Journal of Quality Technology

A Quarterly Journal of Methods, Applications and Related Topics

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/ujqt20

# Scalable level-wise screening experiments using locating arrays

Yasmeen Akhtar, Fan Zhang, Charles J. Colbourn, John Stufken & Violet R. Syrotiuk

**To cite this article:** Yasmeen Akhtar, Fan Zhang, Charles J. Colbourn, John Stufken & Violet R. Syrotiuk (2023): Scalable level-wise screening experiments using locating arrays, Journal of Quality Technology, DOI: 10.1080/00224065.2023.2220973

**To link to this article:** https://doi.org/10.1080/00224065.2023.2220973

JOURNAL OF QUALITY TECHNOLOGY
https://doi.org/10.1080/00224065.2023.2220973



# Scalable level-wise screening experiments using locating arrays

Yasmeen Akhtar[a], Fan Zhang[b], Charles J. Colbourn[c], John Stufken[d], and Violet R. Syrotiuk[c]

[a]Department of Mathematics, Birla Institute of Technology and Science-Pilani Goa Campus, Sancoale, Goa, India; [b]School of Mathematical and Statistical Sciences, Arizona State University, Tempe, Arizona; [c]School of Computing and Augmented Intelligence, Arizona State University, Tempe, Arizona; [d]Department of Statistics, George Mason University, Fairfax, Virginia

**ABSTRACT**

Alternative design and analysis methods for screening experiments based on locating arrays are presented. The number of runs in a locating array grows logarithmically based on the number of factors, providing efficient methods for screening complex engineered systems, especially those with large numbers of categorical factors having different numbers of levels. Our analysis method focuses on levels of factors in the identification of important main effects and two-way interactions. We demonstrate the validity of our design and analysis methods on both well-studied and synthetic data sets and investigate both statistical and combinatorial properties of locating arrays that appear to be related to their screening capability.

**KEYWORDS**

algorithms; analysis of designed experiments; applications and case studies; experimental design

## 1. Introduction

When a system is complex, it is challenging to characterize and predict its behavior and performance. The power grid, the Internet, and transportation networks are examples of complex engineered networks that play an increasingly critical role in society. Yet our understanding of them remains limited (Chiang and Rao 2012).

Screening is often the first step in experimentation with a system, and its objective is to identify important factors that significantly impact the response variables. Screening experiments aim to eliminate factors that are not involved in any active effect, reducing the number of factors included in further experimentation; screening need not ensure that all remaining factors and their effects are important.

The size of the design spaces of complex systems dictates the need for scalable screening designs. Supersaturated designs are among the smallest screening designs, with the expected number of runs growing linearly based on the number of factors (Gilmour 2006; Li and Lin 2003; Lin 1993; Montgomery 2017; Nguyen 1996). Some screening designs aggregate the factors into groups, such as sequential bifurcation (Kleijnen, Bettonvil, and Persson 2006), to reduce the size of the design. Grouping requires care to ensure that factor effects do not cancel each other out. In studies of the Internet, specific levels of factors are known to give rise to active effects, and combinations of levels of factors may be exploited to improve network performance (see, e.g., Djama et al. 2008; Huang et al. 2020; Melodia and Akyildiz 2010; Sagduyu and Ephremides 2007; Shin, Park, and Kwon 2014; Song and Hatzinakos 2007; Vadde and Syrotiuk 2004; Verikoukis, Alonso, and Giamalis 2005). As systems become larger and more complex, domain expertise alone appears to be insufficient to make decisions concerning factor restriction or grouping, and techniques are needed to identify interactions and cope with categorical factors.

This paper is the first to demonstrate the feasibility of a *locating array*, a type of combinatorial design, as a screening design to address *all* these issues.

### 1.1. Scalability

Locating arrays have an expected number of runs that grows *logarithmically* based on the number of factors (Colbourn and McClary 2008), making it practical for experiments to consider more factors—an order of magnitude more—than allowed by traditional screening methods. This can eliminate the need for grouping.

**CONTACT** Violet R. Syrotiuk syrotiuk@asu.edu School of Computing and Augmented Intelligence, Arizona State University, Tempe, AZ 85281.
Supplemental data for this article can be accessed online at https://doi.org/10.1080/00224065.2023.2220973.

### 1.2. Level-wise screening

Locating arrays provide *t-way coverage*, ensuring that all $t$-way level combinations are present in the design; usually, the interest is in $t \leq 2$ because higher-order interactions tend to have less effect (Li, Sudarsanam, and Frey 2006; Montgomery 2017), but the definition is general. Locating arrays also satisfy a locating property that, together with coverage, naturally accommodates categorical factors. The proposed analysis algorithm using a locating array combined with a compressive sensing matrix validates the results of both widely studied experimental data and synthetic data sets. Information about specific levels of factors may be informative for experimenters and users of a system.

Section 2 provides a precise definition of locating arrays and a discussion of some of their properties.

As Section 3 details, a *compressive sensing matrix* (CSM) is used as the model matrix in the analysis method because it is well suited for identifying level-wise effects. Although over-parameterized, it is often effective in identifying a sparse explanation for variability in a response.

To illustrate these aspects of a CSM, we consider a contrived example. Table 1a shows a $3^2$ full-factorial design with two categorical factors $A$ and $B$, each with three levels and two responses, $y_1$ and $y_2$. The mean value for $y_1$ is $\overline{y_1} = \frac{1}{9}\sum_{i=1}^{9} y_{1i} = 5$. The mean response of $A$ at level 0, as well as at levels 1 and 2, is $\frac{2+7+6}{3} = 5$; hence, $A$ does not impact $y_1$. Similarly, $B$ does not impact $y_1$. Yet each of the two-way interaction effects of $A$ and $B$ deviates from 5 and impacts $y_1$.

The CSM for this example has 16 columns: three for each of two factors, one for each of nine two-way interactions, and one for the intercept; see Table 1b. When the CSM is used with the proposed screening method, nine terms with each two-way interaction of $A$ and $B$, except $A = 1$, $B = 1$, and the intercept, are identified for $y_1$. If dummy coding with $A = 1$ and $B = 1$ as the reference level is used with the proposed screening method, it also identifies nine terms: $A = 0$, $A = 2$, $B = 0$, $B = 2$, and the four two-way interactions between these levels for $A$ and $B$. Of course, these results are not wrong, but they may be difficult to interpret.

The mean value of $y_2$ is $\overline{y_2} = \frac{1}{9}\sum_{i=1}^{9} y_{2i} = 1$. The mean value for $y_2$ is 0 at $A = 0$, $A = 2$, $B = 0$, and $B = 2$, and it is 3 at $A = 1$ and $B = 1$. Of the interaction terms, only $A = 1$ and $B = 1$ has an impact on $y_2$. When the proposed screening method is used with a CSM, a single interaction term $A = 1$ and $B = 1$ is identified for $y_2$. If dummy coding with $A = 1$ and $B = 1$ is used as the reference level, it identifies all nine terms because it lacks $A = 1$ and $B = 1$ as a choice. Again, the terms identified are not wrong; this illustrates the issue with sparsity.

**Table 1.** (a) Value of responses $y_1$ and $y_2$ for a $3^2$-design. (b) The compressive sensing matrix for the design.

(a)

| Run | $A$ | $B$ | $y_1$ | $y_2$ |
|---|---|---|---|---|
| 1 | 0 | 0 | 2 | 0 |
| 2 | 0 | 1 | 7 | 0 |
| 3 | 0 | 2 | 6 | 0 |
| 4 | 1 | 0 | 9 | 0 |
| 5 | 1 | 1 | 5 | 9 |
| 6 | 1 | 2 | 1 | 0 |
| 7 | 2 | 0 | 4 | 0 |
| 8 | 2 | 1 | 3 | 0 |
| 9 | 2 | 2 | 8 | 0 |

(b)

| I | $A$ | | | $B$ | | | $AB$ | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 0 | 1 | 2 | 0 0 | 0 1 | 0 2 | 1 0 | 1 1 | 1 2 | 2 0 | 2 1 | 2 2 |
| + | + | − | − | + | − | − | + | − | − | − | − | − | − | − | − |
| + | + | − | − | − | + | − | − | + | − | − | − | − | − | − | − |
| + | + | − | − | − | − | + | − | − | + | − | − | − | − | − | − |
| + | − | + | − | + | − | − | − | − | − | + | − | − | − | − | − |
| + | − | + | − | − | + | − | − | − | − | − | + | − | − | − | − |
| + | − | + | − | − | − | + | − | − | − | − | − | + | − | − | − |
| + | − | − | + | + | − | − | − | − | − | − | − | − | + | − | − |
| + | − | − | + | − | + | − | − | − | − | − | − | − | − | + | − |
| + | − | − | + | − | − | + | − | − | − | − | − | − | − | − | + |

In Section 3, the proposed analysis method is presented in two algorithms. The first algorithm conducts a search in which many possible models with level-wise effects are developed to account for variability in the response. Effects are scored based on their perceived explanatory value. The second algorithm produces a ranking of the important terms through aggregation of the scores across the models developed.

Results using the proposed analysis method on widely studied data sets are presented in Section 4. The purpose of the study is to demonstrate that the proposed analysis method is effective, even for experiments with a small number of factors, and agrees with accepted results. To show that the method does not depend on the CSM, results using alternative model matrices are presented in Section 4.3.1. Guidance on using the tools developed is provided for practitioners in Section 4.1.

What, then, makes a locating array combined with the proposed method of analysis effective? Section 5 examines both statistical and combinatorial properties of designs that contribute to their ability to screen effectively. Also included is a comparison of the results of a locating array with a few supersaturated designs on synthetic data sets. Keeping in mind the coupling between design and analysis, a better

understanding of such properties and other characteristics may yield requirements that can be integrated into the construction of screening designs based on locating arrays. Finally, conclusions and future research directions are discussed in Section 6.

