Measurement and Research Department Reports 94-1

AN ATTEMPT TO QUANTIFY LIMITS ON REPEATED TEST CONSTRUCTION

T.J.J.M. Theunissen

Measurement and Research Department Reports 94-1

AN ATTEMPT TO QUANTIFY LIMITS ON REPEATED TEST CONSTRUCTION

T.J.J.M. Theunissen

Cito
Arnhem, 1994

Abstract

In an item bank, on occasion some descriptive categories will be represented by a relatively small number of items. Because of test specification, transient categories may arise by intersection of original descriptive categories and may also be badly represented by items. In such cases, test construction may have to deal with combinatorial limits. The purpose of this paper is to investigate these limits.

Key words: test construction, item categories, error control codes.

Introduction

It is not usual to hear about limits on test construction in an item bank context: rather the opposite is the case. Statements like 'the number of possible tests to be constructed from n items is $(2^n - 1)$' and 'if a category contains n items and we need w of them in our test this can be done in $\binom{n}{w}$ ways' are not uncommon. Since these figures are reassuringly large in non-trivial cases, they tend to induce a feeling that all is well. Even if one regards test construction as a constrained optimization problem, the notion that generally in test construction the number of constraints is relatively small in comparison with many other optimization problems, will not easily lead to thoughts on limits on test construction. The problem is, that the above figures are context-free.

Since many item banks contain substantial numbers of items, ranging from hundreds to several thousands of items it is convenient to impose some descriptive structure on the collection. In order to do so, some kind of category system must be used. A category can be regarded as a descriptive label that applies to an item; an item may be described by as many categories as are deemed necessary. For example, categories can be used to describe the content domains to which items belong or the type of item, such as knowledge or insight items. Further examples are the item format, for example multiple choice or yes-no items, the difficulty level of items, the presence of gender or minority group reference, the number of blanks in a cloze test items and so on. The membership of items of various categories and transient categories resulting from intersections of categories then determines the actual structure of an item bank. It is in fact this actual structure that is important in determining whether there are limits on test construction in those cases where item banks will be used repeatedly with more or less similar test specifications.

The reason for this is that, once an item bank is in use and tests have been constructed, on many occasions future test construction has to take past construction into account. It is this factor, in conjunction with the actual structure of the item bank, that may cause limitations of the possibilities of test construction. Item banks for which the following material might be relevant are either small item banks with simple structure or large item banks with complex structure. The test construction problem addressed in this paper does not concern the actual design of tests. This paper deals with the question of how often a test can be constructed, given a particular item bank as mentioned above, and a specific test specification that has to be used repeatedly. As will be argued further on, frequently situations will arise where the test specification implies selection of items from small categories. If in such situations repeated test construction according to the same specification is demanded, the possibilities are very quickly exhausted if no overlap between tests is allowed. This paper is not relevant for situations where no overlap is allowed. One could argue that if overlap is allowed, it would make more sense to formulate this at the test form level instead of a single category. This, however, would be missing the point. If the selection of items from one or several small categories is an intrinsic part of the test specification and repeated test construction is a necessity then overlap must be allowed at the category level or repetition will be severely restricted. The material in this paper enables one to study the effects of allowing specified amounts of overlap. At first glance this might seem to be a relatively simple combinatorial problem. As it happens, it is extremely tough for non-trivial cases. Fortunately, there is a formal analogy between some aspects of the theory of error-correcting codes and this combinatorial problem in repeated test construction; this issue will come up again later. Since this paper does not deal with test construction as such it ignores, apart from category constraints, various other quantitative constraints such as, for example, identical information functions, p-values in certain intervals and others. However, it should be clear that combinatorial limitations due to a category system, although existing independently of other constraints, may have repercussions for the ease with which such constraints can be met. This would imply some form of sensitivity analyses which would be far beyond the intention of this paper. In the remainder of this paper the following issues will be addressed: more remarks on the actual structure of the itembank; a combinatorial analysis of an item selection process; why this poses a difficult problem; what are the solutions, and finally, the combinatorial effects of a specific type of constraint.

The Actual Structure of an Item Bank

In this paper primary categories are those categories that are used to describe the items in an item bank and are also utilized by the user of an item bank in the specification of the test to be constructed. It is convenient to regard the items in the bank as the universal set and the primary categories as the names of subsets of items. If the number of elements in each subset is known a fair description of the item bank can be given. This description, however, says very little about the actual structure of the bank. For this, knowledge is needed about the number of elements in other sets that are related to the original subsets by the elementary set operations of union, intersection and complementation. Since the main interest is in limiting factors we will concentrate on intersections. The results of these intersections will be called categories.

