicable constraints, the optimal algorithms realize a decrease in standard error (SE) corresponding to a reduction in test length of up to 10% compared to the maximum priority index (Cheng & Chang, 2009) and up to 30% compared to Kingsbury and Zara's (1991) constrained computerized adaptive testing.

Keywords: computerized adaptive testing, content balancing, constrained item selection, enemy constraints, greedy algorithm, matroid intersection, matroids

Computerized adaptive testing (CAT; Weiss, 1982) enables improvements in measurement accuracy over fixed forms (Segall, 2005). However, while fixed forms make test composition obvious to the test designer, adaptive selection of items based exclusively on statistical considerations may lead to tests whose composition varies considerably among examinees. This gives rise to a number of issues. Among others, face validity may be called into question due to a lack of adequate coverage of content areas (van der Linden & Reese, 1998); and independence assumptions of the test model may be violated if similar, mutually cluing items are administered together, for instance, items derived from a common stem (Leuders et al., 2017). These problems are avoided by imposing adequate constraints on test composition and, hence, on item selection. The literature on constraint management is usually divided into approaches based on heuristics versus approaches involving mathematical programming. The mathematical programming approach formulates the test assembly problem as an optimization problem whose solution yields constraint compliant tests (van der Linden & Reese, 1998) that are optimal at the estimated trait level. Basically, this is achieved by optimizing test information over the space of all complete tests that are constraint compliant.

There are two drawbacks to this approach. First, the combinatorial complexity renders the problem demanding that the optimization problem posed by the shadow test approach (STA) take the form of a 0-1 integer program, which is known to be NP-hard (Karp, 1972), meaning that no polynomial time algorithm for its solution is known to exist. Thus, in the worst case, the time required to compute a shadow test is exponential to the size of the item bank. In practice, however, typical instances of the STA can be solved very quickly by sophisticated optimization algorithms, which usually employ the method of branch and bound (Land & Doig, 1960) or branch and cut (Padberg & Rinaldi, 1991) to narrow down the space in which to search for the optimal solution. These solvers are highly non-trivial to design and implement and, thus, need to be acquired in the form of libraries that are external to the CAT software. This also complicates the development and maintenance of the CAT program and renders its application costly in terms of labor and funds. Therefore, it seems sensible to ask if the computational and practical complexity of the STA is the price to be paid for optimal constrained testing or if simpler, yet optimal, solutions exist. Heuristic approaches (Cheng & Chang, 2009; Kingsbury & Zara, 1991) do have the benefit of conceptual and computational simplicity and can easily be implemented, but fall short of attaining maximal testing efficiency (He, Diao, & Hauser, 2014).

The purpose of this study was to show that certain types of constraints, notably lower and upper bounds on categorical item attributes, induce the structure of a matroid on the item bank. Matroids are algebraic structures that provide an abstraction to concepts such as linear independence in vector spaces and circuits in graphs, and can be considered central to the analysis of various problems in combinatorial optimization (Lawler, 1976). The present interest in matroids arises from the fact that matroids provide a formal description of optimization problems that can be solved optimally by a greedy algorithm (e.g., Edmonds, 1971). Basically, greedy algorithms build a solution to a complex problem by starting from the empty set and then proceeding to repeatedly add the element that maximally increases the objective function. A trivial example of a greedy algorithm can be found in unconstrained maximum information item selection, which assembles a maximally informative test by selecting an item that provides maximal increase in test information for the current trait estimate. By restricting item selection to the tests that are designated feasible by a matroid constraint, this simple strategy carries over to constrained test assembly. By virtue of the matroid's structure, the resulting tests can be shown to be optimal and, hence, coincide with the tests assembled by the STA. Combinations of several matroid constraints do not, in turn, lead to matroid constraints but can be solved by finding optimal matroid intersections.

Building on these facts, matroid optimization algorithms can be used to implement optimal constrained item selection. In the case of one matroid constraint, the greedy algorithm can, for example, be used for content balancing, where it combines the simplicity and low computational expense of heuristics with the superior measurement efficiency afforded by the STA's optimality. Combinations of two matroid constraints—arising, for instance, from flexible content constraints or the need for balancing two categorical content attributes at the same time—can be solved by matroid intersection. Interestingly, matroid intersection requires optimization in terms of complete full-length feasible tests, called "shadow" tests. This indicates that optimization without look-ahead (the usual strategy employed in heuristic methods) is, in general, insufficient for optimal test assembly when complex constraints are involved. In contrast to the integer programming approach used with the STA, both the greedy algorithm and the intersection of two matroids can be executed in polynomial time. However, combinations of three or more constraints render matroid intersection NP-hard (Parker & Rardin, 2014), which defines the theoretical limitation of the matroid optimization approach.

The Problem of Constrained Item Selection

In essence, the increase in measurement accuracy enabled by CAT is achieved by optimizing the measurement properties of administered items for each examinee individually. This is done by selecting items from the item bank $\mathrm{P} = \{1, \ldots, \mathrm{n}\}$ that maximize an information criterion related to the items' usefulness for improving the trait estimate. The statistical usefulness of an item can be conceptualized in a number of ways, such as the reduction in size of posterior credible regions (Segall, 1996), Kullback-Leibler divergence of response likelihoods (Chang & Ying, 1996; Veldkamp & van der Linden, 2002) or traditionally, as Fisher's expected information. In each case, maximization of information is connected to improving measurement precision. For instance, in the case of maximum likelihood estimation (MLE), the standard error of measurement (SEM) at θ,

$$\mathrm{SE_T}(\theta) = \frac{1}{\sqrt{I_{\mathrm{T}(\theta)}}}, \tag{1}$$

depends reciprocally on Fisher information $\mathrm{I_T}$ associated with the test $\mathrm{T} = \{\mathrm{i}_1, \ldots, \mathrm{i}_{\mathrm{N_T}}\}$. Under the assumption of additivity, which is made in the remainder of this section, test information is given by

$$I_{\mathrm{T}}(\theta) = \sum_{\mathrm{i} \in \mathrm{T}} I_i(\theta), \tag{2}$$

the sum of the individual items' information. Now, the very nature of adaptive testing poses two demands on test assembly: First, and most obvious, the true trait level is unknown and thus is replaced by a provisional estimate $\hat{\theta}$; and second, item selection must be carried out during the test. Hence, the objective becomes maximizing test information, shown in Equation 2, by selecting the *k*th item based on its information I_{i_k} at the provisional trait estimate $\hat{\theta} = \hat{\theta}_{k-1}$. In the absence of any non-statistical requirements on test composition, maximum information item selection proceeds to maximize test information, choosing the kth item by

$$i_k = \text{argmax}_{i \in \text{P} \setminus \text{S}_{k-1}} I_i(\hat{\theta}_{k-1}), \quad (3)$$

where $\text{S}_{k-1} = \{i_1, \ldots, i_{k-1}\}$ is the set of items that have been administered as the first $\text{k} - 1$ items.