## 2. Locating arrays: Definition and background

Consider a system with $k$ factors, each having levels chosen from a set $S_i$ of cardinality $s_i$, $1 \le i \le k$. An $N \times k$ array $A$ with levels in the $i$th column chosen from a set $S_i$ has *type* $(s_1, ..., s_k)$. A *t-way level combination* consists of a choice of a set $I = \{i_1, ..., i_t\}$ of $t$ columns and the selection of levels $\sigma_i \in S_i$ for $i \in I$, and it is represented as $T = (i_1 = \sigma_{i_1}, ..., i_t = \sigma_{i_t})$. For brevity, we use the term *level-wise t-way interaction* for both the $t$-way level combination and the corresponding model effect.

A *covering array* of strength $t$ is an $N \times k$ array of type $(s_1, ..., s_k)$ in which, for every $N \times t$ subarray, each level-wise $t$-way interaction is *covered* (i.e., occurs) in at least one run (Hartman 2005). This generalizes the definition of an *orthogonal array*, which is a covering array of strength $t$ in which every interaction is covered exactly the same number of times. Table 2a shows a covering array $A_1$ of strength two for $k = 4$ factors of type $(2, 2, 3, 3)$. Nine runs suffice to cover all 37 of the level-wise two-way interactions. For example, the level-wise two-way interaction $(A = 0, C = 2)$ is covered in run five, shaded in blue. Covering arrays are a commonly used combinatorial testing method for real-world software (Kuhn, Kacker, and Lei 2013).

While a covering array of strength $t$ covers all level-wise $t$-way interactions, it does not ensure that it is possible to separate different interactions. For example, if the response measured for run five of $A_1$ deviates from that in the other runs, it is not possible to determine which of the three level-wise two-way interactions, $(A = 0, B = 1)$, $(A = 0, C = 2)$, or $(C = 2, D = 1)$, is responsible because each one occurs only in run five. Locating arrays extend covering arrays to address this very issue.

A $(d, t)$*-locating array* (Colbourn and McClary 2008) is a covering array of strength $t$ of type $(s_1, ..., s_k)$ with an additional property: any set of $d$ level-wise $t$-way interactions can be distinguished from any other such set by appearing in a distinct set of runs. If an array satisfies this condition, it has the $(d, t)$*-locating property*.

More precisely, for array $A$ and level-wise $t$-way interaction $T$, define $\rho_A(T)$ to be the set of runs of $A$ in which $T$ is covered. For a set $\mathcal{T}$ of level-wise $t$-way interactions, $\rho_A(\mathcal{T}) = \cup_{T \in \mathcal{T}} \rho_A(T)$. Let $\mathcal{I}_t$ be the set of all level-wise $t$-way interactions for an array, and let $\overline{\mathcal{I}_t}$ be the set of all level-wise interactions of size *at most* $t$. Consider a level-wise interaction $T \in \overline{\mathcal{I}_t}$ of size less than $t$. Any level-wise $t$-way interaction $T'$ of size $t$ that contains $T$ necessarily has $\rho_A(T') \subseteq \rho_A(T)$. Call a subset $\mathcal{T}'$ of level-wise $t$-way interactions in $\mathcal{I}_t$ *independent* if $T, T' \in \mathcal{T}'$ does not exist with $T \subseteq T'$.

**Table 2.** (a) A covering array $A_1$ of strength 2; (b) a $(1, \bar{2})$-locating array $A_2$.

| (a) | | | | | (b) | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Run | *A* | *B* | *C* | *D* | Run | *A* | *B* | *C* | *D* |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 1 | 2 | 0 | 0 | 1 | 1 |
| 3 | 0 | 0 | 1 | 0 | 3 | 0 | 0 | 0 | 2 |
| 4 | 0 | 0 | 1 | 2 | 4 | 0 | 1 | 2 | 1 |
| 5 | 0 | 1 | 2 | 1 | 5 | 0 | 1 | 2 | 2 |
| 6 | 1 | 0 | 2 | 2 | 6 | 0 | 1 | 1 | 0 |
| 7 | 1 | 1 | 0 | 2 | 7 | 0 | 1 | 1 | 1 |
| 8 | 1 | 1 | 1 | 1 | 8 | 1 | 0 | 2 | 2 |
| 9 | 1 | 1 | 2 | 0 | 9 | 1 | 0 | 1 | 1 |
| | | | | | 10 | 1 | 0 | 0 | 1 |
| | | | | | 11 | 1 | 1 | 2 | 0 |
| | | | | | 12 | 1 | 1 | 0 | 0 |
| | | | | | 13 | 1 | 1 | 1 | 2 |

Both arrays have type $(2, 2, 3, 3)$.

**Definition 2.1** ($(d, t)$-Locating Array (Colbourn and McClary 2008)).

An array $A$ is $(d, t)$*-locating* if, whenever $\mathcal{T}_1, \mathcal{T}_2 \subseteq \mathcal{I}_t$, $|\mathcal{T}_1| = d$, and $|\mathcal{T}_2| = d$, it holds that $\rho_A(\mathcal{T}_1) = \rho_A(\mathcal{T}_2) \iff \mathcal{T}_1 = \mathcal{T}_2$.

The definition is extended to permit level-wise interactions of at most $t$, writing $\bar{t}$ in place of $t$, by permitting that $\mathcal{T}_1, \mathcal{T}_2 \subseteq \overline{\mathcal{I}_t}$ and requiring that $\mathcal{T}_1$ and $\mathcal{T}_2$ be independent. For all instances of locating arrays, the relationships among them, and numerous examples, see Colbourn and McClary (2008); see also Section A of the Supplementary Material. For a more detailed discussion of the combinatorial requirements of locating arrays as generalizations of covering arrays, see Colbourn and Syrotiuk (2018).

The covering array $A_1$ in Table 2a does not have the $(1, \bar{2})$-locating property because the set of three level-wise two-way interactions $\mathcal{T} = \{(A = 0, B = 1), (A = 0, C = 2), (C = 2, D = 1)\}$ has $\rho_{A_1}(\mathcal{T}) = \{5\}$. However, the array $A_2$ in Table 2b is a $(1, \bar{2})$-locating array. For each level-wise two-way interaction in $\mathcal{T}$ there is a run that distinguishes it from the others: $\rho_{A_2}((A = 0, B = 1)) = \{4, 5, 6, 7\}$, $\rho_{A_2}((A = 0, C = 2)) = \{4, 5\}$, and $\rho_{A_2}((C = 2, D = 1)) = \{4\}$.

To quantify the degree to which level-wise $t$-way interactions can be distinguished in an array, the *separation* between sets of runs for different sets of level-wise $t$-way interactions is introduced (Seidel,

Sarkar, et al. 2018). More precisely, for a positive integer $\delta$, an array $A$ is $(d, t, \delta)$-locating if, whenever $\mathcal{T}_1, \mathcal{T}_2 \subseteq \mathcal{I}_t$, $|\mathcal{T}_1| = d, |\mathcal{T}_2| = d$, we have $|(\rho_A(\mathcal{T}_1) \cup \rho_A(\mathcal{T}_2)) \setminus (\rho_A(\mathcal{T}_1) \cap \rho_A(\mathcal{T}_2))| \geq \delta$. A $(d, t, \delta)$-locating array guarantees that any two sets of $d$ level-wise $t$-way interactions are separated by at least $\delta$ runs. By definition, a locating array has a separation of at least one: for example, $A_2$ is $(1, \bar{2}, 1)$-locating. A locating array with larger $\delta$ is more robust to, for example, outliers or missing data; however, there is a tradeoff between large $\delta$ and small array size.

Colbourn and McClary (2008) use covering arrays of strength $t + 1$ for constructing $(1, t)$-locating arrays and show that their size grows logarithmically based on the number of factors. Their rate of growth is slower than that of orthogonal arrays, which, by the Rao bound, are known to grow polynomially based on the number of factors (Hedayat, Sloane, and Stufken 1999). A locating array is preferable to a covering array for screening because it can separate the effects of any two $t$-way level-wise interactions; it is preferable to an orthogonal array because it is more economical in terms of run size.

Martínez et al. (2010) establish feasibility conditions for a locating array to exist. Tang, Colbourn, and Yin (2012) provide a general construction method for locating arrays. Colbourn and Fan (2016) develop three recursive constructions for locating arrays when $(d, t) = (1, 2)$. The size-optimality of a $(1, 1)$-locating array is studied in Colbourn, Fan, and Horsley (2016). Seidel, Sarkar, et al. (2018) discuss two randomized algorithms for constructing locating arrays based on the Stein-Lovász-Johnson paradigm and the Lovász Local Lemma. The implementation of these algorithms is publicly available (Seidel 2019).

## 3. A level-wise screening method

The proposed level-wise screening method assumes a locating array as the screening design, with one or more response vectors. A compressive sensing matrix (CSM) is used to represent the model matrix for the level-wise effects. For each response, the analysis method identifies a user-specified number of level-wise models that provide the "best" explanations for the response. Aggregation over these models, usually restricted to level-wise one- and two-way effects, results in the identification of important factors for each response. Each of these steps is described next.

### 3.1. The screening design and model matrix

A $(1, \bar{2})$-locating array is proposed as the screening design for two reasons. The coverage and locating properties of such an array are essential for separating level-wise one- and two-way effects. And, as Section 3.2 describes, the proposed analysis method recovers the "strongest" level-wise main effect or two-way interaction, one iteration at a time.