In practice, the following situations will occur frequently:

1. The item bank is relatively small and the number of primary categories is relatively small or large;
2. The bank is relatively large and the number of primary categories is relatively large;
3. The bank is of any size and the distribution of items over the primary categories is haphazard which might occur with banks that have "grown organically" over the years.

In all those cases, new categories formed by intersections of the primary categories can become small relative to the number of items that might be needed from them. It is of course also possible that primary categories are small to begin with; any conclusions to be made later in this paper then apply directly to these sets. As stated by Stocking and Swanson (1992, p 11): "Given the number of intrinsic item features that may be of interest to test specialists, the number of mutually exclusive partitions can be very large and the number of items in each partition can become quite small. For example, consider items that can be classified with respect to only 10 different item properties, each property having only two levels. The number of mutually exclusive partitions of such items is $(2^{10} - 1)$ or over 1000 partitions. Even with a large item pool, the number of items in each mutually exclusive partition can become quite small." For tests that require stimulus material the limiting factor can occur on a different level. For example, in the case of a reading comprehension test where the test specification may refer to reading passages. Using part of an example of test construction in Stocking and Swanson (1992, p 21), one reads the following passage: "The first ten constraints are relevant to the content of reading passages. For example, a passage may be classified as (long or medium), as having content from the fields of (science, humanities, social sciences), as being (argumentative, narrative), containing references to (females, males) and references to (minorities). The next 11 constraints are relevant to the items associated with the reading passages. These items may ask about the main idea of a passage, an explicit statement, or require inference, etc." If part of a (repeated) test specification requires a short reading passage about science in narrative style with reference to females but not to minorities, together with items that require inference, the possibities of test construction will depend on how the bank is stocked with such matters. The following example of a partial enumeration of primary categories and categories could easily occur in practice.

number of items in bank (primary categories)	600
items from domain X	300
knowledge items	175
difficult items (categories)	260
domain X and knowledge items	25
domain X and difficult items	90
difficult and knowledge items	50
domain X, knowledge and difficult items	15

Although the above primary categories each cover a substantial number of items, the results of intersection are categories of diverging magnitudes. Small categories can quickly exhaust the possibilities of test construction, even given a large tolerated overlap between tests. The following example will be used to demonstrate the possible numerical effect of a small category on test construction.

Suppose that because of test specification, intersection of primary categories results in one small category. Assume as given that this category contains 16 items and that the test specification requires 7 items from that category. A further requirement is that in case several tests have to be constructed, the overlap between any two tests should be at most 2 items. This last demand could also be formulated as follows: if we regard the category as a binary vector where after item selection the chosen items are represented by 1 and the others by 0, then identical to the overlap demand is the statement that the Hamming distance between two vectors representing the tests should be at least 10. (The Hamming distance between two binary vectors of equal length is the aggregated element-wise sum (mod 2) of these vectors. In other words, to find the Hamming distance, count the number of positions where the two vectors have different elements.)

From here we use the following terminology and notation: n is the length of a vector (i.e., the number of items in a category); w is the required weight of a vector which is the number of its 1-elements (i.e., the number of items required from that category) and d is the required minimum Hamming distance between any two vectors as determined by the specification of the maximum overlap. On the first occasion of test construction the number of possibilities is given below by (1). Due to the minimum distance requirement the number of possibilities is reduced on each subsequent occasion and is given by (2) - (4) for the second, third and fourth occasion. In the following derivation the maximally allowed overlap is used on each occasion, in order to create the most favourable situation for test construction.

$$\binom{n}{w}. \tag{1}$$

Inspection of Figure 1 shows that the elements of the maximally allowed overlap, $(w - 0.5d)$, have to be selected from the w elements in the first round; the other elements, $(0.5d)$, have to be picked from the remaining $(n - w)$ elements in the first round. Multiplication of both the number of ways these selections can be made produces (2). In a similar way, (3) and (4) are derived.

$$\binom{w}{w - 0.5d} \times \binom{n - w}{0.5d}, \tag{2}$$

$$\binom{0.5d}{w - 0.5d} \times \binom{0.5d}{w - 0.5d} \times \binom{n - w - 0.5d}{d - w}, \tag{3}$$

$$\binom{d - w}{w - 0.5d} \times \binom{d - w}{w - 0.5d} \times \binom{d - w}{w - 0.5d} \times 1 \qquad \left[1 = \binom{n - 1.5d}{1}\right], \tag{4}$$

using $\binom{n}{r} = \binom{n}{n - r}$ (1) - (4) can be rewritten as (5) - (8).