The problem of constrained test assembly can be stated as follows: At each position $k = 1, 2, \ldots, N$, select an item i_{k+1} that maximizes the information measure while making sure that, in the end, the complete test $\text{S}_N = \{i_1, \ldots, i_N\}$ is feasible. In general, when selecting each item, it is necessary to ascertain that a path leading to a feasible full-length test remains open. The STA (van der Linden & Reese, 1998) addresses this problem by assembling full-length feasible tests, called shadow tests, from which individual items are then selected. That is, on selection of item i_k, a feasible test $\text{S}^* \supset \text{S}_{k-1}$ is computed, containing all previously administered items and maximizing test information. This requires solving the optimization problem

$$\text{S}^* = \text{argmax}_{\text{E} \subset \text{P}, \text{S}_{k-1} \subset \text{E}} \sum_{i \in \text{E}} I_i(\hat{\theta}_{k-1}), \quad (4)$$

subject to the condition that E is a feasible test. All items in $\text{A}_k = \text{S}^* \setminus \text{S}_{k-1}$ are available to be selected and among those, item i_k is selected as the item that maximizes information,

$$i_k = \text{argmax}_{i \in \text{A}_k} I_i(\hat{\theta}_{k-1}). \quad (5)$$

By selecting items from optimal feasible tests, the STA not only guarantees that each test assembly process produces a feasible test but also that items providing maximum information at the current provisional estimate, $\hat{\theta}$, are used. Although the shadow tests are optimized for the provisional θ level, they can be shown to approximate the optimal value of the information criterion at the true θ level under reasonable conditions (van der Linden & Glas, 2010).

However, finding S^* in Equation 4 gives rise to the aforementioned computationally hard optimization problem. The approach put forward here simplifies this optimization problem by narrowing down the class of constraints under consideration to matroid constraints. Optimal item selection under matroid constraints can then be implemented based on matroid optimization with either the greedy algorithm or matroid intersection algorithms. By establishing that lower and upper bounds on categorical item attributes can be expressed as matroid constraints, the matroid optimization approach is applicable to practically relevant problems in test assembly such as content balancing and the avoidance of co-occurring "enemy items." The CAT literature describes item banks with compatible content structures—for example, Barnard, 2015; Chang and Ansley, 2003; Davis, 2004; Kröhne, Goldhammer, and Partchev, 2014; Leuders et al., 2017; Leung, Chang, and Hau, 2000, 2003; Li and Schafer, 2005; Luecht, 1996; and Yi, Zhang, and Chang, 2008, 2006**.**

The implications of the findings of this study are, hence, twofold. First, for applicable types of constraints, matroid optimization provides an alternative method of optimal constrained test assembly. In particular, the greedy algorithm provides an attractive method for the assembly of optimal tests because it is very simple to implement, while matroid intersection provides an alternative optimization method that can be used within the STA and which is, in contrast to integer programming, a polynomial time algorithm. Second, the optimality of the greedy choice can be used to analyze related heuristics such as the maximum priority index (MPI; Cheng & Chang, 2009) and Kingsbury & Zara's constrained computerized adaptive testing (CCAT; 1991). As discussed below, a modified version of CCAT, termed MCCAT (Leung et al., 2000, 2003), can be identified as an instance of the greedy algorithm. These results contribute to demarcating constraint specifications that can be treated optimally with simple algorithms and thus can guide test designers in the choice of the appropriate constrained item selection method for a particular test.

Method

The Greedy Algorithm

In order to give a formal description and analysis of the greedy item selection algorithm, it is necessary to introduce some facts and definitions from matroid theory, which can be found in Helman, Moret, and Shapiro (1993) or Lawler (1976). The formalism of set systems is used to describe the collection of feasible subtests, that is, those item sets that constitute admissible intermediate test states. Let S be a set and $\mathcal{F}$ a collection of subsets of S. Then the pair $(\mathrm{S}, \mathcal{F})$ is called a set system. Given a set system $(\mathrm{S}, \mathcal{F})$, the extension of a set $\mathrm{X} \in \mathcal{F}$ is the set of all elements of S that can be included in X yielding a set in $\mathcal{F}$, that is,

$$\text{ext}\,\mathrm{X} = \{y \in \mathrm{S}\colon \mathrm{X} \cup \{y\} \in \mathcal{F}\}. \tag{6}$$

The greedy algorithm for constrained item selection works as follows: Starting from $\mathrm{S}_0 = \emptyset$, it selects for each position $\mathrm{k} = 1, \dots, \mathrm{N}$, the most informative item from ext $\mathrm{S}_{\mathrm{k}-1}$ as the kth item. That is,

$$i_k = \text{argmax}_i\{I_i(\hat{\theta}_k)\colon i \in \mathrm{P}\backslash\mathrm{S}_{k-1}\} \tag{7}$$

If no additional assumptions are made on the structure of $\mathcal{F}$, the greedy algorithm may not be able to terminate on a maximal feasible set. Indeed, a necessary condition for the greedy algorithm to work is that each feasible set can be built from the empty set by including one element at a time. This is ensured by requiring that $\mathcal{F}$ be hereditary, that is, if $\mathrm{S} \in \mathcal{F}$ and $\mathrm{T} \subset \mathrm{S}$, it follows that $\mathrm{T} \in \mathcal{F}$. That is, in a hereditary set system, all subsets of feasible sets are feasible.

When operating on a hereditary set system, the greedy algorithm is guaranteed to arrive at a maximal feasible set M, that is, M is not contained in any other feasible set. To see this, note that if there was a feasible superset M' of M, then by heredity $\mathrm{M}'\backslash\mathrm{M} \subset \text{ext}\,\mathrm{M}$; therefore, the greedy algorithm would not have terminated on M. However, the maximal feasible sets in a hereditary set system $\mathcal{F}$ do not necessarily have the same cardinality; thus, complete tests assembled by the algorithm may have different lengths. This can be remedied by additionally requiring that $\mathcal{F}$ satisfy the axiom of augmentation, that is,

$$\mathrm{X}, \mathrm{Y} \in \mathcal{F}, |\mathrm{X}| = |\mathrm{Y}| + 1 \Rightarrow \exists x \in \mathrm{X}\backslash\mathrm{Y}\colon \mathrm{Y} \cup \{x\} \in \mathcal{F}. \tag{8}$$

By virtue of Equation 8, any feasible set in a hereditary set system can be extended to a feasible set of maximum cardinality. Consequently, all maximal feasible sets in a hereditary set system with this property have the same cardinality.

A hereditary set system that obeys the augmentation axiom is called a matroid. The rank of a matroid is the cardinality of its maximal feasible sets. Matroids are an important class of set systems in the study of greedy algorithms because the greedy algorithm optimizes all linear objective functions (Edmonds, 1971; Gale, 1968; Rado, 1957)[1]; that is, for any assignment of a real-valued weight $\mathrm{w(e)}$ to each element e of M, the greedy algorithm terminates at a maximal feasible set B for which $\mathrm{w(B)} = \sum_{\mathrm{e \in B}} \mathrm{w(e)}$ is greater than for any other feasible set. The problem of optimizing test information with respect to an additive information criterion differs from the optimization of a linear objective inasmuch as the objective function changes when the θ estimate does. The next section establishes that in optimizing test information under a matroid constraint, the greedy algorithm is equivalent to the STA and thus is, in fact, optimal.