For the model matrix, a $\pm 1$ CSM is proposed in which the columns correspond to an intercept and all level-wise one- and two-way effects. A similar idea has been used for the recovery of sparsifiable signals in communications and storage systems (Baraniuk 2007).

A CSM for an $N \times k$ $(1, \bar{2})$-locating array $A$ of type $(s_1, \ldots, s_k)$ has as many rows as runs in $A$, and it has columns corresponding to the candidate level-wise terms. Specifically, the CSM $M = (m_{ij})$ has $N$ rows and $\sum_{1 \leq i \leq k} s_i + \sum_{1 \leq i < j \leq k} s_i s_j + 1$ columns, where $m_{ij} = +1$ if level-wise effect $j$ is covered in the $i$th run of $A$, and $m_{ij} = -1$ otherwise. A column of all $+1$ is required for the intercept. Table 3 shows the CSM for the locating array $A_2$ in Table 2b. For compactness of representation, we write $\pm$ instead of $\pm 1$.

### 3.2. The screening method

To achieve a small run size, locating arrays may exhibit a highly unbalanced structure. This requires the development of a method for level-wise screening that can cope with imbalance. (Locating arrays for few factors are typically close to balanced; imbalance increases with the number of factors.)

The proposed level-wise screening method has two steps. First, a *breadth-first search* (BFS) algorithm is developed to identify a user-specified number of level-wise models that are the "best" explanations of a response using *orthogonal matching pursuit* (OMP; Davis, Mallat, and Avellaneda 1997), which is widely used in signal processing (Tropp and Gilbert 2007) to recover sparse signals. A matching pursuit is a greedy algorithm that progressively refines an approximation of an optimization problem with an iterative procedure instead of solving it optimally. The vector selected at each iteration by the matching pursuit algorithm is generally not orthogonal to the previously selected vectors. In OMP, the approximations are refined by orthogonalizing the directions of projection.

Second, using the models produced in the BFS, the screening algorithm aggregates level-wise main effects and two-way interactions to identify the candidate important effects. The "many-model" method (Holcomb, Montgomery, and Carlyle 2003) also

**Table 3.** The compressive sensing matrix for the locating array $A_2$ in Table 2b.

Compressive Sensing Matrix

| I | A | | B | | C | | | D | | | AB | | | | AC | | | | | | AD | | | | | | BC | | | | | | BD | | | | | | CD | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 0 | 1 | 0 | 1 | 2 | 0 | 1 | 2 | 0 0 | 0 1 | 1 0 | 1 1 | 0 0 | 0 1 | 0 2 | 1 0 | 1 1 | 1 2 | 0 0 | 0 1 | 0 2 | 1 0 | 1 1 | 1 2 | 0 0 | 0 1 | 0 2 | 1 0 | 1 1 | 1 2 | 0 0 | 0 1 | 0 2 | 1 0 | 1 1 | 1 2 | 2 0 | 2 1 | 2 2 |
| + | + | − | + | − | + | − | − | + | − | − | + | − | − | − | + | − | − | − | − | − | + | − | − | − | − | − | + | − | − | − | − | − | + | − | − | − | − | − | − | − | − |
| + | + | − | + | − | − | + | − | − | + | − | + | − | − | − | − | + | − | − | − | − | − | + | − | − | − | − | − | + | − | − | − | − | − | − | − | − | + | − | − | − | − |
| + | + | − | + | − | + | − | − | − | − | + | + | − | − | − | + | − | − | − | − | − | − | − | + | − | − | − | + | − | − | − | − | − | + | − | − | − | − | − | − | − | − |
| + | + | − | − | + | − | − | + | − | + | − | − | + | − | − | − | − | + | − | − | − | − | + | − | − | − | − | − | − | − | − | + | − | − | − | − | − | − | − | + | − |
| + | + | − | − | + | − | − | + | − | − | + | − | + | − | − | − | − | + | − | − | − | − | − | + | − | − | − | − | − | − | − | + | − | − | − | − | − | − | − | − | + |
| + | + | − | − | + | − | + | − | − | − | + | − | − | + | − | − | − | − | + | − | − | − | − | − | + | − | − | − | − | − | − | + | − | − | − | − | − | + | − | − | − |
| + | + | − | − | + | − | + | − | − | + | − | − | + | − | − | − | − | + | − | − | − | − | − | − | − | + | − | − | − | − | − | + | − | − | − | − | − | − | + | − | − |
| + | − | + | + | − | − | − | + | − | − | + | − | − | + | − | − | − | − | − | + | − | − | − | − | − | + | − | − | + | − | − | − | − | − | − | − | − | − | − | − | − | + |
| + | − | + | + | − | − | + | − | − | + | − | − | − | + | − | − | − | − | − | + | − | − | − | − | − | + | − | − | + | − | − | − | − | − | − | − | − | − | + | − | − | − |
| + | − | + | + | − | + | − | − | − | + | − | − | − | + | − | − | − | − | − | + | − | − | − | − | − | + | − | + | − | − | − | − | − | − | − | − | − | − | − | + | − | − |
| + | − | + | − | + | − | − | + | + | − | − | − | − | − | + | − | − | − | − | − | + | − | − | − | − | − | + | − | − | − | − | − | − | + | − | − | − | − | − | − | + | − | − |
| + | − | + | − | + | + | − | − | + | − | − | − | − | − | + | − | − | − | − | − | + | − | − | − | − | − | + | − | − | − | − | − | − | + | − | − | − | − | − | − | − | − | − |
| + | − | + | − | + | − | + | − | − | − | + | − | − | − | + | − | − | − | − | − | + | − | − | − | − | − | + | − | − | − | − | − | − | − | − | − | − | − | + | − | − | − |

The columns correspond to the intercept (I), followed by each main effect and two-way interaction given level-wise.

retains a fraction of best models based on the error sum of squares, but it does not appear to scale to large numbers of factors.

### 3.2.1. The breadth-first search (BFS) algorithm

The BFS algorithm is parameterized by three user-specified variables: $n_{models}$ giving the number of fitted models that the algorithm returns, $n_{new}$ giving the fan-out (i.e., number of children) of each node in the BFS tree, and $n_{terms}$ giving the number of effects in each of the final fitted models. In the BFS tree, the nodes at level $\ell$ correspond to fitted models with $\ell$ effects. The algorithm generates a BFS tree of height $n_{terms}$.

The BFS algorithm is given in Algorithm 1; Figure 1 illustrates the BFS tree. The root of the tree is a single model consisting of the mean response and a score initialized to zero. A BFS expands each node at level $\ell$ to $n_{new}$ nodes at level $\ell+1$ (line 8). For efficiency, the search tree is stored implicitly, with each model of length $\ell$ stored in priority queue $q_\ell$. Each child expands the fitted model of its parent by adding the $i$th most important effect for $1 \le i \le n_{new}$ using OMP (line 9). Specifically, the $i$th most important effect corresponds to the $i$th effect in the ranking of the absolute values of the dot products or correlations of each column in the CSM with the current residual vector. The model is expanded to include the $i$th effect. Then the *ordinary least squares* (OLS) method (Searle 1987) is used to update coefficient estimates in the expanded model, after which the residuals are updated and the score of the added effect is computed (lines 10–14). (OMP for logistic regression (Lozano, Świrszcz, and Abe 2011) can be used for binary responses.)

The increment in $R^2$ of the expanded model that results from adding the $i$th effect is used as the score of the effect. The model, its residuals, its $R^2$ and adjusted $R^2$, and its scores are then inserted into the priority queue of length $\ell+1$ (line 15). The priority queue of length $\ell+1$ retains at most $n_{models}$ in decreasing order by $R^2$ rank. These steps are repeated until a stopping criterion is met. To simplify the algorithm, it stops when each model has $n_{terms}$ level-wise effects (line 5). The model matrix, stopping criterion, and scoring method of the BFS algorithm can each be chosen differently. Indeed, even the search tree itself can be explored using a different method, such as a branch-and-bound or Least-Absolute Shrinkage and Selection Operator (LASSO) (Tibshirani 1996) approach to solving the best-subset selection problem (Miller 1990).

When only level-wise main effects and two-way interactions are considered, a $(1,\bar{2})$-locating array suffices. All effects are separable under such a design—that is, all columns in the CSM are distinct. For binary factors, the absolute values for the two main-effects columns are equal, and either can be selected. The dot product is easy to compute, and it ranks effects based on absolute correlation with residuals of the current model.

**Algorithm 1** BFS($terms, M, data, n_{models}, n_{terms}, n_{new}$)

**Input:** List of candidate *effects*, CSM $M$, response vector *data*, number of fitted models $n_{models}$ to return, number of effects in each final fitted model $n_{terms}$, fan-out of the BFS tree $n_{new}$

**Output:** List of $n_{models}$ best fitted models ranked by $R^2$ with $n_{terms}$ terms each

1: $model_{new} \leftarrow$ mean of data
2: $residuals_{new} \leftarrow data - model_{new}$
3: $scores_{new} \leftarrow 0$
4: enqueue($q_1$, ($model_{new}, residuals_{new}, R^2, adjR^2, scores_{new}$))
5: **for** $\ell \leftarrow 1, \ldots, n_{terms}$ **do**
6: **while** $q_\ell$ is nonempty **do**
7: $(model, residuals, R^2, adjR^2, scores) \leftarrow dequeue(q_\ell)$

8: **for** $i \leftarrow 1, \ldots, n_{new}$ **do**
9: $effects_k \leftarrow \text{argmax}_i |M_i \cdot residuals|$
10: $model_{new} \leftarrow \text{OLS}(effects(model) \cup effects_k, data)$
11: $residuals_{new} \leftarrow data - model_{new}$
12: ${R^2}_{new} \leftarrow R^2$ value of $model_{new}$
13: ${adjR^2}_{new} \leftarrow$ adjusted $R^2$ value of $model_{new}$
14: $scores_{new} \leftarrow$ append increment in $R^2$ attributed to $effects_k$ to $scores$ {Retain at most $n_{models}$ in the priority queue ranked by $R^2$ value}
15: enqueue($q_{\ell+1}$, ($model_{new}$, $residuals_{new}$, ${R^2}_{new}$, ${adjR^2}_{new}$, $scores_{new}$))
16: **end for**
17: **end while**
18: **end for**
19: **return** list of $n_{models}$ fitted models from $q_{n_{terms}}$ ranked by $R^2$ value

It is possible for duplicate fitted models to arise in $q_\ell$, such as when the same terms are selected but in a different order. While only unique fitted models are kept in the queue, duplicates are accounted for by adding the scores of each effect from the duplicate. Thus, duplication is not ignored; it allows more models to be explored.