$$\binom{n}{w}, \tag{5}$$

$$\binom{w}{0.5d} \times \binom{n - w}{0.5d}, \tag{6}$$

$$\binom{0.5d}{d - w} \times \binom{0.5d}{d - w} \times \binom{n - w - 0.5d}{d - w}, \tag{7}$$

$$\binom{d - w}{1.5d - 2w} \times \binom{d - w}{1.5d - 2w} \times \binom{d - w}{1.5d - 2w} \times 1 \left[\binom{n - 1.5d}{1.5d - 2w} = 1\right]. \tag{8}$$

Using the regularity in (5) - (8) in an attempt to make a fifth selection, we see in (9) below that this is impossible, since $(n - 3d + 2w) = 0$.

$$(1.5d - 2w) \times \ldots\ldots (1.5d - 2w) \times (n - 3d + 2w) = 0. \tag{9}$$

Figure 1 presents an example of particular instances of the selection rounds (1) - (4), based on the selection of seven items from 16 items with maximum overlap of two items for different selections.

1.	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
2.	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0
3.	0	0	0	1	1	0	0	0	0	0	1	1	1	1	1	0
4.	0	1	1	0	0	0	0	0	1	1	0	0	0	1	1	1
	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
(5).	1							1					1			1

Figure 1
A Selection of 7 Items from 16 with Maximum Overlap of 2

Next, a specific proof that no more than four selections are possible (the approach is based on a proof of the Johnson bound, to be mentioned later on, as in MacWilliams & Sloane, 1981, p 525). Regard Figure 1 as a 4 by 16 matrix X with elements x_j^i. The sum S of the inner products of the rows of X can be evaluated in two ways, row-wise and column-wise:

$$\sum_{i=1}^{4} \sum_{\substack{j=1 \\ j \neq i}}^{4} \sum_{v=1}^{16} x_{iv}\, x_{jv},$$

and

$$\sum_{v=1}^{16} \sum_{i=1}^{4} \sum_{\substack{j=1 \\ j \neq i}}^{4} x_{iv}\, x_{jv}.$$

If $d = 2\delta$, then the following holds for the first evaluation. The specified distance between between any pair of rows of X is at least 2δ and therefore their inner product is at most $w - \delta$. If A is the number of rows of X, then $S \leq (w - \delta) A (A - 1)$, or $S \leq 2.4.3 = 24$.

The following holds for the second evaluation: if k_v is the number of 1's in the vth column of X then its contribution to S is $k_v(k_v - 1)$. If they exist, the summations must be the same. This means that

$$\sum k_v(k_v - 1) = (w - \delta) A (A - 1).$$

Inspection of Figure 1 shows that this leads to 24 = 24. Now assume that a fifth round is possible, so $A = 5$. Evaluation via the first way leads to: $S \leq (w - \delta)5.4$. Inspection of Figure 1 shows that the smallest possible S according to the second evaluation is obtained by placing, in the fifth round, 1's in the columns where there was only one 1 previously. This happens four times. This implies that in three columns there will be three 1's. Altogether this means three columns with three 1's and thirteen columns with two 1's, giving $S = 18 + 26 = 44$ (i.e., 3(3.2) + 13(2.1)). This implies

$$44 = (w - \delta)\, 20 \text{ or } (w - \delta) = 2.2.$$

This is a contradiction, since in this example $(w - \delta) \leq 2$, so no fifth round is posssible without violating the minimum distance requirement. Therefore, $A = 4$. So, if an item bank has a category containing 16 items and the test specification requires 7 items from this category while there should be an overlap of at most 2 items between any two tests (as a whole or for that category), then this can be done exactly four times. The following notation will be used: $A(16, 10, 7) = 4$.

The problem is how to find the numbers $A(n, d, w)$ in general. Fortunately, there is a theory that is greatly concerned with these numbers; it is the theory of error-correcting codes for constant weight codes. Unfortunately, the problem is regarded as being very difficult (see e.g., MacWilliams & Sloane, 1981; El Gamal et al., 1987; Dueck & Scheuer, 1988).

