Screening to Locate Interactions

Charles J. Colbourn
Computer Science and Engineering
Arizona State University
Tempe, AZ 85287-8809

April 16, 2006

Abstract

Let $\{F_1, \ldots F_k\}$ be a set of k *factors*. Each factor F_i has a set V_i of v_i allowed *values*. A *covering array* of *strength* t and *type* $(v_1 \cdots v_k)$ having N *tests* is an $N \times k$ array with the property that choosing any t columns (factors) $i_1, \ldots, i_t$, each of the $\prod_{j=1}^{t} v_{i_j}$ possible t-tuples of values for $F_{i_1}, \ldots, F_{i_t}$ appears at least once in a test as the values of the corresponding factors. (In other words, for every t factors, every possible combination of values is tested at least once.) We call such a choice of t factors and values for each a *t-way interaction*.

Covering arrays have been widely used to detect the presence of unexpected interactions among factors; examples of applications include component-based software testing, integrated circuit I/O testing, developmental genetic networks, materials development, and combinatorial drug design. One way to use a covering array in a screening experiment is to run each of the N tests to produce a binary *response vector*; the presence of a '1' in the ℓth position indicates that an unexpected interaction arose in the execution of the ℓth test. A standard use would be for defect detection. Covering arrays can in this way detect the presence of certain unexpected interactions, but may be unable to *locate* them. Indeed many different combinations of interactions can lead to the same response vector, and hence the unexpected interactions involved cannot be deduced.

In this talk, we explore a generalization of covering arrays. A (d, t)-*locating array* is a covering array of strength t so that if there are at most d unexpected t-way interactions, we can uniquely determine from the response vector which interactions arose. This location condition imposes a cover-free property on the array; indeed considering the subsets of tests in which t-way interactions arise produces a d-cover-free family.

We pose some questions on the existence of (d, t)-locating arrays, and generalize a recursive construction to establish the existence of many potentially useful examples.

This is joint work with Dan McClary at ASU.