#### 3.2.2. The screening algorithm

In screening, the objective is to identify a few important main effects and two-way interactions. One approach is to examine the scores of the level-wise effects in the list of fitted models produced and select those with higher scores. Instead, the approach in Algorithm 2 aggregates the effects of all $n_{models}$ without explicitly considering levels.

**Algorithm 2** Screening($effects$, $q_{n_{terms}}$, $n_{models}$, $n_{terms}$)

**Input:** List of all candidate *effects*, list of fitted models $q_{n_{terms}}$ and corresponding scores from Algorithm 1, number of fitted models $n_{models}$ in the list, number of effects $n_{terms}$ in each fitted model
**Output:** A list of effects in nonincreasing order by score

Initialize the score of each effect to zero
**for** $i \leftarrow 1, \ldots, n_{models}$ **do**
  $(model_i, scores_i) \leftarrow dequeue(q)$
  **for** $j \leftarrow 1, \ldots, n_{terms}$ **do**
    $k \leftarrow$ index of $effects$ corresponding to term $j$ in $model_{i_j}$
    $effect\text{-}score_k = effect\text{-}score_k + scores_{i_j}$
  **end for**
**end for**
**return** list of $effects$ ranked by $effect\text{-}score$

Effects are reported in nonincreasing order by their aggregate scores to support user interpretation of the results. Screening results are usually reported with a consideration of heredity; therefore, if an interaction effect $X \times Y$ is reported as active, then $X$ and $Y$ are also considered active (Li, Sudarsanam, and Frey 2006).

## 4. Analyzing data from real experiments: Guidance and validation

### 4.1. Guidance for practitioners

To construct a locating array with parameters $(d, t, \delta)$ for $k$ factors of type $(s_1, \ldots, s_k)$, see Section B of the Supplementary Material for available construction tools; it also describes tools to extract a locating array from an existing design and a tool to extend an array with additional runs until it satisfies the requirements of a locating array. Locating arrays appear to have no strong competitors as screening designs capable of isolating one- and two-way effects efficiently.

If the array used for a screening experiment is an $N \times k$ $(1, \bar{2})$-locating array of type $(s_1, \ldots, s_k)$, then the corresponding CSM has $N$ rows and $\sum_{1 \le i \le k} s_i + \sum_{1 \le i < j \le k} s_i s_j + 1$ columns. The run time of Algorithm 1 is

$$O\left(n_{terms} \times n_{models} \times n_{new}\left(\left(\sum_{1 \le i \le k} s_i + \sum_{1 \le i < j \le k} s_i s_j + 1\right) \times N^2\right)\right),$$

that is, it runs in polynomial time. As the fan-out $n_{new}$ increases, more models are evaluated. As the number of models $n_{models}$ decreases, many models are likely to be discarded, as the priority queue retains only $n_{models}$ models at each level. As $n_{terms}$ increases, successive effects identified by OMP in each iteration are likely to have a lower score.

While the choice of $n_{new}$, $n_{models}$, and $n_{terms}$ is application dependent, we have found that $n_{models} = 50$ and $n_{new} = 50$ work well in practice. For the number of terms, we suggest starting with $n_{terms} = 2$ when using the analysis tool (see Section B.3 of the Supplementary Material). When $n_{terms}$ is incremented but the screening result (i.e., the list of top effects returned by Algorithm 2) remains the same, there is likely no need to increment $n_{terms}$ again and explore models with more terms.

### 4.2. Validation of widely studied screening experiments

To validate the results of a screening experiment, a locating array must be extracted as a subset of the runs of the original screening designs because it is not feasible to conduct new runs of the experiment; see

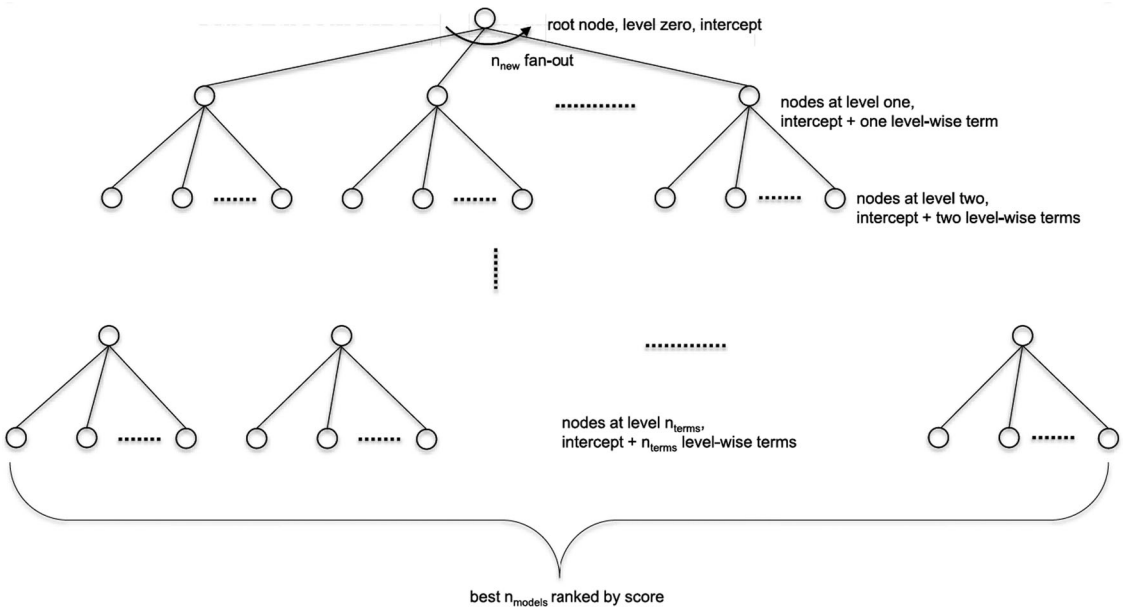


**Figure 1.** The breadth-first search tree generated by the analysis method. It has a fan-out $n_{new}$ from which the best $n_{models}$ models ranked by a score of $R^2$ are retained at each level in the priority queue. $n_{models}$ each with at most $n_{terms}$ level-wise terms are ultimately returned.

Section B.1 of the Supplementary Material for details. If a locating array can be extracted, along with the data collected in each run, the analysis can be performed.

Next, we validate the results of several widely studied real-world screening experiments using $n_{models} = 50$, $n_{new} = 50$, and starting at $n_{terms} = 2$. The purpose of this validation study is not to advocate the use of locating arrays but to demonstrate that the proposed analysis method is effective, even for experiments with a small number of factors, and agrees with accepted results.

### 4.3. Chemical reactor experiment

Box, Hunter, and Hunter (2005) describe a chemical reactor experiment with five binary factors $A$–$E$. The experiment is run as a $2^5$ full-factorial, and the original analysis indicates that $B, D, E, B \times D$, and $D \times E$ are active. Miller and Sitter (2001) extract runs from the full-factorial design that correspond to a folded-over 12-run Plackett-Burman design and identify the same set of effects.

Table 4 shows a $(1, \bar{2}, 1)$-locating array with 9 runs and a $(1, \bar{2}, 2)$-locating array with 11 runs extracted from the full-factorial design. The fitted models generated for the 9-run locating array have $R^2 < 0.71$ when $n_{terms} = 2$, $0.72 < R^2 < 0.92$ when $n_{terms} = 3$, $0.93 < R^2 < 0.97$ when $n_{terms} = 4$, and $R^2 > 0.98$ when $n_{terms} = 5$. Table 5a lists the top five terms and their scores reported by Algorithm 2. Similarly, the fitted models generated for the 11-run locating array have $R^2 < 0.74$ when $n_{terms} = 2$, $0.74 < R^2 < 0.95$ when $n_{terms} = 3$, and $R^2 > 0.95$ when $n_{terms} = 4$. Table 5b reports these results.

The interaction $B \times D$ has the highest score for both locating arrays. Because the score for $D \times E$ is much lower, it may or may not be active. If the top two terms are considered and heredity is used, then $B, D, E, B \times D$, and $D \times E$ are the selected effects. This suggests that the factors $B$, $D$, and $E$ should be included in follow-up experimentation.

Algorithms 1 and 2 identify the influential factors using fewer runs. Moreover, our screening results are consistent across different locating arrays. This indicates that the proposed screening method is robust and does not rely on the choice of runs in a locating array. Similar conclusions were found for a contaminant and a cast fatigue experiment; see Sections C and D, respectively, of the Supplementary Material.