Some Theory on the Numbers *A(n, d, w)*

The theory of error-correcting codes is part of communication and information theory and the purpose of codes is to detect and correct errors on noisy communication channels. The general idea is to add redundancy to a message-string by supplementing or replacing it by a longer string, such that the expected transmission errors are less serious than the added capability. The following example is taken from Blahut (1983). Suppose we only have to transmit sequences of 2-bit binary numbers: these numbers are 00, 01, 10 and 11. Any single-bit transmission error in these numbers is undetectable. But suppose the following arbitrary substitution is made:

$$\begin{aligned} 00 &\rightarrow 10101 \\ 01 &\rightarrow 10010 \\ 10 &\rightarrow 01110 \\ 11 &\rightarrow 11111 \end{aligned}$$

Once the substitution is decided on, receiving a 5-bit codeword implies receiving the corresponding original 2-bit word. If a single-bit transmission error occurs in this situation and a 5-bit codeword is received that is not a member of the substitution set, for example 01100, we search for the original 5-bit codeword that has the smallest Hamming distance to the received codeword and decide this was the original message (01110). Obviously, this is not a particularly good code, since it can only detect very simple error patterns. In practice we try to find codes with many codewords of sufficient word length (depending on the situation), with the codewords as different as possible from each other (specifying minimum Hamming distance).

The nature of this theory need not concern us any further here; important is that part of the problems in this theory is the combinatorial problem of finding $A(n, d, w)$. Each realization of (n, d, w) is a (for our purposes) binary vector of length n and weight w (the number of 1's in the vector) with minimum Hamming distance d to any other vector in the colllection. Each realization is called a codeword and the complete collection is the code. If no other demands are formulated, this is called an unrestrained constant weight code and the problem in the theory of error-correcting codes is to find the largest possible A for (n, d, w). For notational convenience we will use $d = 2\delta$ where necessary. It is obvious that for binary constant weight vectors, d is always even. Why the determination of $A(n, d, w)$ is in general such a difficult problem, MacWilliams and Sloane (1981) mention the following relation:

$$A(n, 2\delta, w) \leq \{n(n-1)\ldots(n-w+\delta)\} / \{w(w-1)\ldots\delta\}, \qquad (10)$$

with equality holding if and only if a Steiner system $S(w-\delta+1, w, n)$ exists. Steiner systems arise in the theory of t-designs (block designs theory). One of the main problems in t-designs is the question of their existence (see, e.g., Constantine, 1987).

A t-design or, more completely, a $t(v, k, \lambda)$ design implies the following: given a set V with v elements then a t-design is a collection of distinct k-subsets of V (or blocks of size k of V) such that any t-subset of V is present in exactly λ blocks (of k-subsets) with $t \leq k < v$. Quoting MacWilliams and Sloane: it is a collection of committees chosen out of v people, each committee having k persons and such that any t persons serve together on exactly λ committees. A Steiner system is a t-design with $\lambda = 1$; such a $t - (v, k, 1)$ design is called a $S(t, k, v)$ system. It is known (Constantine, 1987) that the existence of t-designs for $t > 5$ is problematic; only recently (as of 1988) very few designs for $t = 6$ have been discovered and no design for $t = 7$ is known. In terms of the Steiner system mentioned below (10), this means that for all practical purposes $(w - \delta + 1) \leq 5$, or $w - \delta \leq 4$. If we use c as symbol for the maximally allowed overlap, then $d = 2w - 2c$ and $\delta = w - c$. Substituting we find $c \leq 4$. So, in case of overlap of four items or less, the existence or non-existence of the relevant design can in principle be proven and in the first case $A(n, d, w)$ can be determined.

In a sense, however, this is shifting the problem to a different area. In Constantine (1987) methods of constructing t-designs are presented. The point is, that here too the problem of existence plays a role, or that only specific instances apply. It is not necessary to pursue this matter, since our point was to demonstrate the difficulty of determining $A(n, d, w)$ in general.

