

ROBUSTNESS OF RECOVERY IN LOCATING ARRAY-BASED SCREENING EXPERIMENTS

Stephen A. Seidel
Charles J. Colbourn
Violet R. Syrotiuk

School of Computing, Informatics, and Decision Systems Engineering
Arizona State University, Tempe, AZ, USA

ABSTRACT

Locating arrays (LAs) are experimental designs for screening interactions in engineered systems. LAs are often highly unbalanced, requiring advanced techniques to recover the terms that significantly influence system performance. While perfect recovery is achieved in the absence of noise, real systems are noisy. Therefore, in this paper, we study the robustness of recovery in the presence of noise. Using known models to generate synthetic data, we investigate recovery accuracy as a function of noise. *Separation* is introduced into LAs to allow more coverage for each t -way interaction; when separation is higher, recovery in noisy scenarios should improve. We find that locating arrays are able to recover the influential terms even with high levels of noise and that separation appears to improve recovery. Under the pessimistic assumption that noise depends on the range of responses, it is no surprise that terms with small coefficients become indistinguishable from noise.

Keywords: screening engineered networks, locating arrays, analysis, robustness, recovery.

1 INTRODUCTION

Complexity arises in an engineered system not only from its size, but also from its structure, operation (including control and management), and its evolution over time (Chiang and Rao 2012). Experimentation is often used to study the performance of such systems. A system may be viewed as transforming input variables, or *factors*, into one or more measurable output variables, or *responses*. *Screening experiments* have the objective to identify the factors that are most influential on a response.

Our interest is screening in engineered networks. To reduce their design complexity, most network architectures are layered. The purpose of each layer is to offer services to higher layers, following protocols for communication. Networks have a very large number of factors that potentially influence responses such as throughput, delay, or energy. Interestingly, interactions among factors are known to arise. An example of a two-way cross-layer interaction, between the link and transport layer, is the TCP protocol interpreting access delays in a wireless network due to poor signal quality as congestion, and hence responding incorrectly with congestion control (Cáceres and Ifode 1995). A few other examples of interactions in networks include Athanasiou, Korakis, Ercetin, and Tassiulas (2009), Efazati and Azmi (2016), Cammarano, Presti, Maselli, Pescosolido, and Petrioli (2015), and Wang, Vuran, and Goddard (2012).

We are therefore motivated to screen for interactions, not just factors, influencing performance. The exhaustive *full-factorial design* (Montgomery 2017) could serve this purpose. For k factors each with only two levels, or *values*, such a design has size 2^k . In engineered systems with large k , an exponential number of tests is typically infeasible. Naturally, methods to reduce the number of tests have been developed. Perhaps of most interest are *supersaturated designs*; these contain more factors than tests in the experiment. Supersaturated designs are used to estimate a main effects model (Li and Lin 2003, Montgomery 2017). Thus they do not consider possible interactions at all. If an interaction is not covered, *i.e.*, does not appear in the design, its influence on performance is never observed. Hence there is a need for small screening designs that locate interactions, even when the number of factors is very large. Locating arrays (LAs), introduced in Section 2, address this need.

Through analysis of the measurements collected in each test of the experiment, the influential factors and interactions impacting performance must be recovered. For full-factorial and other balanced designs, *i.e.*, those having uniform or near-uniform coverage, one technique is the *analysis of variance* (ANOVA). Methods for analysis of supersaturated designs include the partially Bayesian method (Gilmour 2006). These analysis methods cannot be used for locating arrays because they are often highly unbalanced. To address this problem, in Aldaco, Colbourn, and Syrotiuk (2015), factors are grouped according to their coverage in the LA. First the most significant factor or interaction was selected from each group using the Wilcoxon rank sum test and the Mann-Whitney U -test (Mann and Whitney 1947, Wilcoxon 1945). From these candidates, the most significant term overall is selected using the Akaike information criterion (Akaike 1974). This process is repeated on the residuals until stopping criteria were met. In an effort to automate recovery, in Compton, Mehari, Colbourn, De Poorter, and Syrotiuk (2016) an analysis method is developed based on *orthogonal matching pursuit* (OMP), as described in Section 3. Using it, Seidel, Mehari, Colbourn, De Poorter, Moerman, and Syrotiuk (2018) show agreement with the statistical-based recovery method. In this paper we use the OMP-based analysis method.