#### 4.3.1. Other model parameterizations instead of the compressive sensing matrix

For illustration, we consider alternative parameterizations for the chemical reactor experiment in Section 4.3. To use dummy coding, for a factor with $k$ levels, we add $\{0, 1\}$ indicator columns for the first $k - 1$ levels of the factor and add two-way interaction-effect columns by multiplying pairs of main-effect columns for different factors. Then we replace the 0s in those columns by $-1$s. The resulting matrix is used instead of the CSM for screening the data with the two locating arrays in Table 4. When $n_{terms} = 8$, $R^2 > 0.98$; Table 6 lists the top five

**Table 4.** (a) A 9-run $(1, \bar{2}, 1)$-locating array; (b) an 11-run $(1, \bar{2}, 2)$-locating array for the chemical reactor experiment.

| (a) | | | | | | | (b) | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Run | *A* | *B* | *C* | *D* | *E* | *y* | Run | *A* | *B* | *C* | *D* | *E* | *y* |
| 1 | + | + | + | + | + | 82 | 1 | − | + | + | − | + | 67 |
| 2 | + | − | − | + | − | 61 | 2 | − | + | − | + | + | 78 |
| 3 | − | + | − | − | + | 70 | 3 | + | + | + | + | + | 82 |
| 4 | + | + | + | − | − | 61 | 4 | − | − | − | − | − | 61 |
| 5 | − | − | − | + | + | 44 | 5 | + | − | − | + | + | 45 |
| 6 | − | − | − | − | − | 61 | 6 | + | + | − | − | − | 61 |
| 7 | − | + | + | + | − | 95 | 7 | − | − | + | + | + | 49 |
| 8 | + | + | − | + | + | 77 | 8 | + | − | − | + | − | 61 |
| 9 | − | − | + | − | − | 53 | 9 | − | + | + | + | − | 95 |
| | | | | | | | 10 | + | − | + | − | − | 56 |
| | | | | | | | 11 | + | − | − | − | + | 63 |

**Table 5.** Top five effects and their scores for the chemical reactor experiment using: (a) the 9-run locating array and (b) the 11-run locating array in Table 4.

| (a) | | (b) | |
|---|---|---|---|
| Term | Score | Term | Score |
| $B \times D$ | 153.08 | $B \times D$ | 145.74 |
| $D \times E$ | 29.31 | $D \times E$ | 28.76 |
| $C \times D$ | 15.74 | $B$ | 3.90 |
| $A \times B$ | 11.53 | $E$ | 1.00 |
| $B$ | 8.72 | $B \times C$ | 0.87 |

**Table 6.** Top five effects and their scores for the chemical reactor experiment using dummy coding: (a) the 9-run locating array; (b) the 11-run locating array in Table 4.

| (a) | | (b) | |
|---|---|---|---|
| Term | Score | Term | Score |
| $B \times D$ | 2571.77 | $B \times D$ | 5189.11 |
| $D \times E$ | 528.26 | $D \times E$ | 1061.99 |
| $C \times D$ | 476.19 | $B \times C$ | 208.21 |
| $B$ | 454.00 | $C \times E$ | 105.16 |
| $C$ | 114.91 | $B$ | 63.47 |

**Table 7.** Top five effects and their scores for the chemical reactor experiment using common effects model: (a) the 9-run locating array; (b) the 11-run locating array in Table 4.

| (a) | | (b) | |
|---|---|---|---|
| Term | Score | Term | Score |
| $B$ | 1842.99 | $B$ | 5597.76 |
| $B \times D$ | 477.66 | $B \times D$ | 2976.89 |
| $A \times E$ | 431.26 | $D \times E$ | 1095.41 |
| $A \times B$ | 374.05 | $D$ | 714.69 |
| $D \times E$ | 309.29 | $E$ | 223.36 |

terms and their scores. With these parameters, the scores for some effects become very high because they occur in many of the $n_{models} = 50$ fitted models, some of which are duplicated more than 200 times.

Table 7 presents the top five terms and their scores using the more typical parameterization for two-level factors consisting of a matrix that contains a single column for each main effect and each two-way interaction instead of using the CSM for the two locating arrays in Table 4 ($n_{terms} = 8$, $R^2 > 0.98$).

**Table 8.** Top five effects and their scores for the rubber experiment using 20-run locating arrays (a) $A_1$; (b) $A_2$; (c) the 28-run design from Sundberg (2008), which is a $(1, \bar{2}, 4)$-locating array.

| (a) | | (b) | | (c) | |
|---|---|---|---|---|---|
| **Term** | **Score** | **Term** | **Score** | **Term** | **Score** |
| $N$ | 68.43 | $I \times N$ | 40.57 | $N$ | 69.16 |
| $P \times S$ | 8.44 | $D \times P$ | 13.50 | $I \times P$ | 17.44 |
| $H \times P$ | 6.96 | $M \times U$ | 11.25 | $P \times S$ | 10.43 |
| $E \times S$ | 3.53 | $A \times D$ | 8.81 | $D \times S$ | 8.01 |
| $C \times P$ | 1.87 | $D \times H$ | 1.30 | $D \times M$ | 6.53 |

The results in Tables 6 and 7 indicate that using the common main effects and two-factor interactions agree with the original analysis better than using dummy-coded variables. They also suggest that the 11-run locating array performs better than the 9-run locating array.

### 4.4. Rubber experiment

Williams (1968) conducted an experiment to improve a rubber-making process. This data set has appeared in numerous studies focusing on supersaturated designs; see Section E of the Supplementary Material for a history of the analyses performed. Sundberg (2008) observed that the original data set has an outlier in run 14 and suggested that the data be analyzed on the log scale. We extracted two distinct $(1, \bar{2}, 1)$-locating arrays $A_1$ and $A_2$ (in Tables E.1 and E.2 in Section E) with 20 runs from the 28-run data set in Sundberg (2008). A log transformation of the data was performed before analysis.

Table 8a reports the top five terms and their scores for $A_1$ when $n_{terms} = 5$; all fitted models have $R^2 \geq 0.96$. Table 8b reports the same results for $A_2$ when $n_{terms} = 7$; all fitted models have $R^2 \geq 0.97$. Screening using $A_1$ indicates that the main effect of $N$ dominates all other terms, while screening using $A_2$ suggests that the interaction $I \times N$ has a significant impact on the response. A possible reason for these scores could be the presence of the outlier in $A_2$; because $A_2$ has $\delta = 1$, a single outlier reduces its locating ability. Nevertheless, by heredity, $N$ is identified as one of the important factors. Therefore, we are able to recover the active effect by screening using 20-run instead of 28-run designs.

As it turns out, the 28-run design used by Sundberg (2008) is a $(1, \bar{2}, 4)$-locating array. Table 8c reports the top five effects and their scores when $n_{terms} = 7$. The proposed screening method identifies factor $N$ as the dominant active effect, even though the outlier is included in this data set. This suggests that a locating array with higher separation ($\delta = 4$) is able to reduce the impact of outliers.

### 4.5. Wireless network test bed experiment

Seidel, Mehari, et al. (2018) studied a Wi-Fi conferencing scenario where a speaker broadcasts voice traffic over a Wi-Fi network testbed to listeners. The speaker can configure 24 different factors of type $(2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 5, 5, 5, 5, 5, 5, 5, 5, 5)$ that may influence the responses. These factors span the operating system kernel's networking stack, the Wi-Fi card driver, the audio codec used, and a source of interference implemented via a dedicated radio. The listeners continuously calculate audio quality and exposure to radio frequencies (RF).

The original analysis indicated that the factors interference channel occupancy (`intCOR`), channel band (`band`), transmission power (`txpower`), and Wi-Fi bit rate (`rate`) have a significant impact on audio quality. Similarly, `rate`, `txpower`, audio codec bit rate (`codecBitrate`), frame length aggregation (`frameLen`), and `band` show a significant impact on RF exposure.

Using the proposed analysis method based on a 109-run $(1, \bar{2}, 1)$-locating array, Table 9a shows the top five effects and their scores for audio quality when $n_{terms} = 9$; in this case, the fitted models have $0.74 \leq R^2 \leq 0.76$. For $n_{terms} = 21$, all the fitted models have $R^2 \geq 0.96$; the results are shown in Table 9b. The method identifies `txpower` and the interaction `intCOR` × `band` as having a significant impact on audio quality.

Table 9c lists the top five effects and their scores for RF exposure when $n_{terms} = 10$; these fitted models have $R^2 \geq 0.96$. The proposed screening method identifies `txpower` and `band` as the important factors for RF exposure.

We also analyze the data from the wireless network experiment using two additional methods: the Dantzig selector method (Phoa, Pan, and Xu 2009) and LASSO regression (Tibshirani 1996). For each method, we use two different coding schemes: dummy coding (as described in Section 4.3.1) and orthogonal polynomial coding.

For the Dantzig selector, we use the R package "flare" (Li 2013). We scale the columns of the model matrix and set the tuning parameter $\delta_{Dantzig} = 0.1$ and the number of $\delta_{Dantzig}$ to 11; for how to choose these parameters, see Phoa, Pan, and Xu (2009). We fix $\gamma$ (a threshold between signal and noise) at zero to keep all selected effects. In LASSO regression, we use the R package "glmnet" for analysis (Friedman, Hastie, and Tibshirani 2009).