Another bound that can be quite useful is the Johnson bound (MacWilliams & Sloane, 1981). It applies when the denominator in the following expression is positive:

$$A(n, 2\delta, w) \leq \left\lfloor \delta n / \left(w^2 - wn + \delta n\right) \right\rfloor, \tag{11}$$

where A is found by taking the integer part of this number. The utility of this bound is rather unpredictable. For example, for $A(10, 4, 3)$ the bound does not apply, since the denominator in (11) becomes negative. Taking the absolute value (in this case 20) is no use and anyhow not allowed; the actual value is $A(10, 4, 3) = 13$. So, in all cases where $w(n - w) > \delta n$ applies, (11) is of no use. Using $2\delta = d = 2w - 2c$ (with c the maximum overlap as above) and substituting gives

$$c < w^2/n,\ n < w^2/c,\ w > \sqrt{(nc)}\ , \tag{12}$$

and the utility of (11) can be checked by means of bounds on the specified overlap (c), bounds on the size of a category (n) or bounds on the number of items needed from that category (w); if (12) is true, then (11) can be used. Even when (11) can be used the bound can become rather loose, as the following example will show. From the literature the following ranges are known: range $A(22, 10, 7) = 15$ to 19, range $A(23, 10, 7) = 19$ to 23 and range $A(24, 10, 7) = 24$ to 27. The Johnson bounds are respectively (rounded to the lower integer) 22, 38 and 120. However, $A(16, 10, 7) = 4$ and this is also the Johnson bound. The impression is, that the bound becomes looser when the denominator in (11) gets smaller relative to n. So, for such cases and when the denominator in (11) is negative, it would be convenient if other methods for finding $A(n, d, w)$ exist. Fortunately, such is the case: in recent years two heuristic approaches to this problem have been developed. Before presenting these, several other bounds are presented which can be useful on occasion, although the Johnson bound is regarded as the best (MacWilliams & Sloane, 1981).

$$A(n, 2\delta, w) \leq (n/w)A(n-1, 2\delta, w-1).$$

This bound can be applied until a known value of $A(n, d, w)$ is reached. Similarly,

$$A(n, 2\delta, w) \leq (n/(n-w)A(n-1, 2\delta, w).$$

It is not possible to say in advance which one of these two will give better results. Furthermore, some highly specific identities are known, depending on very specific combinations of values for d, w and $n = x(\text{mod } y)$. These are too specific to be of any use in an item banking context. One example will be presented.

$$\begin{aligned} &= n(n-1)(n-2)/4.3.2 \text{ if } n \equiv 2 \text{ or } 4 \pmod 6 \\ A(n, 4, 4) &= n(n-1)(n-3)/4.3.2 \text{ if } n \equiv 1 \text{ or } 3 \pmod 6 \\ &= n(n^2-3n-6)/4.3.2 \text{ if } n \equiv 0 \quad \pmod 6. \end{aligned}$$

(These last results are due to Kalbfleisch and Stanton and to Brouwer as mentioned in MacWilliams & Sloane, 1981.)

Heuristic Approaches to *A(n, d, w)*

In solving combinatorial optimization problems we try to find optima of functions of discrete variables. This can be done by either an exact algorithm or a heuristic algorithm. In the first case a global optimum, if it exists, is reached for every instance of the problem; in the second case the algorithm should produce solutions that are close (as defined in some sense) to a global optimum. Note that complete enumeration of the solution space is generally not practical for many optimization problems of realistic size. Proofs of convergence are a desirable property for approximation algorithms. However, the need for heuristic approaches is so great that in the past and present empirical evidence of effectiveness was and is accepted as a second best. The problem is that many combinatorial optimization problems that are of practical interest, are NP-hard (Papadimitriou & Steiglitz, 1982). For this class of problems no algorithm is known that solves any of these problems in a time bounded by a polynomial function of the size (as defined in some way) of the problem. This means that very frequently solving large-scale problems exactly is in general not possible (however, solving for one of these problems means solving for the whole class, since these problems are transformable into each other). All this meant that in the past, for each specific combinatorial optimization problem, a specific approximation algorithm had to be developed (see, e.g., Syslo, Deo & Kowalik, 1983, for examples).

In recent years there have been developments with respect to general approximation algorithms that are much less problem-specific. Some of these developments can be regarded as extensions of local search techniques (see, e.g., Aarts & Van Laarhoven, 1989). Local search involves the definition of configurations (suggested solutions), cost function (to evaluate the solutions) and a generating mechanism (to go from one configuration to another). Disadvantages of local search are the strong dependency on initial configurations and their termination on meeting the first local optimum, which is not necessarely the global optimum. Examples of recent advances that avoid these problems are Simulated Annealing (SA) and a variation on SA, the so-called Treshold Accepting algorithm (TA). Both have been used on the Traveling Salesperson Problem, which is often considered as a benchmark problem, and the problem of finding $A(n, d, w)$; (El Gamal et al., 1987, and Dueck & Scheuer, 1988). The TA algorithm will be used in this chapter to search for $A(n, d, w)$ in practical time. Below, the essentials of both algorithms will be presented in pseudo computer language as in Dueck and Scheuer (1988, p 6). For an explanation and an application of SA in a psychometric context, see DeSarbo et al. (1989). In the following, a configuration is any possible randomly suggested solution to the optimization problem; temperature functions as a control parameter in the same units as the cost function; while its value becomes lower, the chance of the cost 'escaping to a higher energy state' as it where, becomes smaller.

