Submitted to the Annals of Statistics
arXiv: math.PR/0000000

OPTIMUM MIXED LEVEL DETECTING ARRAYS

By Ce Shi Yu Tang* and Jianxing Yin

Soochow University

As a type of search design, a detecting array can be used to generate test suites to identify and detect faults caused by interactions of factors in a component-based system. Recently, the construction and optimality of detecting arrays have been investigated in depth in the case where all the factors are assumed to have the same number of levels. However, for real world applications, it is more desirable to use detecting arrays in which the various factors may have different numbers of levels. This paper gives a general criterion to measure the optimality of a mixed level detecting array in terms of its size. Based on this optimality criterion, the combinatorial characteristics of mixed level detecting arrays of optimum size are investigated. This enables us to construct optimum mixed level detecting arrays with a heuristic optimization algorithm and combinatorial methods. As a result, some existence results for optimum mixed level detecting arrays achieving a lower bound are provided for practical use.

1. Introduction. Testing plays an important role in component-based systems. Due to the complexity of systems, the number of possible tests can be exponentially large. Consider, for example, the manufacturing of a personal computer system. Suppose that a wide variety of choices are available for both hardware and software components including the central unit processor, primary memory, cache memory, interface, operation system, web browser, and email system. For manufacturers, if an exhaustive test suite is applied to test whether there is any component interaction that may cause system failure, the testing burden shall be extremely heavy even when the system is of moderate complexity. Consequently, they are interested in generating test suites that cover the most prevalent interactions. This phenomenon is not unique and similar scenarios exist in other disciplines such as agriculture, manufacturing, networks and so on. For related information,

*Corresponding author

†This work is supported by the NSFC Grants: 11271280, 11271279, 11301342 and the NSF of Jiangsu Province grant: BK2012612. The first author is supported by Shanghai Special Research Fund for Training College's Young Teachers: ZZlx13001. The second author is supported by Qing Lan Project.

AMS 2000 subject classifications: Primary 62K15; secondary 94C12

Keywords and phrases: detecting array, heuristic algorithm, mixed level, optimality, search design

readers are referred to Colbourn et al. (2006), Moura et al. (2003) and the references therein.

For certain component-based systems, it is generally impossible to examine every interaction without conducting an exhaustive testing. A popular and effective strategy is to set a limit on the number of components involved in interactions which may cause a failure. Such an approach not only reduces the testing burden dramatically, but is also supported by many empirical results. Dalal et al. (1999) claimed that a large percentage of the existing faults in a software system can be found only by examining all pairwise interactions, while Kuhn et al. (2004) reported similar results. For this reason, engineers and researchers prefer to employ covering arrays to conduct test suites.

A covering array (CA) is a special type of fractional factorial design. The formal definition of a CA will be given in Section 2. Roughly speaking, a covering array of strength t (index unity) is a matrix, such that for every choice of t columns of the matrix, all possible t-tuples appear within its rows. When applying a CA to testing problems, the columns of the CA are utilized to represent factors affecting a response in which the entries within the columns indicate settings or values for that factor. The rows represent tests to be performed, in which a value for each factor is dictated. Covering arrays have been extensively studied, see Colbourn (2004) for a survey for CAs, and Colbourn et al. (2006), Moura et al. (2003) and Colbourn et al. (2011) for their properties, applications and efficient constructions.

Testing with a CA can determine whether any t-component interactions cause a failure, which is an important step in screening a system for interaction faults. However, as Colbourn and McClary (2008) pointed out, test suites based on covering arrays provide little information about identifying the exact interactions causing faults. Due to practical concerns, tests that reveal the location of interaction faults are of interest. Colbourn and McClary (2008) formalized the problem of nonadaptive location of interaction faults under the hypothesis that the system contains (at most) d faults, each involving (at most) t factors. They proposed the notion of a detecting array, which is a CA with additional requirements.

Under the framework of Colbourn and McClary (2008), detecting arrays have been investigated in depth by a number of authors in the case where all the factors had the same number of levels. A combinatorial characterization and a general criterion of measuring optimality for such detecting arrays were established in Shi et al. (2012) and Tang and Yin (2011). For practical applications, things are not always that simple, and it is desirable to use detecting arrays in which the various factors may have different numbers of

levels.

In Brownlie et al. (1992), the authors considered a test on the copy function of the PMx/StarMAIL system, aiming to provide a pairwise covering solution. Four factors, i.e, function scope, server type, client type and target content were considered, each with three levels. The factor "function scope" referred to three copy options: copy only a single (current) message; copy messages previously marked, or copy all messages contained within a folder. However, in practice it is easy to use some shortcut keys to mark all messages in a folder, thus we can only assign two values, i.e, "current" and "marked", to the factor "function scope", as done by many other electronic mail systems. In view of this, the original fixed level test problem becomes a mixed level one. Another mixed level test example can be found in Cohen et al. (1996), where the authors considered a voice response unit test, affected by four factors with three, three, two and two levels.

In the above two examples, the authors both suggested to employ an orthogonal array or a covering array to conduct the corresponding test suites. As a result, interaction faults in these systems can be found, but their locations cannot always be identified. Such drawbacks can be overcome by using a detecting array. Table 1 shows an example of a detecting array for testing the copy function of the PMx/StarMAIL system (here "function scope" is assigned to be a two-level factor). The array contains 18 tests. If we assume that an error in this system can only be due to some combination of two components, then any single fault can be identified according to the outcome of these tests. For example, the last column of Table 1 lists a possible outcome, which indicates the system error must be caused by copying the current message by an MSNET client. Moreover, if there are more than one combination of two components causing the faults, this can also be detected.

If properly coded, a detecting array is actually a search design without noise (Srivastava, 1975). Let S be the set of all possible combinations of certain components causing faults (for the detecting array in Table 1, S is the set of all $3\times 9+3\times 6=45$ level combinations with two components), Y represent the outcome, i.e, a 0-1 vector (zero means failure, while one means pass), and M be the incidence matrix of the array (denoted by A) related to S, i.e, if a level combination in S occurs in a row of A, the corresponding entry takes the value one, otherwise it takes the value zero (in the above example, M shall be an 18 by 45 zero-one matrix). If A is a detecting array, then the solution for β in equation $M\beta = Y$ is unique over a predefined space for any possible outcome Y (for the detecting array in Table 1, the solution space for β only contains 45 unit vectors with one element 1 and others 0). Notice here the special coding of M and the restricted space of

TABLE 1
A detecting array for the copy function of the PMx/StarMAIL system

test	Function Scope	Server Type	Client Type	Target Content	Outcome
1	Current	Share Mode	MSNET	Empty	F
2	Current	User Mode	MSNET	Partial	F
3	Marked	Share Mode	Enhanced	Partial	P
4	Marked	User Mode	Basic	Partial	P
5	Marked	MSNET	MSNET	Partial	P
6	Marked	MSNET	Basic	Full	P
7	Current	Share Mode	Enhanced	Full	P
8	Marked	User Mode	MSNET	Empty	P
9	Current	User Mode	Enhanced	Empty	P
10	Marked	MSNET	Enhanced	Empty	P
11	Current	Share Mode	Basic	Partial	P
12	Current	MSNET	Basic	Empty	P
13	Marked	Share Mode	MSNET	Full	P
14	Marked	Share Mode	Basic	Empty	P
15	Current	MSNET	Enhanced	Partial	P
16	Current	MSNET	MSNET	Full	F
17	Marked	User Mode	Enhanced	Full	P
18	Current	User Mode	Basic	Full	P

β make it different from the usual search designs discussed in the existing literature, see for example, Shirakura et al. (1996), Ghosh and Burns (2001) and Esmailzade et al. (2011). Under this consideration, the present authors took a new approach to investigate the property and requirement for detecting arrays. In Tang and Yin (2011) and Shi et al. (2012), they obtained many interesting results for fixed level detecting arrays. Although the basic concepts and notations related to fixed level detecting arrays are analogous to those of mixed level detecting arrays, the combinatorial characteristics as well as construction methods cannot be directly duplicated due to the complicated structure of mixed level arrays. In fact, optimum mixed level detecting arrays are not always super-simple (see Definition 2.1), but have the "extendible" property, which will be defined in Section 3. Based on this special property, we will further provide a heuristic algorithm and some combinatorial methods to construct optimum mixed level arrays with specific parameters.