While perfect recovery is achieved in the absence of noise, real systems are noisy. Often, we do not know the ground truth when we analyze the data collected from a network simulation or from an engineered network testbed. Therefore, in this paper, we study the robustness of recovery for known ground truth so that we can evaluate the accuracy of our recovery. Specifically, using known models to generate synthetic data, we vary how much the model satisfies the heavy-hitters assumption, and we vary the noise level. We apply our OMP-based analysis algorithm to recover terms and evaluate the accuracy of the recovered terms to those of the known model. *Separation* is introduced into locating arrays to allow more coverage for each t -way interaction. Intuition suggests that when the separation is higher, recovery in noisy scenarios should improve. Hence we also study recovery for LAs of higher separation.

The primary contribution of this paper is a study of the robustness of the recovery in screening experiments based on locating arrays. Our study, presented in Section 4, indicates that as the heavy-hitters assumption is more strongly satisfied in scenarios with noise, it is difficult to recover the terms with smallest coefficients. Under the pessimistic assumption that noise depends on the range of responses this is no surprise. Furthermore, our results in Section 5 suggest that separation in locating arrays can help the recovery process in the presence of noise. These results improve confidence in our recovery when applied to data collected from engineered networks, whether simulated or real. Our conclusions and future work follow in Section 6.

2 LOCATING ARRAYS

Suppose that the system of interest has k factors, F_1, \dots, F_k , and that each factor F_j has a set $V_j = \{v_{j,1}, \dots, v_{j,\ell_j}\}$, of ℓ_j possible values. A *test* is an assignment of a value from V_j to F_j , for each factor $j = 1, \dots, k$. An *experimental design* (or, design for short) is a collection of tests.

When a design has N tests (or, has *size* N), it may be represented by an $N \times k$ array $A = (a_{ij})$ in which each row i corresponds to a test and each column j to a factor; the entry a_{ij} specifies the value assigned to factor j in the i th test. When run on the system, a test results in the measurement of one or more responses. An *experiment* consists of running each test in the design.

An assignment of values to any subset $t \leq k$ of the factors is a t -way *interaction*. A *covering array* of strength t , is an $N \times k$ array in which for every $N \times t$ subarray, each t -way interaction is *covered* (i.e., occurs) in at least one test (Hartman 2005). An example of a covering array of strength two for three factors, A having two levels $\{1, 2\}$, and B and C each with three $\{1, 2, 3\}$, is given in Table 1.

Table 1: A covering array A_C of strength 2; adding three tests gives a $(1, 2)$ -locating array A_L .

	Test	A	B	C	
Covering Array A_C	1	1	1	2	Locating Array A_L
	2	1	1	3	
	3	1	2	1	
	4	1	3	3	
	5	2	1	1	
	6	2	2	2	
	7	2	2	3	
	8	2	3	1	
	9	2	3	2	
	10	1	2	3	
	11	1	3	1	
	12	2	1	2	

A covering array does not ensure that it is possible to distinguish the influence of different t -way interactions. For example, if the measurement for test seven is different from the other tests, it is not possible to determine which of the two, two-way interactions $(A = 2) \& (C = 3)$ and $(B = 2) \& (C = 3)$ is responsible, because each is covered only by test seven. Locating arrays extend covering arrays to address this very issue.

A (d, t) -*locating array* is a covering array of strength t with an additional property: Any set of d interactions each involving t parameters can be distinguished from any other such set by appearing in a distinct set of tests (Colbourn and McClary 2008). If a design satisfies this definition it has the (d, t) -*locating property*.

More precisely, for an $N \times k$ array A , define $\rho(A, T)$ as the set of tests of A in which T is covered. For a set \mathcal{T} of t -way interactions, $\rho(A, \mathcal{T}) = \cup_{T \in \mathcal{T}} \rho(A, T)$. Now, A is (d, t) -*locating* if $\rho(A, \mathcal{T}_1) = \rho(A, \mathcal{T}_2) \Leftrightarrow \mathcal{T}_1 = \mathcal{T}_2$ whenever $\mathcal{T}_1, \mathcal{T}_2$ are any sets of t -way interactions where $|\mathcal{T}_1| = d$, and $|\mathcal{T}_2| = d$. When $|\mathcal{T}_1| \leq d$ and $|\mathcal{T}_2| \leq d$ and $\mathcal{T}_1, \mathcal{T}_2$ are any sets of interactions with strength at most t , the array is (\bar{d}, \bar{t}) -*locating*. Thus a (\bar{d}, \bar{t}) -locating array guarantees that any two groups of *at most* d interactions with strength *at most* t are distinguishable from each other in the tests where they are covered.