Optimality of the Greedy Algorithm for Item Selection

The purpose of this section is to prove that under the assumption that the system of feasible subtests is a matroid, the greedy algorithm produces exactly the same test sequences as the STA does. Hence, in this case, both algorithms are equivalent, showing that the simple greedy algorithm is sufficient to solve the constrained test assembly optimally.

Proposition 1. *If* $(\mathrm{P}, \mathcal{F})$ *is a matroid, then for each fixed response pattern, the items selected by the greedy algorithm as per Equation 7 coincide with those selected by the STA as per Equation 5.*

Proof. The proof uses mathematical induction over *k* and the fact that the greedy algorithm optimizes all linear weight functions on a matroid. As the first item ($\mathrm{k} = 1$), the greedy algorithm selects

$$i_1 = \mathrm{argmax}_i \{ I_i(\hat{\theta}_0) : \{i\} \in \mathcal{F} \}. \tag{9}$$

By the optimality of the greedy algorithm on the matroid $\mathcal{F}$, the greedy algorithm, if run until termination, would arrive at the test

$$\mathrm{argmax}_{\mathrm{E \subset P, E \in \mathcal{F}}} \left\{ \sum_{l \in \mathrm{E}} I_l(\hat{\theta}_0) \right\}, \tag{10}$$

which is the test S^* used by the STA, as given in Equation 4, and from which the STA indeed proceeds to select the same maximum information item x_1 as the greedy algorithm.

Inductive step: By the inductive hypothesis, $\mathrm{S}_{k-1} \in \mathcal{F}$ is the same for both the greedy algorithm and the STA. By Equation 7, the greedy algorithm selects

[1] The exact class of set systems for which this is true, however, is larger and was characterized by Helman et al. (1993).

$$i_k = \mathrm{argmax}_{i \in \mathrm{P} \setminus \mathrm{S}_k} \{ I_i(\hat{\theta}_0) : \mathrm{S}_k \cup \{i\} \in \mathcal{F} \}. \tag{11}$$

Again, by the optimality of the greedy algorithm on the matroid $\mathcal{F}$, the greedy algorithm starting from S_{k-1}, if run until termination, would arrive at the test

$$\mathrm{argmax}_{\mathrm{E} \subset \mathrm{P}: \mathrm{S}_{k-1} \subset \mathrm{E}, \mathrm{E} \in \mathcal{F}} I_{\mathrm{E}}(\hat{\theta}_{k-1}), \tag{12}$$

which is the test S^* used by the STA as per Equation 4. Now, again, the STA's choice of the maximum information item in $\mathrm{S}^* \setminus \mathrm{S}_{k-1}$ is just the greedy choice i_k.

It should be noted that equivalence of constraints as used by the greedy algorithm and the STA, respectively, is established by equality of the maximal feasible sets (the complete feasible tests) of the matroid. That is, both algorithms are operating on equivalent constraints if the maximal feasible sets of the matroid are the sets over which STA carries out its optimization. For the matroids used in the present paper, this will be shown in the following section.

Specification of Feasible Subtests

Usually, the description of constraint-compliant tests is made on the level of complete tests. The greedy algorithm, however, works in terms of feasible subtests; therefore, it is necessary to specify the item sets that lead to complete feasible tests. The greedy algorithm applies if the collection of feasible subtests has the structural properties of a matroid. The analysis of these properties depends solely on the constraints and can be conducted on an abstract level; in addition, minimal requirements on item bank composition are identified. In what follows, a number of practically relevant constraints are covered.

Categorical Upper Bounds

Generalizing upon the results of the previous section, arbitrary upper bounds on the number of items from certain categories can be enforced as follows: Denote by $\mathrm{C}_1, \ldots, \mathrm{C}_M \subset \mathrm{P}$ the M disjoint categories. Then a test T of length N is feasible if T contains at most N_j items from category C_j. Clearly, any set of items S can be extended to a feasible test of length N if

$$|\mathrm{S}| \leq N \tag{13}$$

and

$$|\mathrm{S} \cap \mathrm{C}_j| \leq N_j, \text{ for all } j = 1, \ldots, M. \tag{14}$$

In defining the system of feasible subtests by

$$\mathrm{S} \in \mathcal{F}_{\mathrm{Upper}} :\Leftrightarrow \mathrm{S} \text{ satisfies Inequalities 13 and 14,} \tag{15}$$

it is evident that a maximal feasible subtest with respect to Inequality 13 is a feasible test of length N that adheres to the upper bound of each category. The set system $\mathcal{F}_{\mathrm{Upper}}$ is a matroid known as the partition matroid (see Lawler, 1976, for a proof of its properties). Note that the rank of the matroid defined by Inequalities 13 and 14 is $\min(\sum_{i=1}^{L} N_j, N)$. Thus, if $N \geq \sum_{i=1}^{L} N_j$ and each category holds a sufficient number of items, that is $|\mathrm{C}_j| \geq N_j$ for all j, then the greedy algorithm will terminate at a test of length N. The greedy algorithm on $\mathcal{F}_{\mathrm{Upper}}$ can be implemented very efficiently based on the observation that the extension of any feasible subtest of length k consists exactly of those items in $\mathrm{P}\backslash\mathrm{S}_{k-1}$ whose category counts have not been exceeded. Denote these categories as the *active* categories and the remaining categories as the *blocked* categories. As only items from the active categories may be selected, there is no need to evaluate the information criterion on items from blocked categories. After selecting the kth item from some active category C, there only remains to check whether category C needs to be blocked, which is a matter of counting items from C in S_k and comparing to the bound. As evaluating the information criterion can be relatively costly, the computational expense can be reduced significantly compared to a naive implementation of the algorithm.

Enemy Constraints

Test specifications may require that certain items are not administered coincidentally to one examinee. For instance, items derived from the same stem may be so similar that they clue each other (Leuders et al., 2017), such that coincidental usage would threaten local independence. This type of constraint can be enforced by using $\mathcal{F}_{\mathrm{Upper}}$ with each C_j containing a set of enemies and setting $N_j = 1$ to ensure that, at most, one item from C_j is administered. This amounts to blocking any enemy set once one of its items is administered, a procedure that coincides with the heuristic treatment of enemy constraints. By recognizing this as an instance of the greedy algorithm on the matroid $\mathcal{F}_{\mathrm{Upper}}$, it becomes obvious that (in the absence of additional constraints) a simple blocking strategy is the optimal way to handle enemy constraints.