The organization of this paper is as follows. In section 2, we give a detailed description on the definition of a mixed level detecting array. In section 3, we establish a general criterion for measuring the optimality of a mixed level detecting array in terms of its size, and then describe the combinatorial characteristics of mixed level detecting arrays of optimum size. Based on properties in Section 3, a heuristic optimization algorithm to generate mixed

level detecting arrays is developed in Section 4 and some optimum detecting arrays with the largest number of factors are then found. Composition methods for constructing optimum detecting arrays based on combinatorial methods are included in Section 5. This provides a number of infinite classes of optimum mixed level detecting arrays. Section 6 concludes, while all proofs are deferred to the Appendix.

2. Definitions and Terminology. Let $v_1, v_2, \cdots, v_k$ be k natural numbers (not necessarily distinct). For each i with $1 \leq i \leq k$, let V_i be a set of cardinality v_i. For given natural numbers N and $t \leq k$, a *mixed covering array* (MCA) of type $(v_1, v_2, \cdots, v_k)$, size N, strength t and index λ, denoted by $\mathrm{MCA}_\lambda(N; t, k, (v_1, v_2, \cdots, v_k))$, is an $N \times k$ array A, which satisfies the following two properties:

1. For each i with $1 \leq i \leq k$, the entries in the i-th column of A are taken from V_i.
2. The rows of each $N \times t$ sub-array of A cover all t-tuples of values from the t columns at least λ times.

The coordinates of $V_1 \times V_2 \times \cdots \times V_k$ are referred to as *factors*, so the elements of V_i represent the *levels* of factor i for $1 \leq i \leq k$ and the parameter k is the number of factors, also called the *degree*. The mixed covering array number $\mathrm{MCAN}_\lambda(t, k, (v_1, v_2, \cdots, v_k))$ is the minimum N required to produce an $\mathrm{MCA}_\lambda(N; t, k, (v_1, v_2, \cdots, v_k))$. A mixed covering array is termed *optimum* if its size N=$\mathrm{MCAN}_\lambda(t, k, (v_1, v_2, \cdots, v_k))$. The term "mixed" in the definition is used to indicate that the k level numbers v_i $(1 \leq i \leq k)$ may take different values. If the MCA has a fixed number of levels, it is often known as a *covering array*. In this case, the notations $\mathrm{CA}_\lambda(N; t, k, v)$ and $\mathrm{CAN}_\lambda(t, k, v)$ are employed. In the literature, the subscript is often dropped from the above notations whenever $\lambda = 1$. A $\mathrm{CA}_\lambda(N; t, k, v)$ of size $N = \lambda v^t$ is known as an *orthogonal array* (OA), or an $\mathrm{OA}_\lambda(t, k, v)$, if the rows of every $N \times t$ sub-array cover all t-tuples of symbols exactly λ times. Clearly, an $\mathrm{OA}_\lambda(t, k, v)$ is an optimum $\mathrm{CA}_\lambda(N; t, k, v)$ of size $N = \lambda v^t$.

Notice that the property of an $\mathrm{MCA}_\lambda(N; t, k, (v_1, v_2, \cdots, v_k))$ is preserved if its columns are permuted. So, the type $(v_1, v_2, \cdots, v_k)$ of an MCA is essentially an unordered multiset and one can assume that the sizes of the k level sets are in a non-decreasing order, that is, $v_1 \leq v_2 \leq \cdots \leq v_k$. It follows that an $\mathrm{MCA}_\lambda(N; t, k, (v_1, v_2, \cdots, v_k))$ can exist only if $N \geq \lambda \prod_{i=k-t+1}^{k} v_i$. It is optimum when $N = \lambda \prod_{i=k-t+1}^{k} v_i$. For convenience, a shorthand notation is adopted to describe the type $(v_1, v_2, \cdots, v_k)$ by combining v_i's that are the same. For example, if three v_i's are of size two, one writes this as 2^3

We use I_k to denote the set of the first k natural numbers. In order to describe the definition of a detecting array, let us formulate the testing problem formally. Consider a system with k factors (parameters or components). One may identify the set of values (levels) of the j-th factor ($1 \leq j \leq k$) with $\mathbf{Z}_{v_j} = \{0, 1, \cdots, v_j - 1\}$, the residue class ring of integers modulo v_j. Suppose that $v_1 \leq v_2 \leq \cdots \leq v_k$. A test is an assignment of values to factors, i.e., a k-tuple from $\mathbf{Z}_{v_1} \times \mathbf{Z}_{v_2} \cdots \times \mathbf{Z}_{v_k}$. The execution of a test can have two outcomes: *pass* or *fail*. A t-way interaction or an interaction of strength t is a set of the form $\{(j_i, \sigma_{j_i}) | 1 \leq i \leq t\}$, where $\{j_1, j_2, \ldots, j_t\} \subseteq I_k$ with $j_1 < j_2 < \ldots < j_t$ and $\sigma_{j_i} \in \mathbf{Z}_{v_{j_i}}$. A test (or a k-tuple) $R = (x_1, \cdots, x_k) \in \mathbf{Z}_{v_1} \times \mathbf{Z}_{v_2} \times \cdots \times \mathbf{Z}_{v_k}$ covers the interaction $\{(j_i, \sigma_{j_i}) | 1 \leq i \leq t\}$ if $x_{j_i} = \sigma_{j_i}$ for $1 \leq i \leq t$. Thus, a test with k factors covers exactly $\binom{k}{t}$ interactions of strength t. A test suite is a collection of tests; the outcomes are the corresponding set of pass/fail results. We assume that failures are caused by faulty interactions. A test fails when it contains at least one of the faulty interactions, and does not fail otherwise. It then turns out that a test suite is essentially an $N \times k$ array A of type $(v_1, v_2, \cdots, v_k)$ whose rows consist of N elements (possibly with multiplicities) of the Cartesian product $\mathbf{Z}_{v_1} \times \mathbf{Z}_{v_2} \times \cdots \times \mathbf{Z}_{v_k}$. In order to observe an interaction fault, it is necessary that the interaction is covered by at least one test in the test suite. This is to say that the array A is required to be an $\mathrm{MCA}_\lambda(N; t, k, (v_1, v_2, \cdots, v_k))$.

Now suppose that $A = (a_{ij})$ $(i \in I_N, j \in I_k)$ is an $\mathrm{MCA}_\lambda(N; t, k, (v_1, v_2, \cdots, v_k))$ whose entries on the j-th column are taken from $\mathbf{Z}_{v_j}$ for $1 \leq j \leq k$. As we have noted before, t-way interactions that cause faults can not be located by testing with A as suite. Thus, in order to determine and identify the faults, the covering array A must possess more structure than stipulated in its definition. To this end, we define $\rho(A, T)$ to be the set of rows of A in each of which, an arbitrary t-way interaction T is covered. To be more precise, for any t-way interaction $T = \{(j_i, \sigma_{j_i}) | 1 \leq i \leq t\}$, define

$$\rho(A, T) = \{r | a_{rj_i} = \sigma_{j_i}, 1 \leq i \leq t\},$$

where a_{rj_i} is the (r, j_i) entry of A. For a set of interactions $\mathcal{T}$, define

$$\rho(A, \mathcal{T}) = \bigcup_{T \in \mathcal{T}} \rho(A, T).$$

Following Colbourn and McClary (2008), an $\mathrm{MCA}_\lambda(N; t, k, (v_1, v_2, \cdots, v_k))$ A is termed a detecting array (DTA), or a (d, t)-DTA$(N; k, (v_1, v_2, \cdots, v_k))$, if

$$\rho(A, T) \subseteq \rho(A, \mathcal{T}) \Leftrightarrow T \in \mathcal{T}, \tag{2.1}$$

whenever T is a t-way interaction and $\mathcal{T}$ is a set of t-way interactions of cardinality d.