(Comparable to a marble in a three-dimensional landscape of peaks and valleys, where the marble starts at a random peak and has to reach the lowest valley: occasionally the landscape is jolted and the intensity of the jolting decreases in time. The jolting here functions as temperature in SA).

```
SA algorithm
Choose an initial configuration
Choose an initial temperature $T > 0$
Opt:  choose a new configuraion which is a stochastic small perturbation of the old
      configuration
    compute $\Delta E$: = quality (new configuration) - quality (old configuration)
    IF $\Delta E > 0$
      THEN old configuration: = new configuration
      ELSE with probability $\exp(- \Delta E/T)$
             old configuration: = new configuration
    IF  a long time no increase in quality or too many iterations
      THEN lower temperature T
    IF some time no change in quality any more
      THEN stop
GOTO Opt
```

From this brief description it is obvious that at very high temperature levels the algorithm makes a near random walk through the solution space; it is also clear that there is a built-in backtracking facility, since at all levels of the parameter T there is a non-zero probality of accepting cost increasing solutions. For theory on the convergence of SA algorithms, see e.g., Faigle and Kern (1989) and Aarts and Korst (1989). The TA algorithm differs from SA only in the removal of the stochastic character of the acceptance rule for new solutions. It is schematically presented as follows.

```
TA algorithm
choose an initial configuration
choose an initial THRESHOLD T > 0
Opt:  choose a new configuration which is a stochastic small perturbation of the old
         configuration
    compute $\Delta$E: = quality(new configuration) - quality(old configuration)
    IF $\Delta$E > -T
       THEN old configuration: = new configuration
    IF a long time no increase in quality or too many iterations
       THEN lower TRESHOLD T
    IF some time no change in quality anymore
       THEN  stop
GOTO Opt.
```

Dueck and Scheuer (1988) report comparisons of the results of their algorithm with results on some benchmark problems as reported in the literature for both the TSP problem and the $A(n, d, w)$ problem. In the majority of comparisons, TA performs as well or better then SA for both problems.

For the $A(n, d, w)$ problem the TA algorithm is implemented to do the following: after the user specifies a guess as regards the size of A, the algorithm generates as start configuration a collection of A codewords of length n and weight w. The stochastic small perturbation consists in the random transposition of two positions in a randomly selected codeword. The quality of the new configuration is measured by

$$\sum (d - d_{a,b})^2,$$

(summation for codewords a and b for which holds that $d_{a,b} < d$, where d is the specified minimum Hamming distance and $d_{a,b}$ the actual Hamming distance between a and b). The purpose of the algorithm is to mimimize this expression. Results for the $A(23, 10, 7)$ problem were presented by El Gamal et al. (1987) and Dueck and Scheuer (1988) with values of 18 and 17 respectively, this being one of the few examples in Dueck and Scheuer where TA did less well then SA. The SA result of 18 was reported as the best result in the literature at that time. In the first appendix we present a result of A(23, 10, 7) = 19, and the corresponding table of Hamming distances obtained with an implementation of the TA algorithm.

As can be seen in this table, no distance is less then 10, so a lower bound for A(23, 10, 7) is 19. On IBM-compatible machines of types 386 and 486 the TA algorithm is very fast and finds the lower bounds as known in the literature or even better as in the case above. (Actually, further trials with the TA algorithm for A(23, 10, 7) = 20 suggested that the lower bound (and probably upper bound) might be 20; however, no actual solution was found).

The general conclusion is that the TA algorithm is a suitable heuristic for finding values for $A(n, d, w)$ that are near-optimal to optimal in most practical situations in an item banking context.

Further Limits due to Constraints on Item Level

In this paragraph we will show that seemingly innocuous restrictions in the test specification can have far reaching results. For this demonstration we will restrict ourselves to simple exclusions from outside the category under consideration, that is, if one particular (or more) item from a different subset is included in the test then one particular (or more) item from the subset under consideration must be exluded. Obviously, the point here is that not a particular item is excluded but also all sets of which this item might be a member. Simple combinatorial considerations show that the effects can be rather restrictive.