Categorical Lower Bounds

If lower bounds are to be enforced on the number of items from certain categories $\mathrm{C}_1, \ldots, \mathrm{C}_m$, a feasible subtest S needs to have the following property: The number of items still needed to reach the lower bound of each category must not exceed the total number of items needed to complete the test. That is, denoting the lower bound of C_j by M_j, any feasible S must satisfy

$$N - |\mathrm{S}| \geq \sum_{j=1}^{m} \left(M_j - |\mathrm{S} \cap \mathrm{C}_j|\right)^+, \tag{16}$$

where

$$x^+ := \max(x, 0), \tag{17}$$

and additionally,

$$|\mathrm{S}| \leq N, \tag{18}$$

to ensure a test length of N. The system $\mathcal{F}_{\mathrm{Lower}}$ of sets satisfying Inequalities 16 and 18 is a matroid (see Lemma 1, Appendix), whose maximal feasible sets contain at least the required number of items from each category. If the sum of the lower bounds does not exceed the desired test length, the rank of $\mathcal{F}_{\text{Lower}}$ is N if only each category j is represented with at least C_j items in the item bank. Thus, under these most natural conditions, the greedy algorithm will terminate at feasible tests of full-length N.

Similar to the case of categorical upper bounds, there is a computational shortcut available that would save computational expense: If in Inequality 16, the number of items needed to complete the test (left-hand side) is strictly greater than the sum of items needed to satisfy all lower bounds (right-hand side), any item maximizing information may be selected, disregarding category membership. If equality holds, however, selection must be constrained to categories whose lower bounds are not yet met, ensuring that the right-hand side of Inequality 16 will be decremented by including the selected item. Again, the value of the information function need not be evaluated for items from the remaining categories in this case.

Matroid Intersection

Many practical applications necessitate the application of combinations of several types of constraints. For instance, flexible content constraints (Cheng, Chang, & Yi, 2007) involve a set of lower bound constraints and a set of upper bound constraints on the number of items from each content category. Similarly, by enforcing a set of upper bound constraints for each pair of categorical attributes, content balancing can be achieved for both attributes simultaneously, allowing, for example, balancing test composition with respect to both content area and answer key. In the matroid optimization approach to item selection, this involves two matroids specified on the item bank. Let M_1 and M_2 be two matroids defined on the item bank P. Then, assembling feasible tests requires that during the test, all subtests are feasible in both M_1 and M_2. However, the set system

$$M_1 \cap M_2 = \{A \cap B | A \in M_1, B \in M_2\} \tag{19}$$

containing all sets that are feasible in both matroids is, in general, not a matroid. Thus, the greedy algorithm is not guaranteed to succeed in building optimal feasible tests. In particular, it is possible that simply selecting maximally informative items, such that the resulting subtest is feasible in both M_1 and M_2, quickly leads to a subtest that cannot be extended, thwarting the assembly of full-length tests.

The related problem of finding a set of maximal weights that is feasible in several matroids has been considered in the literature as "weighted matroid intersection." Although weighted matroid intersection is NP-hard if at least three matroids are involved, polynomial time algorithms for the weighted intersection of two matroids have been devised by, among others, Brezovec, Cornuéjols, and Glover (1986), Edmonds (1979), and Lawler (1976). By using item information $I_i(\hat{\theta}_{k-1})$ as the weight of item i, any of these algorithms can be employed in constrained test assembly. In contrast to the case of one matroid constraint, the feasibility of assembling full-length tests depends on the combination of attributes in the item bank and must, as is the case with STA, be tested in simulation. For the simulation study presented below, a procedure was used based on the first algorithm devised by Brezovec et al.(1986) that returns optimal intersections of the desired length. Using this algorithm, an optimal intersection is computed for each position, and the most informative item is chosen from this intersection for the present provisional estimate of θ. Items that have already been administered are held fixed in the intersection by assigning large weights to them. This procedure can be subsumed under the STA; however, the actual optimization step is carried out by a matroid algorithm. Thus, the theoretical properties of the STA—most importantly, optimality—directly carry over.

Relation to Other Methods

CCAT

Kingsbury and Zara (1991) introduced the CCAT algorithm, which balances content composition of tests under an upper bound constraint on one categorical attribute in a very straightforward fashion. First, the content area that deviates most from its bound is identified; then, item selection is constrained to items belonging to this content area. The procedure, while exceedingly easy to implement, has two drawbacks: The sequence of content areas of administered items can be quite predictable, raising test security concerns (Chen & Ankenman, 2004). Moreover, item selection is constrained more than necessary, leading to an avoidable loss of measurement efficiency. Addressing the first drawback, Leung et al. (2000, 2003) proposed a modification of CCAT. Their MCCAT selects the most informative item from any category that is below its upper bound. This procedure coincides precisely with the greedy algorithm on the respective upper bound matroid introduced above and, hence, by Proposition 1 as well as with the STA.

STA

As described above, the greedy algorithm produces tests that coincide with those produced by the STA, provided that the system of feasible subtests has the properties of a matroid. As discussed below, the same holds for the combination of two matroid constraints by the use of matroid intersection. However, the greedy algorithm is distinct from the STA in that it does not require the computation of shadow tests. Instead, optimal feasible tests can be built by adding one item at a time. Compared to the STA, implementing the greedy algorithm is significantly less involved and requires less computational effort. In general, optimal intersections of matroids cannot be found by a greedy algorithm; hence, the application of matroid intersection to constrained test assembly requires optimizing in terms of complete optimal tests. Test assembly by matroid intersection is thus a variant of the STA, albeit using a different optimization technique. In contrast to integer programming, it can be executed in polynomial time if two matroids are intersected. The fact that combinations of constraints must be optimized in terms of complete tests sheds light on the boundary between simple and hard problems in constrained test assembly and, at the same time, provides some insight into the theoretical limitations of greedy-like heuristics such as those discussed below.

MPI

The MPI method (Cheng & Chang, 2009) works by weighting item information using a priority index. The priority index is specified for the constraint at hand. For content balancing, the priority index can be defined as the remaining quota of items from a specific content category. As more items from the content area are selected, the priority index decreases and the inclusion of further items from the same content area becomes less likely. As a consequence of attaching lower weights to categories that have seen some usage, MPI tends to favor items from categories that deviate further from their target administration counts. Therefore, items from these categories are disregarded even if a more informative item could be chosen without violating the constraints, entailing a loss of efficiency compared to the optimal item choice.

Thus, in general, MPI is not maximally efficient. However, using Proposition 1, it can be shown that MPI solves the case of enemy constraints optimally, as it coincides with the greedy algorithm in this case. To exemplify this, suppose that items belonging to the same set of enemies $\mathrm{E}_1, \ldots, \mathrm{E}_m \subset \mathrm{P}$ may not be administered in one test. In the framework of MPI, the scaled quota left for the constraint associated with E_l is $f_l = 1 - x_l$, where

$$x_l = \begin{cases} 1, & \text{if an item from } \mathrm{E}_l \text{ has been used} \\ 0, & \text{otherwise} \end{cases} \tag{20}$$

As a consequence, assuming that all constraints have weight 1, the priority index of item i will be zero if i belongs to an enemy set from which an item has already been used; otherwise, the priority index of item i will be its unweighted information. In effect, MPI will choose the item having maximum information that is not in violation with any enemy constraint. This is exactly the greedy choice and, equivalently, the choice made by the STA for this set of constraints.