It was proved in Colbourn and McClary (2008) that testing with a (d,t)-DTA$(N;k,(v_1,v_2,\cdots,v_k))$ is able to identify any set of d interaction-faults of strength t from the outcomes. Further, if there are more than d t-way interactions causing the faults, this can also be detected.

As with MCAs, the minimum number N of runs for which a (d,t)-DTA$(N;k,(v_1,v_2,\cdots,v_k))$ exists is called *detecting array number*, denoted by (d,t)-DTAN$(k,(v_1,v_2,\cdots,v_k))$. A (d,t)-DTA$(N;k,(v_1,v_2,\cdots,v_k))$ with $N=(d,t)$-DTAN$(k,(v_1,v_2,\cdots,v_k))$ is said to be *optimum*. In the case $v_1=v_2=\cdots=v_k=v$, the notations (d,t)-DTA$(N;k,v)$ and (d,t)-DTAN(k,v) are adopted.

Remark that the detecting array of Table 1 in Section 1 is a $(1,2)$-DTA$(18;4,(2^1 3^3))$ if its levels are mapped to the corresponding residue class rings of integers. It can be readily checked that for any two distinct 2-way interactions T and T', we have $\rho(A,T)\neq\rho(A,T')$. This means that the condition (2.1) is satisfied, since $\rho(A,T)\neq\rho(A,T')\Longrightarrow T\neq T'$. The detecting array is of strength $t=2$. All its 2-way interactions among columns are covered at least twice. So it is actually an MCA$_2(18;2,4,(2^1 3^3))$. Furthermore, the rows of any $t+1=3$ columns cover every triple of values from the 3 columns at most once. This is a feature of the underlying MCA of a DTA. To characterize this feature we introduce the following concept, which will be frequently used in the subsequent sections.

DEFINITION 2.1. An MCA$_\lambda(N;t,k,(v_1,v_2,\cdots,v_k))$ is said to be *super-simple*, if each of its $N\times(t+1)$ sub-array covers every $(t+1)$-tuple of values from the $t+1$ columns at most once.

3. Optimality Criterion and Combinatorial Feature. This section is devoted to establishing a general criterion for measuring the optimality of DTAs, and giving the combinatorial characterization of mixed level DTAs with optimum size. Suppose that A is an arbitrary (d,t)-DTA$(N;k,(v_1,v_2,\cdots v_k))$ with $v_1\leq v_2\leq\cdots\leq v_k$. We shall exclude the following three trivial cases. The first one is $t=k$. In this case, an optimum DTA is just the full factorial design. The second case is that of $v_1=1$ for which the DTA is determined uniquely by its subarray, the one with the first factor removed. The last case is $d\geq v_1$. In this case, Colbourn and McClary (2008) showed that a (d,t)-DTA$(N;k,(v_1,v_2,\cdots v_k))$ cannot exist. So, the restrictions

$$2\leq v_1\leq v_2\leq\cdots\leq v_k,\quad t<k\ \text{ and }\ d<v_1$$

are always assumed, whenever we speak of a (d,t)-DTA$(N;k,(v_1,v_2,\cdots v_k))$. In addition, the j-th level-set of cardinality v_j is understood to be $\mathbf{Z}_{v_j}$ for $1 \le j \le k$, unless otherwise stated. Under these conventions, we have the following lemma which is an analogue of Theorem 2.3 in Shi et al. (2012). Hence, we omit its proof for brevity.

LEMMA 3.1. Suppose that A is a (d,t)-DTA$(N;k,(v_1,v_2,\cdots,v_k))$. Then $|\rho(A,T)| \ge d+1$ for any t-way interaction T.

By invoking Lemma 3.1, we are able to establish a lower bound on the function (d,t)-DTAN$(k,(v_1,v_2,\cdots,v_k))$, which serves as a general criterion for measuring the optimality of detecting arrays.

THEOREM 3.2. Let v_j $(1 \le j \le k)$ be k integers with $2 \le v_1 \le v_2 \le \cdots \le v_k$. Then

$$(d,t)\text{-DTAN}(k,(v_1,v_2,\cdots,v_k)) \ge (d+1)\prod_{i=k-t+1}^{k} v_i. \tag{3.1}$$

Taking $v_1 = v_2 = \cdots = v_k = v$ in Theorem 3.2, we retrieve the lower bound on the function (d,t)-DTAN(k,v) given in Shi et al. (2012).

COROLLARY 3.3. Let t,k and v be positive integers with $t < k$. Then

$$(d,t)\text{ -DTAN }(k,v) \ge (d+1)v^t. \tag{3.2}$$

It was proved in Shi et al. (2012) that an optimum (d,t)-DTAN$(N;k,v)$ meeting the lower bound in (3.2) is equivalent to a super-simple OA$_{d+1}(t,k,v)$. However, the structure of an optimum, mixed level DTA is much more complicated than that of an optimum, fixed level DTA even though the lower bounds in (3.1) and (3.2) look similar. The aforementioned equivalence does not hold in the mixed level case, which can be seen in the following example.

EXAMPLE 3.4. An optimum $(2,2)$-DTA$(48;5,(3^3 4^2))$ is formed by taking the transpose of the superimposition of the following two 24×5 arrays:

$$\begin{pmatrix}
1&2&0&0&2&0&2&2&1&0&0&1&1&2&1&2&0&2&0&0&1&0&1&2\\
2&1&2&0&0&2&2&1&0&1&1&2&0&1&1&2&2&2&0&1&0&2&1&2\\
1&0&0&2&1&2&2&1&2&1&1&1&1&2&0&2&0&0&1&0&0&0&0&1\\
2&3&1&2&1&3&3&0&1&1&2&3&0&2&0&1&3&0&2&1&3&2&0&0\\
1&2&1&1&1&0&1&2&0&2&3&2&3&0&1&2&3&3&0&0&1&2&0&0
\end{pmatrix},$$

$$\begin{pmatrix}
0&0&2&1&2&2&1&2&1&1&2&0&1&2&2&0&1&1&1&0&0&0&2&1\\
1&0&0&2&2&1&1&0&0&1&1&0&2&0&2&1&0&1&2&0&0&2&0&1\\
1&2&2&0&2&0&2&1&0&1&0&2&2&1&1&2&0&2&1&2&0&1&0&2\\
3&0&0&2&2&1&2&2&1&3&2&3&0&3&1&0&2&3&1&1&0&0&3&1\\
1&0&1&0&3&3&2&2&2&0&1&2&2&3&0&3&3&3&3&3&2&1&0&1
\end{pmatrix}.$$

It is readily checked that this detecting array is not super-simple. □

Since an optimum, mixed level DTA may not be super-simple, we define more set-theoretic configurations to explore the combinatorial features of an optimum, mixed level DTA.

DEFINITION 3.5. Let $T_t = \{(j_i, \sigma_{j_i}) | 1 \le i \le t\}$ be a t-way interaction. Then each $(t+1)$-way interaction $T_{t+1} = ((j_1, \sigma_{j_1}), \ldots, (j_t, \sigma_{j_t}), (j_{t+1}, \sigma_{j_{t+1}}))$ is said to be an extension of T_t, where $j_{t+1} \in I_k \setminus \{j_1, j_2, \ldots, j_t\}$.

The following lemma follows Definition 3.5 immediately.