Table 10 summarizes the terms found by each of the three methods. There is good agreement on the screening results for both responses when the

**Table 9.** Top five effects and their scores for audio quality when (a) $n_{terms} = 9$ and (b) $n_{terms} = 21$ and for (c) RF exposure in the wireless network experiment.

| (a) | | (b) | | (c) | |
|---|---|---|---|---|---|
| Term | Score | Term | Score | Term | Score |
| `intCOR×band` | 325.51 | `txpower` | 24961.10 | `txpower` | 1489.92 |
| `txpower` | 320.00 | `intCOR×band` | 18237.70 | `band` | 1035.76 |
| `intCOR` | 68.29 | `intCOR` | 14748.40 | `rate` | 589.91 |
| `intCOR×sensing` | 35.36 | `intCOR×sensing` | 7358.25 | `frameLen` | 149.49 |
| `codecBitrate×channel` | 35.47 | `sensing×band` | 5242.84 | `codecBitrate` | 44.02 |

**Table 10.** Screening results for wireless network experiment listed by the method used. Only significant factors are listed.

| Method | Audio Quality | RF Exposure |
|---|---|---|
| Proposed method | `txpower`<br>`intCOR`<br>`band` | `txpower`<br>`band`<br>`rate` |
| Dantzig selector (polynomial coding) | `txpower`<br>`intCOR`<br>`band` | `txpower`<br>`band`<br>`rate` |
| Dantzig selector (dummy coding) | `txpower`<br>`band`<br>`rate`<br>`udp_mem_pressure`<br>`ipfrag_high_thresh` | `band`<br>`txpower` |
| LASSO regression (polynomial coding) | `txpower`<br>`intCOR`<br>`band` | `txpower`<br>`band`<br>`rate` |
| LASSO regression (dummy coding) | `band`<br>`rate`<br>`txpower` | `txpower`<br>`band` |

Table 11. A comparison of properties of the three designs $D_0$, $D_1$, and $D_2$ used in the study.

| Design | Covering Array | Locating Array | Factor-Level Balance | $E(s^2)$- or $UE(s^2)$ Optimality | max\|s\| Value | r-Rank |
|---|---|---|---|---|---|---|
| $D_0$ | strength 2 | $(1,\bar{2})$ | ✗ | ✗ | 8 | 17 |
| $D_1$ | strength 2 | (1, 1) | ✗ | ✓ | 8 | 9 |
| $D_2$ | strength 2 | (1, 1) | ✓ | ✗ | 6 | 15 |

polynomial model is used. However, with dummy coding, neither the Dantzig selector nor the LASSO regression method appears to be as accurate.

## 5. Combinatorial and statistical properties of designs

In an effort to understand what contributes to the ability of locating arrays to screen effectively, we discuss a few of their combinatorial and statistical properties.

To understand the significance of the locating property in our screening method, we study it in a more controlled environment. For the study, we use three supersaturated designs, $D_0$, $D_1$, and $D_2$, each with 18 runs for 21 binary factors; see Tables F.1, F.2, and F.3, respectively, in Section F of the Supplementary Material.

All three designs are strength-two covering arrays. Only $D_0$ is a $(1,\bar{2},1)$-locating array. While the minimum size of a strength-two covering array for 21 binary factors is eight runs, additional runs are required to satisfy the locating property. The lower bound on the number of runs for a $(1,2)$-locating array for 21 binary factors is 12; see Theorem 2.2 in Tang, Colbourn, and Yin (2012).

The $E(s^2)$-criterion for choosing binary supersaturated designs minimizes the sum of squares of the information matrix entries for a main-effects model over balanced designs. The $UE(s^2)$-criterion is similar to the $E(s^2)$-criterion, except that the requirement of factor-level balance is dropped (Jones and Majumdar 2014; Cheng et al. 2018). Both $D_0$ and $D_1$ are unbalanced in terms of factor-level occurrences, but only $D_1$ is $UE(s^2)$-optimal. The locating array $D_0$ does not satisfy either optimality criterion. Though balanced, $D_2$ is not $E(s^2)$-optimal.

The max$|s|$-criterion for binary designs considers correlations between columns of the model matrix for the main-effects model and selects a design that minimizes the maximum absolute correlation. $D_2$ has a smaller max$|s|$-criterion than $D_0$ and $D_1$. The resolution rank ($r$-rank) for a binary design is the largest value $r$ so that all main effects are estimable for any model with $r$ main effects (Deng, Lin, and Wang 1999); a discussion of this property follows a generalization of $r$-rank later in this section. The properties of $D_0$, $D_1$, and $D_2$ are summarized in Table 11.

We obtain simulated data from $D_0$, $D_1$, and $D_2$ using the 98 linear models in Table G.1 in Section G of the Supplementary Material. The models studied explore the impact of indistinguishable pairs of level-wise interactions. The arrays $D_1$ and $D_2$, which do not satisfy the $(1,\bar{2})$-locating property, have 21 and 22 pairs of level-wise interactions not separated by any runs, respectively.

Table G.1 lists the results of our proposed screening method with $n_{terms} = 6$ and the Dantzig selector method with standard orthogonal polynomial coding, minimum $\delta_{Dantzig} = 0.1, 0.25$, number of $\delta_{Dantzig} = 11$, and $\gamma = 0$. The results indicate that the proposed screening method using the locating array $D_0$ is able to recover the interactions.

Here we discuss the first two models in Table G.1, $y_1$ and $y_2$, in some detail:

$$y_1 = 7 \times Ind(D=1, O=-1) + Ind(D=1, O=1) + Ind(D=-1, O=1) + Ind(D=-1, O=-1) + \varepsilon \quad [1]$$

$$y_2 = 7 \times Ind(B=1, P=-1) + Ind(B=1, P=1) + Ind(B=-1, P=1) + Ind(B=-1, P=-1) + \varepsilon \quad [2]$$

where the indicator function $Ind(\cdot) = 1$ only in runs where the factors equal the levels specified. The error term $\varepsilon$ is randomly selected from $N(0,1)$.

The two-way interactions $(D=1, O=-1)$ and $(L=1, S=-1)$ are indistinguishable in $D_2$ because $\rho_{D_2}((D=1, O=-1)) = \rho_{D_2}((L=1, S=-1)) = \{6, 13, 18\}$. Table 12 shows that, for $y_1$, the interaction $D \times O$ has a relatively high score for $D_0$ and $D_1$, whereas both $D \times O$ and $L \times S$ have high scores for $D_2$.

Table 13 shows the top five scores for $y_2$. In this case, the two-way interactions $(B=1, P=-1)$ and $(S=1, T=-1)$ are indistinguishable in $D_1$ because $\rho_{D_1}((B=1, P=-1)) = \rho_{D_1}((S=1, T=-1)) = \{8, 10, 12\}$. The proposed method identifies the interactions $B \times P$ when analyzing $y_2$ using $D_2$ and $D_0$. However, two interactions, $B \times P$ and $S \times T$, have the same score when analyzing $y_2$ using $D_1$. Not

**Table 12.** The top five scores for response $y_1$ in Model 1 produced by the proposed screening method using each of the arrays $D_0$, $D_1$, and $D_2$.

| $D_0$ | | $D_1$ | | $D_2$ | |
|---|---|---|---|---|---|
| Term | Score | Term | Score | Term | Score |
| $D \times O$ | 86.37 | $D \times O$ | 155.15 | $D \times O$ | 36.43 |
| $I \times J$ | 11.30 | $A \times Q$ | 34.30 | $L \times S$ | 36.41 |
| $I \times N$ | 1.65 | $B$ | 1.20 | $B \times U$ | 6.17 |
| $D \times L$ | 1.26 | $M \times O$ | 1.85 | $J \times S$ | 4.67 |
| $M \times N$ | 1.24 | $B \times D$ | 1.46 | $L \times R$ | 2.69 |

**Table 13.** The top five scores for response $y_2$ in Model 2 produced by our screening method using each of the arrays $D_0$, $D_1$, and $D_2$.

| $D_0$ | | $D_1$ | | $D_2$ | |
|---|---|---|---|---|---|
| Term | Score | Term | Score | Term | Score |
| $B \times P$ | 70.42 | $B \times P$ | 20.30 | $B \times P$ | 80.05 |
| $I \times J$ | 8.47 | $S \times T$ | 20.30 | $B \times U$ | 3.61 |
| $D \times L$ | 1.55 | $A \times Q$ | 10.22 | $L \times U$ | 2.51 |
| $D \times Q$ | 1.53 | $B \times S$ | 2.95 | $F \times M$ | 2.09 |
| $I \times M$ | 1.48 | $A \times S$ | 0.87 | $L \times R$ | 1.03 |

surprisingly, these examples indicate that arrays without the locating property are unable to distinguish level-wise interactions that are not separated.

We now look for some statistical properties that contribute to locating arrays' ability to screen effectively. The $r$-rank of a design has been used as a statistical indicator of screening effectiveness for main-effect models for binary factors. Since we are also interested in two-way interaction effects, we propose a generalization for binary designs that also considers interaction effects:

**Definition 5.1** ($(r,i)$-Rank)

Let $X = (X_1|X_2)$, where $X_1$ corresponds to all main-effects columns and $X_2$ to all two-factor interaction columns under the common parameterization for binary factors. For a given $i \geq 0$, the $(r,i)$-rank of $X$ is defined as the largest value $r$ so that all effects are estimable for any model with $r - i$ columns from $X_1$ and $i$ columns from $X_2$.

The $(r,0)$-rank is just the usual $r$-rank. The $(r,i)$-ranks for the three designs used in the simulation study are given in Table 14. The $(r,0)$- and $(r,2)$-ranks of $D_0$ are greater than those of $D_1$ and $D_2$. When we compare the screening performance of $D_1$ and $D_2$ in Table G.1, the design $D_2$ appears better able to recover the interactions. This suggests the possibility of sequentially maximizing the $(r,i)$-rank for improved screening and estimation of interaction effects.