Simulation Study

The simulation study was based on an item bank of 500 synthetic items assumed to be calibrated under a two-parameter logistic model to a level of precision that allows treating item parameters as known during the adaptive test. The item bank was generated by drawing item difficulties and item discriminations from a standard normal distribution and a log-normal distribution with mean 0.25 and standard deviation (SD) of 0.125, respectively. From this item bank, derived item banks were generated by assigning a trichotomous categorical attribute, *content category*, in different ways. In item bank A, item difficulties and assigned content areas were independent. In item bank B, items from Category 1 were more difficult [mean = -1.11, standard deviation (SD) $= 0.83$] and items from Category 3 were easier (mean = -1.50 SD $= 0.76$) than those from Category 2 (mean = -0.05, SD = 0.87). Additionally, a second categorical attribute, *answer key*, was introduced. It assumed four different, uniformly distributed values and was used only in the condition involving upper bounds on two attributes. The influence on item bank size was investigated by comparing the complete item banks A and B against smaller versions generated by sampling a 250- and a 125-item subset from each. θ estimates were computed using the expected a posteriori estimator with a standard normal prior. Constraint specifications were used that involved either lower or upper bound constraints on one categorical attribute, as well as flexible content constraints and upper bound constraints on two categorical attributes. The values used for the bounds are summarized in Table 1. In all simulation runs, the test length was 30.

Table 1. Values of Upper and Lower Bounds Used in the Simulations

	Content Category			Answer Key			
Value	0	1	2	0	1	2	3
Lower bound	5	5	5	-	-	-	-
Upper bound	11	18	11	10	10	10	10

Average SEM (Equation 1) realized by matroid optimization, MPI, and CCAT (where applicable) across 5,000 replications are reported, where each replication was carried out for seven equidistant values of θ between −2 and 2. As a baseline, the SEM associated with the best feasible test is reported. The latter was computed by any of the equivalent optimal algorithms (STA or matroid optimization), using the true θ instead of an estimate. The best feasible test's information can be understood as effective item bank information when taking into account constraints on test length and test composition. In order to allow interpretation of the differences in SEM, the potential saving in test length that could be realized by choosing the best-performing algorithm over the others is reported, whereas raw SEMs are reported in Table 2. The reported number was determined by successively cutting the test length of the bestmethod until measurement precision was decreased to the level of the other methods for each θ; then, the maximum over the θ grid was taken.

Table 2. Mean SEM Attained by Item Selection Algorithms for Different Types of Constraints Using Item Banks with 125, 250, and 500 Items

Type of Constraint and Method	Item Bank A			Item Bank B		
	125	250	500	125	250	500
Upper bound						
Best feasible test	0.606	0.464	0.406	0.626	0.479	0.415
Matroid optimization	0.609	0.467	0.409	0.630	0.482	0.419
MPI	0.614	0.469	0.411	0.640	0.490	0.427
CCAT	0.619	0.472	0.414	0.655	0.501	0.440
Lower bound						
Best feasible test	0.604	0.464	0.405	0.620	0.477	0.418
Matroid optimization	0.607	0.466	0.408	0.624	0.480	0.422
MPI	0.607	0.466	0.408	0.624	0.480	0.422
CCAT	0.607	0.466	0.408	0.624	0.480	0.422
Flexible constraint						
Best feasible test	0.606	0.464	0.406	0.627	0.481	0.420
Matroid optimization	0.609	0.467	0.409	0.631	0.484	0.425
MPI	0.614	0.469	0.411	0.640	0.490	0.428
CCAT	0.619	0.472	0.414	0.655	0.501	0.440
Two upper bounds						
Best feasible test	0.606	0.465	0.406	0.626	0.479	0.415
Matroid optimization	0.609	0.468	0.409	0.631	0.483	0.419
MPI	0.615	0.473	0.413	0.641	0.492	0.429

Note. The results of STA coincided—as postulated by theory—with those of matroid optimization and, hence, are not shown.

Computational Expense

During the simulation runs, precise computation times were recorded using the high-resolution clock of the computer (see Table 3). With mean computation times between approximately 0.3 and 30 milliseconds, all item selectors were fast enough for practical application by a large margin. In the settings involving one constraint, the simple algorithms (MPI, CCAT, and the greedy algorithm) were quite close to each other, but 10 to 20 times faster

Table 3. Mean Computation Times (Standard Error of the Mean in Parentheses) Measured in Milliseconds

Type of Constraint and Method	Item Bank A			Item Bank B		
	125 Items	250 Items	500 Items	125 Items	250 Items	500 Items
Upper bound						
Matroid optimization	0.31 (0.001)	0.34 (0.001)	0.38 (0.001)	0.33 (0.001)	0.35 (0.001)	0.38 (0.001)
STA	3.82 (0.003)	7.20 (0.006)	13.15 (0.009)	3.82 (0.003)	7.03 (0.005)	13.33 (0.011)
MPI	0.36 (0.001)	0.42 (0.001)	0.48 (0.001)	0.36 (0.001)	0.43 (0.001)	0.45 (0.001)
CCAT	0.35 (0.001)	0.36 (0.001)	0.39 (0.001)	0.33 (0.001)	0.35 (0.001)	0.39 (0.001)
Lower bound						
Matroid optimization	0.27 (0.001)	0.30 (0.001)	0.34 (0.001)	0.29 (0.001)	0.31 (0.001)	0.35 (0.001)
STA	3.78 (0.003)	7.24 (0.005)	13.15 (0.009)	3.81 (0.003)	7.02 (0.004)	13.51 (0.012)
MPI	0.29 (0.001)	0.32 (0.001)	0.36 (0.002)	0.28 (0.001)	0.31 (0.001)	0.36 (0.002)
CCAT	0.27 (0.001)	0.28 (0.001)	0.34 (0.001)	0.28 (0.001)	0.29 (0.001)	0.32 (0.001)
Flexible constraint						
Matroid optimization	4.04 (0.004)	5.53 (0.009)	10.62 (0.026)	4.90 (0.017)	10.89 (0.038)	24.31 (0.078)
STA	7.50 (0.027)	14.94 (0.066)	29.21 (0.110)	8.34 (0.054)	14.10 (0.053)	20.53 (0.117)
MPI	0.39 (0.004)	0.41 (0.002)	0.51 (0.007)	0.41 (0.004)	0.44 (0.005)	0.49 (0.007)
CCAT	0.37 (0.004)	0.37 (0.002)	0.41 (0.006)	0.38 (0.003)	0.40 (0.005)	0.43 (0.004)
Two upper bounds						
Matroid optimization	3.50 (0.003)	6.04 (0.009)	11.85 (0.031)	3.50 (0.005)	6.37 (0.010)	11.78 (0.023)
STA	5.16 (0.003)	9.70 (0.005)	19.02 (0.011)	5.05 (0.003)	9.93 (0.005)	18.54 (0.010)
MPI	0.47 (0.000)	0.54 (0.001)	0.65 (0.001)	0.47 (0.000)	0.54 (0.001)	0.61 (0.000)

Note. All calculations were implemented on a 2Ghz Intel Xeon E5-2620 CPU equipped with 64GB of RAM.

than the STA. When a second constraint was added, MPI became only slightly slower. The increase in computational time needed to compute matroid intersections over applying the greedy algorithm was more pronounced, but the computation times were at least competitive to the STA, and in many settings significantly shorter. These results suggest that the greedy algorithm can be implemented just as computationally efficiently as the heuristic methods of MPI and CCAT and that the computation times for matroid intersection are comparable to those of the integer programming model commonly employed in the STA. These timing results must be interpreted with some caution, however, as they reflect properties of the concrete implementations used. At least for the matroid intersection—which is more complex than, for example, the greedy algorithm—more efficient implementations than the one used here may very well be devised.