LEMMA 3.6. Suppose that A is an $N \times k$ array of type $(v_1, v_2, \cdots, v_k)$. Let T_t be a t-way interaction and T_{t+1} an extension of T_t. Then $\rho(A, T_{t+1}) \subseteq \rho(A, T_t)$.

Lemma 3.6 motivates us to introduce the following concept.

DEFINITION 3.7. Let A be an $N \times k$ array of type $(v_1, v_2, \cdots, v_k)$. If for any t-way interaction T_t of A and any d extension $T_{t+1,i}$ $(1 \le i \le d)$ of T_t, we have $\rho(A, T_t) \setminus \bigcup_{i=1}^{d} \rho(A, T_{t+1,i}) \neq \emptyset$, then the array is said to be *d-extendible*.

It is interesting to note that in the particular case where the array A is an $\mathrm{OA}_{d+1}(t, k, v)$, the terms "$d$-extendible" and "super-simple" are essentially the same. Specifically, we have the following lemma.

LEMMA 3.8. Let A be an $\mathrm{OA}_{d+1}(t, k, v)$. Then A is d-extendible if and only if A is super-simple.

Now we are in a position to characterize the combinatorial features of an optimum (d, t)-DTA$(N; k, (v_1, v_2, \cdots, v_k))$.

THEOREM 3.9. Suppose that A is an $N \times k$ array of type $(v_1, v_2, \cdots, v_k)$. Then A is a (d,t)-DTA$(N; k, (v_1, v_2, \cdots, v_k))$ if and only if A is a d-extendible MCA$_{d+1}(N; t, k, (v_1, v_2, \cdots, v_k))$.

As an immediate consequence of Theorem 3.9, we have the following result.

THEOREM 3.10. An optimum (d,t)-DTA$(N; k, (v_1, v_2, \ldots, v_k))$ meeting the lower bound in (3.1) is equivalent to a d-extendible MCA$_{d+1}(N; t, k, (v_1, v_2, \cdots, v_k))$ of optimum size $N = (d+1)\prod_{i=k-t+1}^{k} v_i$.

Combining Lemma 3.8 with Theorem 3.10 gives us the following useful corollary, which was first stated in Shi et al. (2012).

COROLLARY 3.11. An optimum (d,t)-DTA$((d+1)v^t; k, v)$ meeting the lower bound in (3.2) is equivalent to a super-simple OA$_{d+1}(t, k, v)$.

The following two theorems take care of the cases $k = t+1$ and the type $(v_1, v_2, \cdots, v_{t+1}) = (a^1 b^{k-1})$, respectively.

THEOREM 3.12. A (d,t)-DTA$(N; t+1, (v_1, v_2, \ldots, v_{t+1}))$ of optimum size N meeting the lower bound in (3.1) is equivalent to a super-simple MCA$_{d+1}(N;$
$t, t+1, (v_1, v_2, \cdots, v_{t+1}))$ of optimum size $N = (d+1)\prod_{i=2}^{t+1} v_i$.

THEOREM 3.13. Let a and b be positive integers satisfying $a \le b$. Then, an optimum $(\lambda-1, t)$-DTA$(N; k, (a^1 b^{k-1}))$ meeting the lower bound in (3.1) is equivalent to a super-simple MCA$_{\lambda}(N; t, k, (a^1 b^{k-1}))$ of optimum size $N = \lambda b^t$.

Finally, we give a sufficient condition for the existence of an optimum detecting array, meeting the lower bound in (3.1).

THEOREM 3.14. If a super-simple MCA$_{d+1}(N; t, k, (v_1, v_2, \cdots, v_k))$ of optimum size $N = (d+1)\prod_{i=k-t+1}^{k} v_i$ exists, then an optimum (d,t)-DTA$(N; k,$ $(v_1, v_2, \cdots, v_k))$ meeting the lower bound in (3.1) also exists.

4. A Heuristic Algorithm. In this section, we present an algorithm to search for optimum detecting arrays meeting the lower bound in (3.1) with a small number of runs. The basic idea is to use a heuristic optimization algorithm, which is based on the combinatorial features described in the previous section. We first provide two necessary conditions for the existence of an optimum detecting array, meeting the lower bound in (3.1).

LEMMA 4.1. If there exists an optimum $(1,2)$-DTA$(2q^2;k,q)$ meeting the lower bound in (3.1), then $k \leq 2q$.

LEMMA 4.2. Let u, k and w be non-negative integers and $w \geq 3$. If there exists an optimum $(1,2)$-DTA$(N; 1+k+u, 2^u 3^k w^1)$ meeting the lower bound in (3.1), then either u or k is less than 2. Moreover, if $k=0$, $u \leq 3$; if $k=1$, $u \leq 4$; if $u=0$, $k \leq 5$; if $u=1$, $k \leq 3$.

For parameters N, k, $v_1, v_2, \ldots, v_k$ satisfying Lemma 4.1 or 4.2 constraints, we randomly generate an $N \times k$ matrix, denoted by A_0, with each of its j-th column taking values from Z_{v_j}. For any 2-way interaction T and any of its possible extension T_a, record the values of $|\rho(T, A_0)|$ and $|\rho(T_a, A_0)|$. Given that A_0 is a $(1,2)$-detecting array if and only if these two values are not equal, we then define the quantity $\sum_{T,T_a} I_A\left(|\rho(T,A_0)| = |\rho(T_a,A_0)|\right)$ as an objective function to minimize, where $I_A(\cdot)$ represents the indicator function. We use the simulated annealing method to find a final design with the objective function taking the value zero. The pseudo code in Algorithm 1 illustrates the details.

Algorithm 1 Pseudo code for searching $(1,2)$-detecting arrays.

```
Randomly generate A_0 and calculate Δ(A_0) = Σ_{T,T_a} I_A(|ρ(T,A_0)| = |ρ(T_a,A_0)|)
while Δ(A_0) > 0 do
  Randomly exchange two entries in the same column of A_0 to get A_new
  Compute ∇ = Δ(A_new) − Δ(A_0)
  if ∇ < 0 then
    Set A_0 = A_new
  else
    Randomly generate u from uniform distribution U[0,1]
    if u < p = 0.01 then
      Set A_0 = A_new
    end if
  end if
end while
```

We list our search results for $N \leq 30$ in Table 2. All the obtained D-TAs are of optimum size meeting the bound in (3.1). These DTAs are not only useful in real world applications but also as ingredients in the construction of new optimum DTAs via combinatorial approaches developed in the next section. The found $(1,2)$-DTA$(18;4,(2^1 3^3))$ indicates the optimality of the test scheme presented in Section 1 regarding the copy function of P-Mx/StarMAIL system. Meanwhile, the $(1,2)$-DTA$(18;4,(2^2 3^2))$ in Table 2 gives an optimum solution to the voice response unit test problem (without additional restrictions) studied by Cohen et al. (1996).

We note that an exhaustive computer search shows that neither a $(1,2)$-DTA$(18;5,(2^1 3^4))$ nor a $(1,2)$-DTA$(24;5,(2^1 3^3 4^1))$ can exist, although their parameters are compatible with Lemma 4.2 constraints.

TABLE 2
optimum $(1,2)$*-DTA*$(N;k,(v_1,v_2,\cdots,v_k))$ *for* $N \le 30$

N	8	12	16	18	18	18	20	24
Levels	2^a	$2^a 3^1$	$2^a 4^1$	$2^1 3^a$	$2^a 3^2$	3^a	$2^a 5^1$	$2^a 6^1$
Existence	$a \le 4$	$a \le 3$	$a \le 3$	$a \le 3$	$a \le 4$	$a \le 6$	$a \le 3$	$a \le 3$
N	24	24	24	28	30	30	30	
Levels	$2^a 3^2 4^1$	$2^a 3^1 4^1$	$3^a 4^1$	$2^a 7^1$	$2^a 3^3 5^1$	$2^a 3^1 5^1$	$3^a 5^1$	
Existence	$a \le 1$	$a \le 4$	$a \le 5$	$a \le 3$	$a \le 1$	$a \le 4$	$a \le 5$	

5. Combinatorial Approaches. In real world applications, some testing problems may have factors with a large number of levels. For example, suppose a program consists of several functions or subroutines. Each function needs an input parameter and each parameter may take hundreds of values. This causes the number of factor level combinations to be very large. The objective of this section is to present some combinatorial approaches to find optimum, mixed level detecting arrays with a large number of levels per factor for practical applications.