The covering array in Table 1 does not have the $(1, 2)$ -locating property because the set of two-way interactions $\mathcal{T} = \{(A = 2) \& (C = 3), (B = 2) \& (C = 3)\}$ has $\rho(A_C, \mathcal{T}) = \{7\}$. However, the addition of three additional tests to the covering array in Table 1 produces a $(1, 2)$ -locating array A_L . Now, $\rho(A_L, (B = 2) \& (C = 3)) = \{7, 10\}$ while $\rho(A_L, (A = 2) \& (C = 3)) = \{7\}$, allowing the influence of the two interactions to be distinguished.

2.1 A Separation δ Requirement

In principle, the responses for each test can identify the set of influential interactions whenever the assumptions on number and strengths of interactions are met. In practice, however, a problem arises. Suppose that two sets of (at most) d interactions, \mathcal{T}_1 and \mathcal{T}_2 , have $|\rho(\mathcal{T}_1) \setminus \rho(\mathcal{T}_2)| = 1$. If this occurs, the response measured in a single test is the sole “witness” to the difference between the two. In the absence of noise or measurement error, one such witness suffices to differentiate. In our experiments, however, outliers and missing responses do occur. These compromise our ability to analyze the response data. This can be mitigated by exploring a number of possible sets of significant interactions, as in Seidel, Sarkar, Colbourn, and Syrotiuk (2018), rather than identifying a single set.

To improve the effectiveness of recovery in locating array-based screening, we must allow for outliers and missing responses. Fortunately this can be treated by further requirements on the array, by enforcing a separation between sets of rows for different sets of interactions. We make this precise next. Let $\delta \geq 1$ be an integer, the *distance*. Let \mathcal{I}_t be the set of all t -way interactions for an array A . An array A is (d, t, δ) -*locating* if whenever $\mathcal{T}_1, \mathcal{T}_2 \subseteq \mathcal{I}_t$, $|\mathcal{T}_1| = d$, and $|\mathcal{T}_2| = d$, we have that

$$|(\rho(A, \mathcal{T}_1) \cup \rho(A, \mathcal{T}_2)) \setminus (\rho(A, \mathcal{T}_1) \cap \rho(A, \mathcal{T}_2))| < \delta \Leftrightarrow \mathcal{T}_1 = \mathcal{T}_2.$$

This requires that at least δ tests witness the difference. The variants for \bar{d} and \bar{t} are immediate.

Separation by distance δ ensures that any $\delta - 1$ or fewer tests can fail to provide a response, or provide an outlier response, without losing the differentiation supported by the locating array.

3 ANALYSIS OF LOCATING ARRAY-BASED SCREENING EXPERIMENTS

We utilize *orthogonal matching pursuit* (OMP) (Tropp and Gilbert 2007) to iteratively recover the influential terms impacting a response. A screening design with the $(\bar{1}, \bar{2})$ -locating property suffices because in each iteration we recover one term ($\bar{d} = 1$), a main effect or a two-way interaction ($\bar{t} = 2$); we are able to do so provided that there is a unique strongest term, i.e., the term is a “heavy hitter.”

Algorithm 1 Orthogonal Matching Pursuit : $\text{OMP}(\text{terms}, M, \text{data})$

Require: List of candidate terms, compressive sensing matrix, vector of responses

Ensure: Model (list of pairs of coefficients and terms)

```

1: model  $\leftarrow$  [(mean of data, intercept term)]      /* Square brackets denote a list */
2: residuals  $\leftarrow$  data  $-$  model
3: while the stopping criterion is not yet met do
4:    $i \leftarrow \arg \max_i |\widehat{M}_i \cdot \widehat{\text{residuals}}|$ 
5:   model  $\leftarrow$  LLS(terms(model)  $\cup$  [termsi], data)      /*  $\cup$  denotes list concatenation */
6:   residuals  $\leftarrow$  data  $-$  model
7: end while
8: return model

```

Algorithm 1 maintains a model and a residual vector. The model is initialized as an intercept term equal to the mean of the measured responses, while the residuals are the data minus the model. In each iteration, we select a term to add to the model based on the dot product of the residuals with each column in the compressive sensing matrix, a matrix whose columns correspond the terms that may be included in the model. The term yielding the highest-magnitude dot product is added to the model, after which linear least squares is used to update the model coefficients. The residuals are then recalculated. These steps are repeated until a stopping criterion is met; see Compton, Mehari, Colbourn, De Poorter, and Syrotiuk (2016).