Again, let n be the number of items in the category and w the number of items to be selected from that category, according to the specification; let t be the number of items in that category that are excluded by the activation of one or more restrictions.

If we have a set of n elements then the cardinality of the collection of subsets of size w is $\binom{n}{w}$. If we have other subsets of size ($0 \leq t \leq w$), then these subsets are present in exactly $\binom{n-t}{w-t}$ subsets of cardinality w (see, e.g., Constantine, 1987). If subsets of items of size w that contain t specific items are excluded (because one or more constraints are activated) from test construction, then the relative reduction in the number of possibilities is given by

$$\binom{n-t}{w-t} \Big/ \binom{n}{w}.$$

By increasing the number of such constraints the reducing factor grows very quickly. In order to keep the formula's simple this will be demonstrated for the case of two restrictions that exclude one item each in the category under consideration, so $t_1 = t_2 = 1$. For this specific example the reducing factor becomes

$$\left\{\binom{n-1}{w-1} + \binom{n-1}{w-1} - \binom{n-2}{w-2}\right\} \Big/ \binom{n}{w}, \tag{13}$$

(The third term is subtracted to prevent double counting).
This example shows also (if necessary) that the exclusion of two specific items has different combinatorial effects compared to the exclusion of one specific pair of items. In the latter case the reducing factor is

$$\binom{n-2}{w-2} \Big/ \binom{n}{w}.$$

Why (13) signifies a 'growth' in the reducing factor can better be seen by using the identity $\binom{n}{w} = \binom{n}{n-w}$ and rewriting (13) accordingly as

$$\left\{\binom{n-1}{n-w} + \binom{n-1}{n-w} - \binom{n-2}{n-w}\right\} \Big/ \binom{n}{n-w}. \tag{14}$$

Inspection of (14) shows also that the effect of exclusions (in this specific example of two exclusions of one item) on the reducing factor increases while the difference between n and w grows smaller or, in other words, while the number of items required from that category increases. A simple numerical example for $n = 10$ and w decreasing from 9 will demonstrate the point.

$$\left\{\binom{9}{1} + \binom{9}{1} - \binom{8}{1}\right\} \Big/ \binom{10}{1} = 1.00$$

$$\left\{\binom{9}{2} + \binom{9}{2} - \binom{8}{2}\right\} \Big/ \binom{10}{2} = .98$$

$$\left\{\binom{9}{3} + \binom{9}{3} - \binom{8}{3}\right\} \Big/ \binom{10}{3} = .93$$

$$\left\{\binom{9}{4} + \binom{9}{4} - \binom{8}{4}\right\} \Big/ \binom{10}{4} = .87$$

$$\left\{\binom{9}{5} + \binom{9}{5} - \binom{8}{5}\right\} \Big/ \binom{10}{5} = .78$$

Etcetera.

This example shows that one has to be careful with exclusions of items from small categories.

Conclusions

As mentioned in this paper, the categories from which items are drawn can be small, either because the primary categories used to describe the structure of the item bank are small or, probably more common, because the more transient categories that result from test specification are small. If no overlap between tests is allowed it is easy to see what the possibilities for test construction are and that they will be very limited. If overlap is allowed, it is more difficult to see what the possibilties are but they are at least more numerous, although generally still small in number. Generally speaking, if overlap between tests is allowed at all, it will rarely be more than a small number of items. If that small number is concentrated in one particular small category, because of the combinatorial limitations then, if other small categories are being used, the problem is transfered to those categories because no overlap will be allowed in further categories. Taking the example of $A(23, 10, 7) = 19$; if this uses the maximally allowed overlap and, for example , 4 items from another category have to be used for 19 tests, then this category should contain at least 76 items. If item exclusions could be operative at that moment, this number could be considerably larger.

The following matter is also a potential source of difficulties. Again taking the example of $A(23, 10, 7)$, it is obvious that the contribution of 7 items from a set of 23 to total test information can be realized in $\binom{23}{7} = 245157$ ways; a reduction to 19 ways could mean that elsewhere problems arise as regards this topic.

The conclusions are unpleasant but simple. One should have as many items as is possible under the circumstances. None of the primary descriptive categories should be small. The test specification should be such that no small transient categories arise in the process. If for whatever reason small categories of any kind do arise, one should allow substantial overlap between tests and supply the user with the TA algorithm or an equivalent heuristic, in order to estimate the possibilities for test construction. However, the interpretation of terms such as many, small and substantial depends on actual circumstances and since the subject matter of this paper is usually beyond the capabilities of the average test constructor, it will be the task of the psychometrician to advise in these matters.