Lower Bound Constraints

Among the item selection algorithms tested, only the STA and the greedy algorithm support lower bound constraints natively, while for MPI and CCAT, a two-phase item selection procedure (Cheng et al., 2007) needs to be used. In the first phase, the lower bounds are treated as upper bounds in a test of length $N_1 = \sum_{i=1}^{m} M_i$. Thus, at the end of the first phase, the lower bounds are satisfied, and unconstrained item selection can be used in the second phase for the remaining $N - N_1$ items. The differences in SEM between the methods found in this setting are negligibly small (see Table 2). As described in the next section, item selection with upper bounds produces marked differences between the methods. Hence, this is attributable to the second, unconstrained phase of item selection that makes up for the second half of the test and apparently suffices to all but level out any differences accumulated in the first phase.

Upper Bound Constraints

The SEM realized by the greedy algorithm was lower than that of CCAT and MPI (see Table 2). The difference between the methods varied markedly between the two item banks. For item bank A, which exhibited no confounding of categories and item parameters, differences were relatively small, amounting to an approximate one-item advantage of the greedy algorithm over MPI and CCAT. The small difference was expected, since each category offers a large number of items at every level of θ. Consequently, the limiting or biasing of item selection toward under-represented categories in CCAT and MPI is without severe consequences, but still amounts to a difference of one to three items. However, item bank B, with its pronounced dependency between item difficulty and category membership, led to more pronounced differences among the item selection methods. Here, the SEM realized by CCAT was also greatest, while MPI fared markedly better than CCAT. The advantage of the greedy algorithm corresponded to four items over MPI and eight to nine items over CCAT (see Table 4).

Flexible Constraints

Flexible content constraints involve imposing a lower bound, as well as an upper bound, on the number of items from each content area. As described above, both lower bounds and upper bounds induce a matroid on the item bank. Hence, management of flexible constraints can be achieved by matroid intersection. This allows for an optimal solution that provides full adaptivity, equivalent to the STA. While the latter allows specifying flexible constraints directly, implementing flexible

Table 4. Difference in Mean SEM Between Item Selection Methods Expressed in Terms of the Number of Items Required to Reach Measurement Precision of the Best Performing Method

Type of Constraint and Method	Item Bank A			Item Bank B		
	125	250	500	125	250	500
Upper bound						
Matroid optimization	-	-	-	-	-	-
MPI	2	1	2	4	4	4
CCAT	3	2	3	8	8	9
Lower bound						
Matroid optimization	-	-	-	-	-	-
MPI	-	-	-	-	-	-
CCAT	-	-	-	-	-	-
Flexible constraint						
Matroid optimization	-	-	-	-	-	-
MPI	2	1	2	4	4	4
CCAT	3	2	3	8	10	10
Two upper bounds						
Matroid optimization	-	-	-	-	-	-
MPI	2	3	3	5	5	5

Note. The results of STA coincide—as postulated by theory—with those of matroid optimization and, hence, are not shown.

constraints with MPI and CCAT requires the application of a two-phase item selection procedure (Cheng & Chang, 2009; Cheng et al., 2007). Although thisapproach could, in principle, be used with the greedy algorithm as well, the two-phase procedure effectively divides the test into two adaptive testlets, which comes at the cost of reduced adaptivity and efficiency. Empirically, this is reflected in the superior performance of matroid intersection (see Tables 2 and 4). The advantage over MPI amounts to a difference in test length of one to two items for item bank A and four items for item bank B, while CCAT is between two and three items (item bank A) and eight to 10 items (item bank B) behind.

Two Upper Bounds

Upper bounds on two distinct categorical attributes—here, content category and answer key—can be managed in the matroid framework by intersecting two upper bound matroids. As managing multiple categorical attributes is out of the scope of CCAT, it is not available in this setting.[2] Consistent with the other settings, the theoretically optimal matroid-based algorithm achieves a smaller SEM than MPI across all item banks (see Table 2). Again, the size of the gap depends on the conditional distribution of difficulty by content category, amounting to a difference in test length of two to three items for item bank A and five items for item bank B.

[2] In principle, CCAT could be used by forming the Cartesian product of categories and imposing constraints on each combination of categories. However, this leads to a more severely constrained tests and, during preliminary trials, was found to lead to non-competitive performance.

Discussion

Constraint specifications involving upper or lower bounds on categorical item attributes were shown to introduce a matroid on the item bank, thus enabling the use of matroid optimization algorithms for constrained item selection. Given equivalent sets of constraints, the tests assembled by matroid optimization coincided with those assembled by the STA. Thus, the matroid optimization approach can be understood as an alternative to the STA, which, by exploiting the structure warranted by matroid constraints, can be computed at a reduced computational complexity. The matroid optimization approach and the STA can be subsumed as methods using combinatorial optimization, whereas matroid intersection takes the role of an alternative optimization technique within the STA. Of the matroid optimization algorithms that have been applied in the present work, the greedy algorithm seemed to be of greater practical appeal, as it combines the simplicity of heuristic item selection methods such as MPI and CCAT with the theoretical guarantees and optimality afforded by the STA. Furthermore, by using the optimality of the greedy item choice as an analytic tool, it becomes obvious where exactly the heuristics fall short of attaining optimality even for simple constraints—and if they, conversely, as in the case of MCCAT, are actually optimal.

Using efficient algorithms for weighted matroid intersection, the combination of two matroid constraints can be treated. In this case, however, more involved algorithms for matroid intersection are required, and optimization has to be implemented in terms of complete tests. The latter is of considerable interest, as it highlights the necessity of using look-aheads for optimal test assembly when complex constraints are involved.

The range of problems solvable by matroid optimization remains limited by two factors. First, only matroid constraints can be used and, although other examples of matroids relevant to test assembly may be identified, not all conceivable types of constraints should be expected to have this property. Second, solving the intersection problem across three and more matroids is NP-hard just as the 0-1 integer program, which is commonly employed within the STA.