5.1. *New optimum DTAs from the Kronecker Product.* For given two matrices $A = (a_{ij})$ $(i \in I_m, j \in I_k)$ and $B = (b_{rs})$ $(r \in I_n, s \in I_k)$, the Kronecker Product of A and B is defind to be the $(mn) \times k$ matrix

$$A \otimes B = \begin{pmatrix} (a_{11} & a_{12} & \cdots & a_{1k}) \otimes B \\ (a_{21} & a_{22} & \cdots & a_{2k}) \otimes B \\ \vdots & \vdots & \vdots & \vdots \\ (a_{m1} & a_{m2} & \cdots & a_{mk}) \otimes B \end{pmatrix},$$

where, for any i with $1 \le i \le m$, $(a_{i1}, a_{i2}, \cdots, a_{ik}) \otimes B$ is defined to be the matrix

$$\begin{pmatrix} (a_{i1},b_{11}) & (a_{i2},b_{12}) & \cdots & (a_{ik},b_{1k}) \\ (a_{i1},b_{21}) & (a_{i2},b_{22}) & \cdots & (a_{ik},b_{2k}) \\ \vdots & \vdots & \vdots & \vdots \\ (a_{i1},b_{n1}) & (a_{i2},b_{n2}) & \cdots & (a_{ik},b_{nk}) \end{pmatrix}.$$

THEOREM 5.1. Suppose that A is a super-simple MCA$_{d_1+1}(N_1;t,k,(v_1, v_2,\cdots,v_k))$ of optimum size $N_1 = (d_1+1)\prod_{i=k-t+1}^{k} v_i$ and B is a super-simple MCA$_{d_2+1}(N_2;t,k,(u_1,u_2,\cdots,u_k))$ of optimum size $N_2 = (d_2+1)\prod_{i=k-t+1}^{k} u_i$.

Then $A \otimes B$ is a (d,t)-DTA$(N_1N_2; k, (v_1u_1, v_2u_2, \cdots, v_ku_k))$ of optimum size N_1N_2 attaining the lower bound in (3.1), where $d = (d_1+1)(d_2+1)-1$.

COROLLARY 5.2. Suppose that there exists a super-simple MCA$_{d+1}(N; t, k, (v_1, v_2, \cdots, v_k))$ of optimum size $N = (d+1)\prod_{i=k-t+1}^{k} v_i$. Then

(1) there exists a $(\lambda(d+1)-1, t)$-DTA$(\lambda m^t N; k, (mv_1, mv_2, \cdots, mv_k))$ of optimum size $\lambda m^t N$ meeting the lower bound in (3.1) for any integer λ with $2 \le \lambda \le m$, provided that an OA$(t+1, k+1, m)$ exists;
(2) there exists a (d,t)-DTA$(m^t N; k, (mv_1, mv_2, \cdots, mv_k))$ of optimum size $m^t N$ meeting the lower bound in (3.1), provided that an OA(t, k, m) exists.

Theorem 5.1 and Corollary 5.2 provide the basic idea for constructing new optimum detecting arrays meeting the lower bound in (3.1) from the old ones. Combining with the existence results for the ingredient arrays, we obtain the following theorem.

THEOREM 5.3. There exists

(1) a $(2^m-1, 2)$-DTA$(18^m; 4, (2^m)^1(3^m)^3)$ achieving the lower bound in (3.1) for any integer $m \ge 2$;
(2) a $(2\lambda-1, 2)$-DTA$(18\lambda v^2; 4, (2v)^1(3v)^3)$ achieving the lower bound in (3.1) for any integer $\lambda \le v$, where $v \ge 4$ and $v \not\equiv 2 \pmod 4$ and
(3) a $(1, 2)$-DTA$(18v^2; 4, (2v)^1(3v)^3)$ achieving the lower bound in (3.1) for any integer $v \ne 2, 4, 6$.

5.2. *Existence of optimum DTAs with $k = t+1$.* We begin with the following known results.

LEMMA 5.4. Suppose that $2 \le v_1 \le v_2 \le v_3 \le v_4$. Then there exists:

(1) an MCA$(N; 2, 3, (v_1, v_2, v_3))$ of optimum size $N = v_2v_3$ (Moura et al., 2003);
(2) an MCA$(N; 3, 4, (v_1, v_2, v_3, v_4))$ of optimum size $N = v_2v_3v_4$ (Colbourn et al., 2011).

Taking advantage of some composition constructions, we can completely determine the existence spectrum for a (d,t)-DTA$(N; t+1, (v_1, v_2, \cdots, v_{t+1}))$ of optimum size N meeting the lower bound in (3.1).

LEMMA 5.5. Suppose that an MCA$(N; t, k, (v_1, v_2, \cdots, v_i, \cdots v_k))$ and an MCA$(N'; t-1, k-1, (v_1, v_2, \cdots, v_{i-1}, v_{i+1}, \cdots, v_k))$ all exist, where $1 \le$

$i \le k$. Then an MCA$(N'', t, k, (v_1, v_2, \cdots, v_{i-1}, (v_i + e), v_{i+1}, \cdots, v_k))$ exists for any positive integer e, where $N'' = N + eN'$.

LEMMA 5.6. Suppose that $2 \le v_1 \le v_2 \le \cdots \le v_{t+1}$. Then an optimum MCA$(N; t, t+1, (v_1, v_2, \cdots, v_{t+1}))$ with $N = \prod_{i=2}^{t+1} v_i$ exists for any integer $t \ge 2$.

THEOREM 5.7. Suppose that $2 \le v_1 \le v_2 \le \cdots \le v_{t+1}$. Then an optimum $(d-1, t)$-DTA$(N; t+1, (v_1, v_2, \cdots, v_{t+1}))$ with $N = \prod_{i=2}^{t+1} v_i$ exists if and only if $d \le v_1$.

6. Conclusion. Detecting arrays are a special class of covering arrays. However, detecting arrays, especially mixed level detecting arrays, have much more complicated structure than that of usual covering arrays. They are introduced to generate test suites that are capable of identifying and determining the faulty interactions between factors in a component-based system. Compared to using covering arrays, this is a more useful approach in an application context. In this paper, the notion of "d-extendible" is proposed and used to explore the combinatorial features of mixed level detecting arrays. By way of equivalent combinatorial configurations, an optimality criterion is found. As a consequence, combinatorial approaches and heuristic algorithms are employed to produce a number of examples and infinite classes of mixed level detecting arrays, which are optimum in size meeting the lower bound in (3.1).

Appendix: Proofs. *Proof of Theorem 3.2.* Let A be a (d, t)-DTA$(N; k, (v_1, v_2, \cdots, v_k))$. From Lemma 3.1, we know that $|\rho(A, T)| \ge d + 1$ for any t-way interaction T. Since there are $v_{k-t+1} \times v_{k-t+2} \times \cdots \times v_k$ different t-way interactions, $\{(r, x_r) : k-t+1 \le r \le k\}$, A must contain $(d+1)\prod_{i=k-t+1}^{k} v_i$ rows, which implies (d, t)-DTAN$(k, (v_1, v_2, \cdots, v_k)) \ge (d+1)\prod_{i=k-t+1}^{k} v_i$. □

Proof of Lemma 3.8. "⇒" If A is not super-simple, then there exists a $(t+1)$-way interaction $T_{t+1,1}$ such that $|\rho(A, T_{t+1,1})| \ge 2$. Select the first t elements of $T_{t+1,1}$ to form a t-way interaction T_t, then $T_{t+1,1}$ is an extension of T_t. Let $\mathcal{T} = \{T | T$ is an extension of T_t with $|\rho(A, T)| \ge 1\}$. Obviously $T_{t+1,1} \in \mathcal{T}$. Notice that $|\rho(A, T_{t+1,1})| \ge 2$ and $|\rho(A, T_t)| = d+1$ as A is an OA$_{d+1}(t, k, v)$, we have $|\mathcal{T}| \le d$. However, $\rho(A, T_t) \backslash \rho(A, \mathcal{T}) = \emptyset$. This is a contradiction to the d-extendible property.