Appendix

1.	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0	1	1	1
2.	1	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	1	0	0	0
3.	1	0	1	0	0	0	1	1	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0
4.	1	0	1	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	1
5.	1	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0	0	0	1	1	0	0	0
6.	1	1	0	0	0	0	0	0	0	0	1	1	0	1	0	1	0	0	0	0	0	1	0
7.	0	0	1	0	1	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	1	1
8.	1	1	0	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0
9.	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	1	1	1	1	0
10.	0	1	0	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0	1	0	1	0	0
11.	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1	1	0	1	0	0	1	0	0
12.	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
13.	0	0	1	1	1	0	0	0	0	0	1	1	0	0	0	0	1	0	0	0	1	0	0
14.	0	0	0	1	0	0	1	1	0	1	1	0	0	0	0	0	0	0	0	1	0	1	0
15.	0	1	0	0	0	0	0	0	1	1	0	1	0	0	1	0	1	0	1	0	0	0	0
16.	0	0	1	0	0	0	0	1	1	0	0	0	0	1	0	1	0	0	0	1	1	0	0
17.	0	0	0	0	0	1	0	1	0	0	1	0	0	1	1	0	0	0	1	0	0	0	1
18.	0	0	0	1	1	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	1	0
19.	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	1	1	0	0	1	0	0	1

The table of Hamming distances belonging to this solution is presented below. The first row is the distance between the second and the first codeword; the second row are the distances between the third and first, respectively the third and the second codeword, and so on.

10																	
10	10																
10	10	10															
10	10	10	10														
10	10	10	10	10													
10	10	10	10	10	10												
10	12	10	10	10	10	12											
10	10	12	12	10	10	10	10										
10	10	10	10	10	10	10	10	10									
12	10	10	10	10	10	10	10	10	10								
10	10	10	10	10	12	10	10	10	10	10							
10	10	10	10	10	10	10	10	10	10	10	10						
10	10	10	10	10	10	12	10	10	10	10	10	10					
10	10	10	10	10	10	14	10	10	10	10	10	10	12				
10	10	10	10	10	10	10	10	10	10	10	10	10	10	12			
12	10	10	10	10	10	10	10	10	10	10	10	12	10	10	10		
10	10	12	10	10	10	10	10	10	12	10	10	10	10	10	10	10	
10	10	12	10	10	10	10	10	10	10	10	10	12	12	10	10	10	10

References

Aarts, E. & Korst, J. (1989). *Simulated annealing and Boltzmann machines.* Chichester: Wiley.

Aarts, E.H.L. & Van Laarhoven, P.J.M. (1989). Simulated annealing: An introduction. *Statistica Neerlandica, 43*, 31 - 52.

Blahut, R.E. (1983). Theory and practice of error control codes. Massachusetts: Addison-Wesley, Reading.

Constantine, G.M. (1987). *Combinatorial theory and statistical design.* New York: Wiley.

DeSarbo, W.S., Oliver, R.L. & Rangaswamy, A. (1989). A simulated annealing methodology for clusterwise linear regression. *Psychometrika, 54,* 707 - 736.

Dueck, G. & Scheuer, T. (1988). *Treshold Accepting: A general purpose optimization algorithm appearing superior to simulated annealing.* Heidelberg: IBM Scientific Center TR 88.10.011.

El Gamal, A.A., Hemachandra, L.A., Shperling, I. & Wei, V.K. (1987). Using simulated annealing to design good codes. *IEEE Transactions on information theory. IT-33, 116 - 123.*

Faigle, U. & Kern, W. (1989). Note on the convergence of simulated annealing algorithms. *(Memorandum no. 774)*. Faculty of Applied Mathematics: University of Twente.

MacWilliams, F.J. & Sloane, N.J.A. (1981, 3d pr.). *The theory of error-correcting codes.* North-Holland: Amsterdam.

Papadimitriou, C.H. & Steiglitz, K. (1982). *Combinatorial optimization: algorithms and complexity.* New Jersey: Prentice-Hall, Englewood Cliffs.

Stocking, M.L. & Swanson, L. (1992). A method for severely constrained item selection in adaptive testing. *ETS Research Reports 92-37*. New Jersey: Princeton.

Syslo, M.M., Deo, N. & Kowalik, J.S. (1983). *Discrete optimization algorithms.* New Jersey: Prentice-Hall, Englewood Cliffs.