The matroid-based approach thus has two advantages: First, the greedy algorithm combines the simplicity and efficiency of heuristic methods with the optimality of the significantly more complicated mathematical programming approaches, and matroid intersection provides an alternative optimization algorithm that can be used with the STA. Second, matroid theory opens up a new perspective on the structure of constrained test assembly problems, which can be used for the analysis of existing methods as well as in the implementation of CATs and, hence, allows for making theoretically grounded choices in item selection methods. The simulation study presented above indicates that, even for the simplest case of an upper bound on one categorical attribute, the advantage of the matroid optimization approach over common heuristics can be equivalent to a reduction in test length of up to 30%. The advantage of the optimization approach became particularly obvious when mean item difficulties varied across content categories, whereas item bank size did not seem to impact relative differences between the methods. While the STA is equivalent in terms of measurement efficiency, using matroid optimization by the greedy algorithm in place of the STA rids CAT programs from the dependency on external solvers, thus enabling savings in development, maintenance, and licensing costs. Furthermore, the greedy algorithm can easily be implemented and run on platforms that either lack available solvers or computational power, thereby enabling the delivery of efficient CAT programs on a wider range of platforms and devices.

References

Barnard, J. J. (2015). Implementing a CAT: The AMC experience. *Journal of Computerized Adaptive Testing*, *3*(1), 1–12. CrossRef

Brezovec, C., Cornuéjols, G., & Glover, F. (1986). Two algorithms for weighted matroid intersection. *Mathematical Programming*, *36*(1), 39–53. CrossRef

Chang, H.-H., & Ying, Z. (1996). A global information approach to computerized adaptive testing. *Applied Psychological Measurement*, *20*(3), 213–229. CrossRef

Chang, S.-W., & Ansley, T. N. (2003). A comparative study of item exposure control methods in computerized adaptive testing. *Journal of Educational Measurement*, *40*(1), 71–103. CrossRef

Chen, S.-Y., & Ankenman, R. D. (2004). Effects of practical constraints on item selection rules at the early stages of computerized adaptive testing. *Journal of Educational Measurement*, *41*(2), 149–174. CrossRef

Cheng, Y., & Chang, H.-H. (2009). The maximum priority index method for severely constrained item selection in computerized adaptive testing. *British Journal of Mathematical and Statistical Psychology*, *62*(2), 369–383. CrossRef

Cheng, Y., Chang, H.-H., & Yi, Q. (2007). Two-phase item selection procedure for flexible content balancing in CAT. *Applied Psychological Measurement*, *31*(6), 467–482. CrossRef

Davis, L. L. (2004). Strategies for controlling item exposure in computerized adaptive testing with the generalized partial credit model. *Applied Psychological Measurement*, *28*(3), 165–185. CrossRef

Edmonds, J. (1971). Matroids and the greedy algorithm. *Mathematical Programming*, *1*(1), 127–136. CrossRef

Edmonds, J. (1979). Matroid intersection. In P.L. Hammer, E. L. Johnson, & B. H. Korte (Eds.), *Discrete Optimization I: Vol. 4. Proceedings of the Advanced Research Institute on Discrete Optimization and Systems Applications of the Systems Science Panel of NATO and of the Discrete Optimization Symposium* (pp. 39–49). Amsterdam, The Netherlands: Elsevier. CrossRef

Gale, D. (1968). Optimal assignments in an ordered set: An application of matroid theory. *Journal of Combinatorial Theory*, *4*(2), 176–180. CrossRef

He, W., Diao, Q., & Hauser, C. (2014). A comparison of four item-selection methods for severely constrained CATs. *Educational and Psychological Measurement*, *74*(4), 677–696. CrossRef

Helman, P., Moret, B. M. E., & Shapiro, H. D. (1993). An exact characterization of greedy structures. *SIAM Journal on Discrete Mathematics*, *6*(2), 274–283. CrossRef

Karp, R. M. (1972). Reducibility among combinatorial problems. In R. Miller (Ed.), *IBM Research Series: Complexity of computer computations.* (pp. 85–103). New York, NY: Springer. CrossRef

Kingsbury, C. G., & Zara, A. R. (1991). A comparison of procedures for content-sensitive item selection in computerized adaptive tests. *Applied Measurement in Education*, *4*(3), 241–261. CrossRef

Kröhne, U., Goldhammer, F., & Partchev, I. (2014). Constrained multidimensional adaptive testing without intermixing items from different dimensions. *Psychological Test and Assessment Modeling*, *56*(4), 348–367.

Land, A. H., & Doig, A. G. (1960). An automatic method of solving discrete programming problems. *Econometrica*, *28*(3), 497–520. CrossRef

Lawler, E. L. (1976). *Combinatorial optimization: Networks and matroids*. North Chelmsford, MA: Courier Corporation.

Leuders, T., Bruder, R., Kroehne, U., Naccarella, D., Nitsch, R., Henning-Kahmann, J., … Wirtz, M. (2017). Development, validation, and application of a competence model for mathematical problem solving by using and translating representations of functions. In D.Leutner, J. Fleisher, J. Grűkorn, & E. Kleime (Eds.), *Competence Assessment in Education Series: Methodology of educational measurement and assessment* (pp. 389–406). Cham, Switzerland: Springer International Publishing. CrossRef

Leung, C.-K., Chang, H.-H., & Hau, K.-T. (2000, April). *Content balancing in stratified computerized adaptive testing designs*. Paper presented at the annual meeting of the American Educational Research Association, New Orleans.

Leung, C.-K., Chang, H.-H., & Hau, K.-T. (2003). Computerized adaptive testing: A comparison of three content balancing methods. *The Journal of Technology, Learning and Assessment*, *2*(5), 1-16.

Li, Y. H., & Schafer, W. D. (2005). Trait parameter recovery using multidimensional computerized adaptive testing in reading and mathematics. *Applied Psychological Measurement*, *29*(1), 3–25. CrossRef

Luecht, R. M. (1996). Multidimensional computerized adaptive testing in a certification or licensure context. *Applied Psychological Measurement*, *20*(4), 389–404. CrossRef

Padberg, M., & Rinaldi, G. (1991) A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM Review*, *33*(1) 60-100. CrossRef

Parker, R.G., & Rardin, R.L. (2004) *Discrete optimization.* Elsevier.

Rado, R. *(1957)* Note on independence functions. *London Mathematical Society: Series 3, Vol. 7. Proceedings of the London Mathematical Society (pp. 300-320).* CrossRef

Segall, D.O. (1996) Multidimensional adaptive testing. *Psychometrika*, *6*(2), 331-354. CrossRef

Segall, D.O. (2005) Computerized adaptive testing. In K. Kempf-Leonard (Ed.), *Encyclopedia of Social Measurement* (pp. 429-438). New York, NY: Elsevier. CrossRef

van der Linden, W. J., & Glas, C. A. W. (Eds.). (2010). *Elements of adaptive testing*. New York, NY: Springer. CrossRef

van der Linden, W. J., & Reese, L.M. (1998). A model for optimal constrained adaptive testing. *Applied Psychological Measurement*, *22*(3), 259–270. CrossRef

Veldkamp, B. P., & van der Linden, W. J. (2002). Multidimensional adaptive testing with constraints on test content. *Psychometrika*, *67*(4), 575–588. *CrossRef*

Weiss, D. J. (1982). Improving measurement quality and efficiency with adaptive testing. *Applied Psychological Measurement*, *6*(4), 473–492. *CrossRef*

Yi, Q., Zhang, J., & Chang, H.-H. (2006). *Severity of organized item theft in computerized adaptive testing: An empirical study*. (ETS Research Report Series RR-06-22). Princeton, NJ: Educational Testing Service. *CrossRef*

Yi, Q., Zhang, J., & Chang, H.-H. (2008). Severity of organized item theft in computerized adaptive testing: A simulation study. *Applied Psychological Measurement*, *32*(7), 543–558. *CrossRef*

Author's Address

Daniel Bengs, Deutsches Institut für Internationale Pädagogische Forschung, Schloßstraße 29, 60486 Frankfurt, Germany. Email: bengs@dipf.de.