"⇐" Suppose there exist a t-way interaction T_t and its d extensions, $A_{t+1,i}$, $i = 1, \ldots, d$, such that $\rho(A, T_t) \backslash \bigcup_{i=1}^{d} \rho(A, T_{t+1,i}) = \emptyset$. Since A is an

$\mathrm{OA}_{d+1}(t,k,v)$, $|\rho(A,T_t)| = d+1$, there must exist $i \in \{1,2,\ldots,d\}$, such that $|\rho(A,T_{t+1,i})| \geq 2$. This is a contradiction to the super-simple property. □

Proof of Theorem 3.9. "⇐" If A is not a (d,t)-DTA$(N;k,(v_1,v_2,\cdots,v_k))$, then there exist a t-way interaction T and a set of d t-way interactions $\mathcal{T} = \{T_1,T_2,\cdots,T_d\}$ such that $T \notin \mathcal{T}$ and $\rho(A,T) \subseteq \rho(A,\mathcal{T})$. If there exists $T_i \in \mathcal{T}$, such that all first coordinates of elements of T_i and T coincide, then define any extension of T as $T_{t+1,i}$. On the other hand, for each $T_i \in \mathcal{T}$, if there exists an element of T_i, say (j_i,σ_{j_i}), such that the first coordinate of any element of T is not j_i, then add the element (j_i,σ_{j_i}) to T to form an extension of T, denoted by $T_{t+1,i}$. Now for any row $R \in \rho(A,T) \subseteq \rho(A,\mathcal{T})$, if $R \in \rho(A,T_i)$, then $R \in \rho(A,T_{t+1,i})$, which tells $\rho(A,T_t)\backslash\bigcup_{i=1}^{d}\rho(A,T_{t+1,i}) = \emptyset$. Notice that for $i \neq j$, $T_{t+1,i}$ and $T_{t+1,j}$ can be the same $(t+1)$-way interaction.

"⇒" Otherwise, there exist a t-way interaction T and d extensions of T, denoted by $T_{t+1,i}$, $1 \leq i \leq d$, such that $\rho(A,T_t)\backslash\bigcup_{i=1}^{d}\rho(A,T_{t+1,i}) = \emptyset$. Then $\rho(A,T_t) = \bigcup_{i=1}^{d}\rho(A,T_{t+1,i})$, as $\rho(A,T_{t+1,i}) \subseteq \rho(A,T)$ for $1 \leq i \leq d$ by Lemma 3.6. For each extension $T_{t+1,i}$, we can select a t-way interaction $T_{t,i} \neq T$ such that $T_{t+1,i}$ is their common extension. Notice that for $i \neq j$, $T_{t,i}$ and $T_{t,j}$ cannot be the same t-way interaction, otherwise, $T_{t+1,i}$ and $T_{t+1,j}$ must be the same $(t+1)$-way interaction. Now we have $\rho(A,T) = \bigcup_{i=1}^{d}\rho(A,T_{t+1,i}) \subseteq \bigcup_{i=1}^{d}\rho(A,T_{t,i})$. This is a contradiction to the definition of a (d,t)-detecting array. □

Proof of Theorem 3.12. "⇒" If A is an optimum (d,t)-DTA$(N;t+1,(v_1,v_2,\cdots,v_{t+1}))$, then A is "d-extendible". By Lemma 3.1, $|\rho(A,T)| \geq d+1$ for any t-way interaction T. Thus, A is an optimum $\mathrm{MCA}_{d+1}(N;t,t+1,(v_1,v_2,\cdots,v_{t+1}))$. In the following, we prove that A is super-simple. Otherwise, suppose that there are two identical row vectors, denoted by $R_1 = (x_1,x_2,\cdots,x_{t+1})$. Let $T = (x_2,x_3,\cdots,x_{t+1})$. T occurs at least $d+1$ times as the subarray indexed by the columns $\{2,3,\cdots,t+1\}$, and hence exactly $d+1$ times (since A contains precisely $(d+1)\prod_{i=2}^{t+1}v_i$ rows). Write $T_{t+1,i}$ $(i = 1,2,3,\cdots,d)$ for the d extensions of T, where $T_{t+1,1} = R_1$. Then $\rho(A,T)\backslash\bigcup_{i=1}^{d}\rho(A,T_{t+1,i}) = \emptyset$, a contradiction to A being "d-extendible".

"⇐" Let B be a super-simple $\mathrm{MCA}_{d+1}(N;t,t+1,(v_1,v_2,\cdots,v_{t+1}))$ with $N = (d+1)\prod_{i=2}^{t+1}v_i$. We prove that B is a d-extendible $\mathrm{MCA}_{d+1}(N;t,k,(v_1,v_2,\cdots,v_k))$. If not, there exists a t-way interaction T and d extensions $T_{t+1,i}$ $(i = 1,2,3,\cdots,d)$ for T such that $\rho(A,T)\backslash\bigcup_{i=1}^{d}\rho(A,T_{t+1,i}) = \emptyset$. Since T occurs at least $d+1$ times, there is one $T_{t+1,i}$ such that $|\rho(A,T_{t+1,i}| \geq 2$. This implies that there is at least one $(t+1)$-tuple of symbols occurring in some $t+1$ columns as rows more than once. This is a contradiction to the

super-simple property of B. The conclusion then follows from Theorem 3.9. □

Proof of Theorem 3.13. The proof can be divided into two cases, one is similar to that of Theorem 3.12, and the other is actually a copy of the proof of Theorem 2.4 in Shi et al. (2012). □

Proof of Theorem 3.14. The proof is similar to the counterpart in Theorem 3.12. □

Proof of Lemma 4.1. Tang and Yin (2011) proved that an optimum $(1, 2)$-DTA$(N; k, q)$ with $N = 2q^2$ is equivalent to a super-simple orthogonal array $\mathrm{OA}_2(2, k, q)$. So for any 2-way interaction T, $|\rho(A, T)| = 2$. Moreover, the pair of rows covering T must be different for each distinct 2-way interaction T. The number of 2-way interactions for a k-factor array is $\binom{k}{2} \times q^2 = k(k-1)q^2/2$. However, there are totally $\binom{N}{2} = q^2(2q^2-1)$ pairs of rows. Thus we have $k(k-1)q^2/2 \leq q^2(2q^2-1)$, which implies $k \leq 2q$ as k and q are both positive integers. □