**Table 14.** The $(r, i)$-ranks for the designs $D_0$, $D_1$, and $D_2$.

| Rank | $D_0$ | $D_1$ | $D_2$ |
|---|---|---|---|
| $(r,0)-$rank | 17 | 9 | 15 |
| $(r,1)-$rank | 12 | 8 | 16 |
| $(r,2)-$rank | 5 | 3 | 3 |

Returning to combinatorial properties, the *separation* between sets of runs for different sets of level-wise interactions is an indicator of screening effectiveness. It quantifies the degree to which level-wise interactions can be distinguished, and a high separation can be useful when dealing with outliers or missing data. Intuitively, the better the ability to distinguish between sets of interactions, the better the screening effectiveness. For example, both the 28-run design and the 20-run design $A_2$ in the rubber experiment in Section 4.4 contain an outlier. The 28-run design has a separation of four, while $A_2$ has a separation of one. The screening results in Table 8 indicate that the 28-run design performs better.

To compare locating arrays of the same dimensions with the same separation, we introduce a new quantifier:

**Definition 5.2** ($(d,t,\delta)$-Separation Deficiency)

For a given strength $t$ locating array $A$ and positive integers $d$ and $\delta$, the $(d,t,\delta)$-separation deficiency is the number of pairs of sets $\mathcal{T}_1, \mathcal{T}_2 \subseteq \mathcal{I}_t$ of cardinality $d$ for which $|(\rho_A(\mathcal{T}_1) \cup \rho_A(\mathcal{T}_2)) \setminus (\rho_A(\mathcal{T}_1) \cap \rho_A(\mathcal{T}_2))| < \delta$.

The $(d,\bar{t},\delta)$-*separation deficiency* is defined in a similar fashion by considering level-wise interactions of strength at most $t$. An array with a $(d,t,\delta)$-separation deficiency of zero is a $(d,t,\delta)$-locating array. When $\delta = 1$, such an array is simply a $(d,t)$-locating array. A $(d,t,\delta)$-locating array with lower $(d,t,\delta+1)$-separation deficiency is preferred over the other $(d,t,\delta)$-locating arrays for the proposed screening method.

We believe that an array with lower deficiency is better able to distinguish between sets of level-wise interactions. For example, the design matrix in the rubber experiment with 24 binary factors in Section 4.4 has 1012 level-wise two-way interactions in $\mathcal{I}_2$, and no two of them are covered in the same set of runs in arrays $A_1$ and $A_2$ in Tables E.1 and E.2. For any $T_1, T_2 \in \mathcal{I}_2$, $|(\rho_{A_1}(\mathcal{T}_1) \cup \rho_{A_1}(\mathcal{T}_2)) \setminus (\rho_{A_1}(\mathcal{T}_1) \cap \rho_{A_1}(\mathcal{T}_2))| \geq 1$. Therefore, $A_1$ is a $(1,\bar{2},1)$-locating array. However, $A_1$ is not $(1,\bar{2},2)$-locating. There are 218 pairs of two-way interactions that can be distinguished using only one run in $A_1$. Therefore, the $(1,\bar{2},2)$-separation deficiency of $A_1$ is 218.

Similarly, $A_2$ is a $(1,\bar{2},1)$-locating array, and there are 161 pairs of two-way interactions that can be distinguished using only one run in $A_2$. Consequently, $A_2$ is not a $(1,\bar{2},2)$-locating array. The $(1,2,2)$-separation deficiency of $A_2$ is 161. The $(1,\bar{2},\delta)$-separation deficiencies for $A_1$, $A_2$, and the 28-run design in Sundberg (2008) used in the rubber experiment in Section 4.4 are given in Table 15.

**Table 15.** The $(1, 2, \delta)$-separation deficiency for the designs $A_1$, $A_2$, and 28-run design in Sundberg (2008)

| | $(1, 2, \delta)$-separation deficiency | | | | |
|---|---|---|---|---|---|
| Design | $\delta = 1$ | $\delta = 2$ | $\delta = 3$ | $\delta = 4$ | $\delta = 5$ |
| $A_1$ | 0 | 218 | 1925 | 9727 | 37308 |
| $A_2$ | 0 | 161 | 1832 | 10432 | 38972 |
| 28-run design (Sundberg 2008) | 0 | 0 | 0 | 0 | 2544 |

Keeping in mind the coupling between design and analysis, a more thorough understanding of these and other statistical and combinatorial properties may yield requirements that can be integrated into the construction of screening designs based on locating arrays.

## 6. Conclusions and future work

This paper proposed designs and methods of analysis for screening experiments based on locating arrays. Locating arrays grow logarithmically based on the number of factors and are therefore well suited for identifying the factors that significantly impact the response variables of complex systems. Because such systems may have a large number of categorical factors with many levels, the proposed analysis method focuses on level-wise effects. This suggests the use of a parameterization method that works well to identify these effects, such as the compressive sensing matrix. As demonstrated by results on many real data sets, the analysis method appears to screen correctly using fewer runs than the original designs. Locating arrays with high separation may also provide some resistance to outliers or missing responses.

While we have demonstrated that the proposed design and analysis methods validate results of small, well-studied data sets, we anticipate additional advantages for studies of more complex systems with many factors, some of which are categorical. A more diverse and in-depth study of the statistical and combinatorial properties of locating arrays may better inform design construction and analysis.

## About the authors

**Yasmeen Akhtar** is an Assistant Professor at the Department of Mathematics, BITS Pilani K K Birla Goa Campus, India. She received her Ph.D. from the Indian Institute of Science Education and Research (IISER) Pune, India. She held postdoctoral positions at the Institute of Statistical Science, Academia Sinica, Taiwan, and the School of Computing, Informatics & Decision Systems Engineering (now the School of Computing and Augmented Intelligence), Arizona State University, USA. Her research interests include applied combinatorics and the design and analysis of experiments.

**Fan Zhang** is currently a Ph.D. candidate in the School of Mathematical and Statistical Sciences, Arizona State University. She received her B.S. degree in Statistics from the Beijing Normal University, China, in 2016, and M.S. degree in Statistics from the National Cheng Kung University, Taiwan, in 2018. Her research interests include design and analysis of experiments.

**Charles J. Colbourn** earned his Ph.D. in 1980 from the University of Toronto in Computer Science. He has held academic positions at the University of Saskatchewan, the University of Waterloo, the University of Vermont, and is currently a Professor of Computer Science and Engineering at Arizona State University. Dr. Colbourn is the author of The Combinatorics of Network Reliability (Oxford) and Triple Systems (Oxford). He is editor-in-chief of the Journal of Combinatorial Designs. He edited The CRC Handbook of Combinatorial Designs. He is the author of about 400 refereed journal papers focusing on combinatorial designs and graphs with applications in networking, computing, and communications. He was awarded the Euler Medal for Lifetime Research (2003) and the Stanton Medal for Lifetime Service (2020) by the Institute for Combinatorics and its Applications.

**John Stufken** is a Professor of Statistics at George Mason University. His primary research interests include design and analysis of experiments and analysis of big data. He is a former Editor for the Journal of Statistical Planning and Inference and for The American Statistician. He held positions as Program Director for Statistics (National Science Foundation), Head of Department of Statistics (University of Georgia), Coordinator for Statistics (Arizona State University), and Director for Informatics and Analytics (University of North Carolina at Greensboro). He is an Elected Fellow of the Institute of Mathematical Statistics and the American Statistical Association and an Elected Member of the International Statistical Institute.

**Violet R. Syrotiuk** is an Associate Professor of Computer Science and Engineering in the School of Computing and Augmented Intelligence at Arizona State University. Her research interests include applications of combinatorial design theory to computer networks, self driving networks, and programmable networks. Her work makes use of mid-scale experimental infrastructure including the new NSF FABRIC testbed. Dr. Syrotiuk serves on the editorial boards of Elsevier's COMNET and COMCOM journals and is a TPC Co-Chair of IFIP Networking 2023.

## Acknowledgements

The authors sincerely thank the editor and the referees for their valuable comments, which have greatly improved our paper.

We thank Stephen Seidel for writing the code to construct locating arrays and to analyze the data collected from their use in the experimentation. Thanks also to Isaac Jung for additional construction algorithms.

## Disclosure statement

No potential conflict of interest was reported by the authors.

## Funding

The work of Yasmeen Akhtar, Charles J. Colbourn, and Violet R. Syrotiuk was supported by the US National Science Foundation (NSF) under Grants NeTS-1813451 and NeTS-1421058. The work of John Stufken was supported by the US National Science Foundation (NSF) under Grants DMS-1935729 and DMS- 2304767.

## ORCID

Yasmeen Akhtar http://orcid.org/0000-0002-7875-852X
Fan Zhang http://orcid.org/0009-0008-3105-063X
Charles J. Colbourn http://orcid.org/0000-0002-3104-9515
John Stufken http://orcid.org/0000-0001-9026-3610
Violet R. Syrotiuk http://orcid.org/0000-0002-5025-0337

## Data availability statement

The data that support the findings of this study are openly available at https://www.public.asu.edu/∼syrotiuk/tools.html.

## References

Abraham, B., H. Chipman, and K. Vijayan. 1999. Some risks in the construction and analysis of supersaturated designs. *Technometrics* 41 (2):135–41. doi: 10.1080/00401706.1999.10485634.

Baraniuk, R. G. 2007. Compressive sensing. *IEEE Signal Processing Magazine* 24 (4):118–21. doi: 10.1109/MSP.2007.4286571.

Box, G. E. P., J. S. Hunter, and W. G. Hunter. 2005. *Statistics for experimenters*. 2nd ed. Hoboken, NJ: John Wiley & Sons, Inc.

Cheng, C.-S., A. Das, R. Singh, and P.-W. Tsai. 2018. $E(s^2)$- and $UE(s^2)$-optimal supersaturated designs. *Journal of Statistical Planning and Inference* 196:105–14. doi: 10.1016/j.jspi.2017.10.012.