Appendix

Proof of Lemma 1

Lemma 1 can be proved by showing that $\mathcal{F}_{\text{Lower}}$ has the properties of heredity and allows augmentation and, thus indeed, is a matroid.

Lemma 1. *Let* P *be an item bank and* $\{\mathrm{C_j} \subset \mathrm{P} | \mathrm{j} = 1, \dots, \mathrm{L}\}$ *a mutually disjoint system of item categories with non-negative lower bounds* $\mathrm{M_1}, \dots, \mathrm{M_L}$. *If* N *is such that*

$$N \geq \sum_{j=1}^{L} M_j, \tag{21}$$

then the set system $\mathcal{F}_{\text{Lower}}$ *of feasible subtests* $\mathrm{S} \subset \mathrm{P}$ *with the properties*

$$N - |\mathrm{S}| \geq \sum_{j=1}^{L} \left(M_j - |\mathrm{S} \cap \mathrm{C}_j|\right)^{+} \tag{22}$$

and

$$|\mathrm{S}| \leq N \tag{23}$$

is a matroid of rank N.

Proof. Heredity. Clearly, $\mathcal{F}_{\text{Lower}}$ is non-empty since $\emptyset \in \mathcal{F}_{\text{Lower}}$. The claim of heredity follows from the assertion that if S is feasible and x ∈ S, then $\mathrm{S}\backslash\{x\}$ is feasible. Let $\mathrm{S} \in \mathcal{F}_{\text{Lower}}$ and x ∈ S. Then S\{x} is feasible if and only if

$$N - |\mathrm{S}\backslash\{x\}| = N - |\mathrm{S}| + 1 \geq \sum_{j=1}^{L} \left(M_j - \left|(\mathrm{S}\backslash\{x\}) \cap \mathrm{C}_j\ \right|\right)^{+}. \tag{24}$$

Without loss of generality, it may be assumed that $x \in \mathrm{C}_{j^*}$, for some $1 \leq j^* \leq L$, because if the categories do not exhaust the item bank, a residual category with lower bound 0 may be introduced without altering $\mathcal{F}_{\text{Lower}}$. Thus, Inequality 24 may be written as

$$N - |\mathrm{S}| + 1 \geq \sum_{j \in \{1,\dots,L\}, j \neq j^*} \left(M_j - |\mathrm{S} \cap \mathrm{C}_j|\right)^{+} + \left(M_{j^*} - \left|\left(\mathrm{S}\backslash\{x\} \cap \mathrm{C}_{j^*}\right)\right|\right)^{+} \tag{25}$$

which by subtraction of Inequality 22 for S transforms to

$$1 \geq \left(M_{j^*} - \left|\left(\mathrm{S}\backslash\{x\} \cap \mathrm{C}_{j^*}\right)\right|\right)^+ - \left(M_{j^*} - \left|\mathrm{S} \cap \mathrm{C}_{j^*}\right|\right)^+. \tag{26}$$

As $\left|(\mathrm{S}\backslash\{x\}) \cap C_{j^*}\right| = \left|\mathrm{S} \cap \mathrm{C}_{j^*}\right| - 1$, which is true because by assumption $x \in \mathrm{C}_{j^*}$, Inequality 26 holds if and only if

$$1 \geq \left(M_{j^*} - \left|\mathrm{S} \cap \mathrm{C}_{j^*}\right| + 1\right)^+ - \left(M_{j^*} - \left|\mathrm{S} \cap \mathrm{C}_{j^*}\right|\right)^+. \tag{27}$$

Inequality 27 holds because for any real number ν,

$$(\nu + 1)^+ - \nu^+ = \begin{cases} 0, & \text{if } \nu < -1, \\ \nu + 1, & \text{if } -1 \leq \nu < 0, \\ 1, & \text{if } \nu \geq 0. \end{cases} \tag{28}$$

Augmentation. Let $\mathrm{S}, \mathrm{T} \in \mathcal{F}_{\mathrm{Lower}}$, such that $|\mathrm{T}| = |\mathrm{S}| - 1$. If for T in Inequality 22, the left-hand side is strictly greater than the right-hand side, any $x \in \mathrm{T}\backslash\mathrm{S}$ can be added to T yielding a feasible subtest. Thus, it may be assumed that

$$N - |\mathrm{S}| = \sum_{j=1}^{L} \left(M_j - |\mathrm{S} \cap \mathrm{C}_j|\right)^+ \tag{29}$$

and $\mathrm{T} \cup \{x\}$ is feasible if and only if adding x to T decreases the right-hand side of the last equation, which holds true if x belongs to a content category whose lower bound has not yet been met. Assume that

$$\forall j: \left(M_j - \left|\mathrm{T} \cap \mathrm{C}_j\right|\right)^+ \leq \left(M_j - \left|\mathrm{S} \cap \mathrm{C}_j\right|\right)^+. \tag{30}$$

Then, it follows that

$$\begin{aligned} N - |\mathrm{T}| &= \sum_{j=1}^{L} \left(M_j - \left|\mathrm{T} \cap \mathrm{C}_j\right|\right)^+ \\ &\leq \sum_{j=1}^{L} \left(M_j - \left|\mathrm{S} \cap C_j\right|\right)^+ \\ &\leq N - |\mathrm{S}| \\ &= N - |\mathrm{T}| - 1, \end{aligned} \tag{31}$$

which is a contradiction. Thus, it exists j^* such that

$$\left(M_{j^*} - \left|\mathrm{T} \cap \mathrm{C}_{j^*}\right|\right)^+ > \left(M_{j^*} - \left|\mathrm{S} \cap \mathrm{C}_{j^*}\right|\right)^+ \tag{32}$$

Therefore $\left|\mathrm{T} \cap \mathrm{C}_{j^*}\right| < \left|\mathrm{S} \cap \mathrm{C}_{j^*}\right|$, which implies existence of $x \in \mathrm{S} \cap \mathrm{C}_{j^*}$ with $x \notin \mathrm{T}$ (and thus $x \in \mathrm{S}\setminus\mathrm{T}$) such that

$$\left(M_{j^*} - \left|\mathrm{T} \cap \mathrm{C}_{j^*}\right|\right)^+ = \left(M_{j^*} - \left|(\mathrm{T} \cup \{x\}) \cap \mathrm{C}_{j^*}\right|\right)^+ + 1. \tag{33}$$

`