Proof of Lemma 4.2. (i.) We first prove that either u or k is less than 2. Otherwise, if both u and k are greater than or equal to 2, then we have $N = 6w$. Let A_0 be the sub-array with the last column taking the value 0. Then the number of rows of A_0 is 6. For three-level factors of A_0, there are two rows taking level zero, one and two; for two-level factors of A_0, there are three rows taking level zero and one. Moreover, each column of the factors with the same level of A_0 cannot be a level permutation of another one. Thus, two two-level factors occupy $2 \times 2 \times \binom{3}{2} - 2 = 10$ pairs of rows, on which there is at least one two-level factor taking the same level. Two three-level factors occupy $2 \times 3 = 6$ pairs of rows of A_0, on which there is at least one three-level factor taking the same level. These $10 + 6 = 16$ pairs of rows are mutually different, which is a contradiction to the $\binom{6}{2} = 15$ possible pairs of six rows of A_0.
(ii.) If $k = 0$, then $N = 4w$. Let A_0 be the sub-array with the last column taking the value 0. Then the number of rows of A_0 is 4. For columns of A_0 other than the last one, there are two zeros and two ones. Moreover, each of these zeros or ones in the same column occupies a different pair of rows of A_0, otherwise we can always find a two-way interaction T (which contains the last column as a component) and one of its extension T_a, so that the rows covering T and T_a are identical. Thus we must have $\binom{4}{2} \geq 2u$, i.e, $u \leq 3$.
(iii.) If $k = 1$, then $N = 6w$. Let A_0 be the sub-array with the last column taking the value 0. Then the number of rows of A_0 is 6. Without loss of generality, for the three-level factor of A_0, assume the first two rows take

level zero, the middle two take level one, and the last two take level two. For columns of A_0 other than the last two, the first two, the middle, and the last two rows must take distinct levels, i.e, one takes zero and the other takes one. The number of such possible columns is $2 \times 2 \times 2 = 8$. Finally, among all these u columns, one cannot be a level permutation of another. Thus, $u \leq 4$.
(iv.) If $u = 0$ and $k \geq 1$, then $N = 6w$. Let A_0 be the sub-array with the last column taking the value 0. Then the number of rows of A_0 is 6. For columns of A_0 other than the last one, there are two zeros, two ones and two twos. Moreover, each distinct element in the same column of A_0 occupies a different pair of rows of A_0. Thus we must have $\binom{6}{2} \geq 3k$, i.e, $k \leq 5$.
(v.) If $u = 1$ and $k \geq 1$, then $N = 6w$. Let A_0 be the sub-array with the last column taking the value 0. Then the number of rows of A_0 is 6. The two-level factor occupies $2 \times \binom{3}{2} = 6$ pairs of rows, on which the two-level factor takes the same level. Thus $k \times 3 + 6 \leq \binom{6}{2}$, which gives $k \leq 3$. □

Proof of Theorem 5.1. Write $A = (a_{ij})$ $(i \in I_{N_1}, j \in I_k)$ whose entries on the j-th column are taken from $\mathbf{Z}_{v_j}$ for $1 \leq j \leq k$. Write $B = (b_{ij})$ $(i \in I_{N_2}, j \in I_k)$ whose entries on the j-th column are taken from $\mathbf{Z}_{u_j}$ for $1 \leq j \leq k$. Then, by definition, the Kronecker product C is an $(N_1N_2) \times k$ matrix whose entries on the j-th column are taken from the Cartesian product $\mathbf{Z}_{v_j} \times \mathbf{Z}_{u_j}$ for $1 \leq j \leq k$. Furthermore, each row $(a_{i1}, a_{i2}, \cdots, a_{ik})$ of A generates N_2 rows in the Kronecker product C whose projections on the second component are precisely the N_2 rows of B. So the rows of each $(N_1N_2) \times t$ sub-array of C cover all t-tuples of values from the t columns at least $(d_1 + 1)(d_2 + 1)$ times, as A and B are both MCAs of strength t with indices $d_1 + 1$ and $d_2 + 1$ respectively. Similarly, the super-simple property of A and B guarantees that C is super-simple. This is because the projection on the first component of each row in any $t+1$ columns of C is a row of the corresponding columns of A, while the projection on the second component is a row of the corresponding columns of B. It follows that C is a super-simple $\mathrm{MCA}_{d+1}(N_1N_2; k, (v_1u_1, v_2u_2, \cdots, v_ku_k))$, where the size $N_1N_2 = (d_1 + 1)(d_2 + 1)\prod_{i=k-t+1}^{k}(v_iu_i)$. The conclusion then follows from Theorem **3.12**. □

Proof of Corollary 5.2. From Shi et al. (2012) we know that an $\mathrm{OA}(t+1, k+1, m)$ implies the existence of a super-simple $\mathrm{OA}_\lambda(t, k, m)$ with $2 \leq \lambda \leq m$. So the conclusion (1) holds by taking $u_1 = u_2 = \cdots = u_k = m$ in Theorem 5.1. The conclusion (2) follows directly from Theorem 5.1 by taking $u_1 = u_2 = \cdots = u_k = m$ and $d_2 = 0$. □
Proof of Corollary 5.3. (1) Apply Theorem 5.1 with a super-simple $\mathrm{MCA}_2(18;$

$2, 4, 2^1 3^3$) given in Section 4; (2) Apply the conclusion (1) in corollary 5.2 with a super-simple $\mathrm{MCA}_2(18; 2, 4, 2^1 3^3)$ and an $\mathrm{OA}(3, 5, v)$ given in Ji and Yin (2010); (3) Apply the conclusion (2) in corollary 5.2 with a super-simple $\mathrm{MCA}_2(18; 2, 4, 2^1 3^3)$ and an $\mathrm{OA}(2, 4, v)$ given in Colbourn and Dinitz (2007) and Hedayat et al. (1999). □

Proof of Lemma 5.5. Let A be an $\mathrm{MCA}(N; t, k, (v_1, v_2, \cdots, v_i, \cdots, v_k))$ whose entries on the j-th column are taken from $\mathbf{Z}_{v_j}$ for $1 \le j \le k$. For any particular value i with $1 \le i \le k$, we form an $\mathrm{MCA}(N'; t-1, k, (v_1, v_2, \cdots, v_{i-1}, 1, v_{i+1}, \cdots, v_k))$ by simply inserting one constant column

$$(v_i - 1 + r, v_i - 1 + r, \cdots, v_i - 1 + r)^T$$

in the i-th position of the given $\mathrm{MCA}(N'; t-1, k-1, (v_1, v_2, \cdots, v_{i-1}, v_{i+1}, \cdots, v_k))$. For given positive integer e, we do this for $r = 1, 2, \cdots, e$. This produces e MCAs of index unity whose values on the i-th column are taken from $\{v_i, v_i + 1, \cdots, v_i + e - 1\}$. We then concatenate the obtained e MCAs together with the MCA A to form an $\mathrm{MCA}(N'', t, k, (v_1, v_2, \cdots, v_{i-1}, (v_i + e), v_{i+1}, \cdots, v_k))$ with $N'' = N + eN'$, as desired. □

Proof of Lemma 5.6. The proof is done by mathematical induction on t. For $t = 2, 3$, the conclusion follows from Lemma 5.4. Assume that the conclusion holds when $t = n \ge 3$ and consider the case $t = n + 1$. It is well known that both an $\mathrm{OA}(n+1, n+2, v_1)$ and an $\mathrm{OA}(n, n+1, v_1)$ exist for any given integer v_1 (Hedayat et al., 1999). In essence, they are an optimum $\mathrm{CA}(v_1^{n+1}; n+1, n+2, v_1)$ and an optimum $\mathrm{CA}(v_1^n; n, n+1, v_1)$, respectively. When $v_2 > v_1$, we apply Lemma 5.5 with $i = 2, e = v_2 - v_1, (t, k) = (n+1, n+2)$ and these two MCAs. This creates an $\mathrm{MCA}(N_2; n+1, n+2, (v_1, v_2, v_1, \cdots, v_1))$ of optimum size, denoted by A_2, where $N_2 = v_1^{n+1} + (v_2 - v_1)v_1^n = v_1^n v_2$. In the case $v_1 = v_2$, A_2 can be simply taken to be an $\mathrm{OA}(n+1, n+2, v_1)$. By the induction hypothesis, we have also an $\mathrm{MCA}(v_1^n v_2; n, n+1, (v_1, v_2, \cdots, v_{n+1}))$ denoted by B. So we can again apply Lemma 5.5, as above, with $(t, k) = (n+1, n+2)$, $i = 3, e = v_3 - v_1$ and the MCAs A_2, B to form an $\mathrm{MCA}(N_3; n+1, n+2, (v_1, v_2, v_3, v_1, \cdots, v_1))$ of optimum size $N_3 = v_1^n v_2 + (v_3 - v_1)v_1^{n-1} v_2 = v_1^{n-1} v_2 v_3$, denoted by A_3. This process can be continued recursively until an optimum $\mathrm{MCA}(N_{n+2}; n+1, n+2, (v_1, v_2, \cdots, v_{n+2}))$ is obtained, where $N_{n+2} = \prod_{i=2}^{n+2} v_i$. Consequently, the assertion also holds when $t = n + 1$. This completes the proof. □