Chiang, M., and N. Rao. 2012. Networking and information technology research and development (NITRD) large scale networking (LSN) workshop report on complex engineered networks, September.

Colbourn, C. J., and B. Fan. 2016. Locating one pairwise interaction: Three recursive constructions. *Journal of Algebra Combinatorics Discrete Structures and Applications* 3 (3):127–34.

Colbourn, C. J., B. Fan, and D. Horsley. 2016. Disjoint spread systems and fault location. *SIAM Journal on Discrete Mathematics* 30 (4):2011–26. doi: 10.1137/16M1056390.

Colbourn, C. J., and D. W. McClary. 2008. Locating and detecting arrays for interaction faults. *Journal of Combinatorial Optimization* 15 (1):17–48. doi: 10.1007/s10878-007-9082-4.

Colbourn, C. J., and V. R. Syrotiuk. 2018. On a combinatorial framework for fault characterization. *Mathematics in Computer Science* 12 (4):429–51,December. doi: 10.1007/s11786-018-0385-x.

Davis, G., S. Mallat, and M. Avellaneda. 1997. Greedy adaptive approximation. *Constructive Approximation* 13 (1):57–98. doi: 10.1007/BF02678430.

Deng, L.-Y., D. K. J. Lin, and J. Wang. 1999. A resolution rank criterion for supersaturated designs. *Statistica Sinica* 9 (2):605–10.

Djama, I., T. Ahmed, A. Nafaa, and R. Boutaba. 2008. Meet in the middle cross-layer adaptation for audiovisual content delivery. *IEEE Transactions on Multimedia* 10 (1):105–20. doi: 10.1109/TMM.2007.911243.

Draper, N. R., and H. Smith. 1981. *Applied regression analysis*. 2nd ed. New York: John Wiley & Sons, Inc.

Friedman, J., T. Hastie, and R. Tibshirani. 2009. URL glmnet: LASSO and elastic-net regularized generalized linear models. R package version 1.1-4. http://CRAN.R-project.org/package=glmnet.

Gilmour, S. 2006. Factor screening via supersaturated designs. In *Screening*, ed. A. Dean and S. Lewis, 169–90. New York: Springer.

Hartman, A. 2005. Software and hardware testing using combinatorial covering suites. Chap. 10 in M. C. Golumbic and I. B.-A. Hartman, editors, *Graph theory, combinatorics and algorithms: Interdisciplinary application*, 237–66. New York: Springer US.

Hedayat, A. S., N. J. A. Sloane, and J. Stufken. 1999. *Orthogonal arrays. Springer Series in Statistics*. New York: Springer-Verlag.

Holcomb, D. R., D. C. Montgomery, and W. M. Carlyle. 2003. Analysis of supersaturated designs. *Journal of Quality Technology* 35 (1):13–27. doi: 10.1080/00224065.2003.11980188.

Huang, K., C. Zhou, Y. Qin, and W. Tu. 2020. A game-theoretic approach to cross-layer security decision-making in industrial cyber-physical systems. *IEEE Transactions on Industrial Electronics* 67 (3):2371–9. doi: 10.1109/TIE.2019.2907451.

Hunter, G. B., F. S. Hodi, and T. W. Eagar. 1982. High cycle fatigue of weld repaired cast Ti-6AI-4V. *Metallurgical Transactions A* 13 (9):1589–94. doi: 10.1007/BF02644799.

Jones, B., and D. Majumdar. 2014. Optimal supersaturated designs. *Journal of the American Statistical Association* 109 (508):1592–600. doi: 10.1080/01621459.2014.938810.

Jung, I. 2022. Covering, locating, and detecting array generator. https://github.com/gatoflaco/Array-Generator.

Kleijnen, J. P. C., B. Bettonvil, and F. Persson. 2006. Screening for the important factors in large discrete-even simulation models: Sequential bifurcation and its applications. Chap. 13 in *Screening: Methods for experimentation in industry, drug discovery and genetics*, ed. A. M. Dean and S. M. Lewis, 287–307. New York, NY: Springer-Verlag.

Kuhn, D. R., R. N. Kacker, and Y. Lei. 2013. *Introduction to combinatorial testing*. New York, NY: Chapman and Hall/CRC.

Li, R., and D. K. J. Lin. 2003. Analysis methods for supersaturated designs: Some comparisons. *Journal of Data Science* 1 (3):249–60.

Lin, D. K. J. 1993. A new class of supersaturated designs. *Technometrics* 35 (1):28–31. doi: 10.1080/00401706.1993.10484990.

Li, X., N. Sudarsanam, and D. D. Frey. 2006. Regularities in data from factorial experiments. *Complexity* 11 (5):32–45. doi: 10.1002/cplx.20123.

Li, X. 2013. URL Family of LASSO regression, R package flare. https://www.rdocumentation.org/packages/flare/versions/0.9.8.

Lozano, A. C., G. Świrszcz, and N. Abe. 2011. Group orthogonal matching pursuit for logistic regression. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS), Volume PMLR 15*, 452–60.

Martínez, C., L. Moura, D. Panario, and B. Stevens. 2010. Locating errors using ELAs, covering arrays, and adaptive testing algorithms. *SIAM Journal on Discrete Mathematics* 23 (4):1776–99. doi: 10.1137/080730706.

Melodia, T., and I. F. Akyildiz. 2010. Cross-layer QoS-aware communication for ultra wide band wireless multimedia sensor networks. *IEEE Journal on Selected Areas in Communications* 28 (5):653–63. doi: 10.1109/JSAC.2010.100604.

Miller, A. J. 1990. *Subset selection in regression*. London, UK: Chapman and Hall.

Miller, A., and R. R. Sitter. 2001. Using the folded-over 12-run Plackett-Burman design to consider interactions. *Technometrics* 43 (1):44–55. doi: 10.1198/00401700152404318.

Montgomery, D. C. 2017. *Design and analysis of experiments*. 9th ed. Hoboken, NJ: John Wiley & Sons, Inc.

Nguyen, N.-K. 1996. An algorithmic approach to constructing supersaturated designs. *Technometrics* 38 (1):69–73. doi: 10.1080/00401706.1996.10484417.

Phoa, F. K. H., Y.-H. Pan, and H. Xu. 2009. Analysis of supersaturated designs via the Dantzig selector. *Journal of Statistical Planning and Inference* 139 (7):2362–72. doi: 10.1016/j.jspi.2008.10.023.

Sagduyu, Y. E., and A. Ephremides. 2007. On joint MAC and network coding in wireless ad hoc networks. *IEEE Transactions on Information Theory* 53 (10):3697–713. doi: 10.1109/TIT.2007.904980.

Searle, S. R. 1987. *Linear models for unbalanced data*. New York, NY: John Wiley & Sons.

Seidel, S. A., K. Sarkar, C. J. Colbourn, and V. R. Syrotiuk. 2018. Separating interaction effects using locating and detecting arrays. In *Combinatorial algorithms*, ed. C. Iliopoulos, H. W. Leong, and W.-K. Sung, 349–60. Cham: Springer International Publishing.

Seidel, S. A. 2019. Tools for locating arrays. https://github.com/sseidel16/v4-la-tools.

Seidel, S. A., M. T. Mehari, C. J. Colbourn, E. De Poorter, I. Moerman, and V. R. Syrotiuk. 2018. Analysis of large-scale experimental data from wireless networks. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 535–540, April.

Shin, C., H. Park, and H. M. Kwon. 2014. PHY-supported frame aggregation for wireless local area networks. *IEEE Transactions on Mobile Computing* 13 (10):2369–81. doi: 10.1109/TMC.2014.2304393.

Song, L., and D. Hatzinakos. 2007. A cross-layer architecture of wireless sensor networks for target tracking. *IEEE/ACM Transactions on Networking* 15 (1):145–58. doi: 10.1109/TNET.2006.890084.

Sundberg, R. 2008. A classical dataset from Williams, and its role in the study of supersaturated designs. *Journal of Chemometrics* 22 (7):436–40. doi: 10.1002/cem.1167.

Tang, Y., C. J. Colbourn, and J. Yin. 2012. Optimality and constructions of locating arrays. *Journal of Statistical Theory and Practice* 6 (1):20–9. doi: 10.1080/15598608.2012.647484.

Tibshirani, R. 1996. Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society: Series B (Methodological)* 58 (1):267–88. doi: 10.1111/j.2517-6161.1996.tb02080.x.

Tropp, J. A., and A. C. Gilbert. 2007. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory* 53 (12):4655–66. doi: 10.1109/TIT.2007.909108.

Vadde, K. K., and V. R. Syrotiuk. 2004. Factor interaction on service delivery in mobile ad hoc networks. *IEEE Journal on Selected Areas in Communications* 22 (7):1335–46. doi: 10.1109/JSAC.2004.829351.

Verikoukis, C., L. Alonso, and T. Giamalis. 2005. Cross-layer optimization for wireless systems: A European research key challenge. *IEEE Communications Magazine* 43 (7):1–3, July. doi: 10.1109/MCOM.2005.1470797.

Wang, P. C., N. Kettaneh-Wold, and D. K. J. Lin. 1995. Comments on Lin (1993). *Technometrics* 37 (3):358–9. doi: 10.2307/1269944.

Westfall, P. H., S. S. Young, and D. K. J. Lin. 1997. Forward selection error control in the analysis of supersaturated designs. *Statistica Sinica* 8:101–17.

Williams, K. R. 1968. Designed experiments. *Rubber Age* 100:65–71.

Wu, C. F. J., and M. Hamada. 2009. *Experiments: Planning, analysis, and optimization*. 2nd ed. Hoboken, NJ: John Wiley & Sons, Inc.