Proof of Theorem 5.7. By taking advantage of the equivalence described in Theorem 3.12, we need only to show that there exists a super-simple $\mathrm{MCA}_d(d{\cdot}N; t, t+1, (v_1, v_2, \cdots, v_{t+1}))$ of optimum size $d{\cdot}N = d{\cdot}\prod_{i=2}^{t+1} v_i$. From

Lemma 5.6, we know that an MCA$(N; t, t+1, (v_1, v_2, \cdots, v_{t+1}))$ of optimum size $N = \prod_{i=2}^{t+1} v_i$ exists. Write A for such an MCA. For each i with $0 \le i \le d-1$, we form a new optimum MCA$(N; t, t+1, (v_1, v_2, \cdots, v_{t+1}))$, denoted by A_{σ^i}, by permuting the entries in the first column of A with the permutation σ^i. Write $D = (A_{\sigma^0}^T \cdots A_{\sigma^{d-1}}^T)^T$ for the concatenation of the obtained d MCAs. It is obvious that D is an MCA$_d(dN; t, t+1, (v_1, v_2, \cdots, v_{t+1}))$. It is optimum, since its size $d \cdot N = d \cdot \prod_{i=2}^{t+1} v_i$. It is now left to show that D is super-simple. In fact, otherwise, there would be a $(t+1)$-tuple of values from certain $t+1$ columns of D occurring as rows of these columns at least twice. Without loss of generality, we assume that the $(t+1)$-tuple $\underline{a} = (a_1, a_2, \cdots, a_{t+1})$ of values from the first $t+1$ columns of D occurs twice, as rows of these columns. This implies that there must be i and j with $0 \le i \ne j \le d-1$ so that the $(t+1)$-tuple $\underline{a}$ occurs in the corresponding columns of both A_{σ^i} and A_{σ^j}, since each MCA A_{σ^r} $(0 \le r \le d-1)$ has index $\lambda = 1$. Let b_1 be an arbitrary value from the first column of D. Notice that the MCA A is of strength t and index $\lambda = 1$. By considering the $t+1$-tuple $\underline{b} = (b_1, a_2, \cdots, a_{t+1})$, one can see that $\sigma^i(b_1) = a_1 = \sigma^j(b_1)$ from the construction of A_{σ^i} and A_{σ^j}. This leads $\sigma^i = \sigma^j$. It is a contradiction to the definition of σ (as the order of σ is v_1). □

Acknowledgments. The authors would like to thank the associate editor and two referees for their comments and suggestions, which were greatly beneficial for this paper.

References.

[1] T. Beth, D. Jungnickel, and H. Lenz (1999), Design theory, Cambridge University Press, Cambridge.
[2] J. Brownlie, J. Plowse, and M. Phadke (1992), Robust Testing of AT &T PMXStar-Mail using OATS, *AT&T Technical Journal* 3 (71), 41-47.
[3] K. A. Bush (1952), Orthogonal arrays of index unity, *Ann. Math. Stat.* 23, 426-434.
[4] K. A. Bush (1952), A generalization of the theorem due to MacNeish, *Ann. Math. Stat.* 23, 293-295.
[5] D. M. Cohen, S. R. Dalal, J. Parelius, and G. C. Patton (1996), The combinatorial design approach to automatic test generation, *IEEE Software* 13 (5), 83-88.
[6] M. B. Cohen (2004), Designing test suits for softerware interaction testing, Ph.D. Thesis, University of Auckland.
[7] C. J. Colbourn (2004), Combinatorial aspects of covering arrays, Le Matematiche (Catania) 58, 121-167.
[8] C. J. Colbourn and J. H. Dinitz (2007), The CRC Handbook of Combinatorial Designs, CRC Press, Boca Raton, FL.
[9] C. J. Colbourn, S. S. Martirosyan, G. L. Mullen, D. E. Shasha, G. B. Sherwood, J. L. Yucas (2006), Products of mixed covering arrays of strength two, *J. Combin. Des.* 14, 124-138.

[10] C. J. Colbourn and D. W. McClary (2008), Locating and detecting arrays for interaction faults, *J. Comb. Optim.* 15, 17-48.
[11] C. J. Colbourn, C. Shi, C. Wang and J. Yan (2011), Mixed covering arrays of strength three with few factors, *J. Statist. Plann. Inference* 141, 3640-3647.
[12] S. R. Dalal, A. J. N. Karunanithi, J.M. L. Leaton, G. C. P. Patton and B.M. Horowitz (1999), Model-based testing in practice. In: Proceedings of the international conference on software engineering (ICSE 99), 285-294.
[13] N. Esmailzade, H. Talebi, K. Masahara and M. Jimbo (2011), A new series of main effects plus one plan for 2^m factorial experiments with $m = 4\lambda \pm 1$ and $2m$ runs, *J. Statist. Plann. Inference* 141, 1567-1574.
[14] S. Ghosh and C. Burns (2001), Two general classes of search designs for factor screening experiments with factors at three levels, *Metrika* 54, 1-17.
[15] A. S. Hedayat, N. J. A. Slone and J. Stufken (1999), Orthogonal Arrays, Springer, New York.
[16] L. Ji and J. Yin (2010), Constructions of new orthogonal arrays and covering arrays of strength three, *J. Combin. Theory Ser. A* 117-3, 236-247.
[17] D. R. Kuhn, D. R. Wallace and A. M. Gallo (2004), Software fault interactions and implications for software testing, *IEEE Trans. Softw. Eng.*, 30-6, 418-421.
[18] H. F. MacNeish (1922), Euler squares, *Ann. Math. NY* 23, 221-227.
[19] R. Mandl (1985), Orthogonal latin squares: an application of experiment design to compiler testing, *Commun. ACM* 28-10, 1054-1058.
[20] C. Martinez, L. Moura, D. Panario and B. Stevens (2009), Locating error using ELAs, covering arrays, and adaptive testing algorithms, *SIAM J. Discrete Math.* 23, 1776-1799.
[21] L. Moura, J. Stardom, B. Stevens, A. Williams (2003), Covering arrays with mixed alphabet sizes, *J. Combin. Des.* 11, 413-432.
[22] C. Shi, Y. Tang and J. Yin (2012), The Equivalence between Optimal Detecting Arrays and Super-simple OAs, *Des. Codes Cryptogr.* 62, 131-142.
[23] T. Shirakura, T. Takahashi and J. N. Srivastava (1996), Searching probabilities for nonzero effects in search designs for the noisy case, *Ann. Statist.* 24, 2560-2568.
[24] N. J. Sloane (1993), Covering arrays and intersecting codes, *J. Combin. Des.* 1, 51-63.
[25] J. N. Srivastava (1975), Designs for searching non-negligible effects. In: J. N. Srivastava (ed.) A survey of statistical design and linear models. North-Holland, Amsterdam, 507-519.
[26] Y. Tang and J. Yin (2011), Detecting arrays and their optimality, *Acta Math. Sin. (Engl. Ser.)* 27, 2309-2318.

School of Mathematical Sciences
Soochow university
Suzhou, Jiangsu 215006, China
E-mail: shice060@163.com; ytang@suda.edu.cn; jxyin@suda.edu.cn