One disadvantage of Algorithm 1 is that it only adds terms to the model. If it finds other terms that explain one away, it cannot delete the term, only update its coefficient to be negligible. To counter this problem, we use a breadth-first search algorithm to consider alternatives for the choices it makes using OMP (Seidel, Mehari, Colbourn, De Poorter, Moerman, and Syrotiuk 2018). This approach produces and returns the top $nModels$ models, each with $nTerms$ terms. The occurrences of each factor in the top $nModels$ models are then counted, and the factors are ranked by number of occurrences. A factor *occurrence* is defined as any time a factor appears in a model, no matter its level, or if it is part of an interaction.

4 ROBUSTNESS STUDY

Algorithm 1 relies on the heavy-hitters assumption, *i.e.*, that there is one term that influences the response more significantly than all remaining terms, and when this term is removed, there is again one term that influences the response more significantly than all remaining terms. In other words, the influence of the terms follows an exponentially decreasing pattern. We study the recovery accuracy varying how much the model satisfies the heavy-hitters assumption, and also varying the noise in the system.

Suppose we are interested in screening a system with 100 factors, F_1, F_2, \dots, F_{100} , each with three levels, $\{v_1, v_2, v_3\}$. A $(\bar{1}, \bar{2}, 1)$ -locating array used for screening this system has just 70 rows. The parameters of the analysis algorithm used to produce all results are $nModels = 50$ and $nTerms = 8$.

Table 2 gives the true system model used in our study in this section. Throughout, the models in our study have the form $\beta + \sum_i c_i F_i + \sum_{i,j} c_{i,j} F_i F_j$, *i.e.*, they are linear models with intercept β , and terms that include main effects as well as two-way interactions from the 100 factors. Each term has a coefficient. In this scenario, the coefficient of every term is the same (0.1); hence the heavy-hitters assumption is not satisfied. Further scenarios in this section vary the coefficients of the model.

Table 2: Robustness Scenario 1A - Heavy-hitters not satisfied.

True Model – Coefficients exhibit 0% decrease	
Coefficient	Term
0.1	$F_{29} = v_1$
0.1	$F_{98} = v_3 \ \& \ F_{34} = v_2$
0.1	$F_{50} = v_2$
0.1	$F_{22} = v_1$
0.1	<i>INTERCEPT</i>
0.1	$F_{69} = v_1 \ \& \ F_{23} = v_1$
0.1	$F_{10} = v_2$
0.1	$F_{82} = v_1$

Table 3 shows two sets of recovery results. A check mark in the column labelled “True” indicates that the factors recovered of the given rank are in the true model. The first set of results shown in columns two through four in Table 3 corresponds to the recovery process without any noise introduced in the system. Although the true model in Table 2 does not satisfy the heavy-hitters assumption, all nine factors in the true model are recovered. However, the four factors ranked six through nine have occurrence counts that are less than 50 indicating that these factors did not appear in every model. The sudden drop-off in the occurrence count (from 30 to 6) between the ninth and tenth factors in the ranking is characteristic of noiseless data. When there is noise in the system the drop-off is often more gradual, making it more difficult to distinguish the influential factors.

Table 3: Robustness scenario 1A – Recovery.

Occurrence Counts						
Rank	No Noise ($\delta = 1$)			10% Noise ($\delta = 1$)		
	Count	Factor	True	Count	Factor	True
1	50	F_{82}	✓	50	F_{82}	✓
2	50	F_{50}	✓	50	F_{50}	✓
3	50	F_{29}	✓	50	F_{29}	✓
4	50	F_{22}	✓	50	F_{22}	✓
5	50	F_{10}	✓	50	F_{10}	✓
6	38	F_{23}	✓	41	F_{34}	✓
7	35	F_{34}	✓	40	F_{98}	✓
8	34	F_{98}	✓	23	F_{23}	✓
9	30	F_{69}	✓	21	F_{69}	✓
10	6	F_6		8	F_2	✓

In the absence of knowledge of noise, we make a pessimistic assumption that noise depends on the largest difference in the responses. More specifically, uniform random noise is added to our output responses with the magnitude of the noise characterized as a percentage of the range of all output responses.

Columns five through seven in Table 3 show occurrence counts when 10% noise is added to the scenario. The top five ranked factors are all main effects in the true model. The remaining factors (F_{34} , F_{98} , F_{23} , F_{69}) are all involved in interactions and show lower occurrence counts. However, the top nine ranked factors correspond to the nine factors in the true model; 10% noise does not appear to have an effect on recovery.

The scenario in Table 4 brings the true model closer to satisfying the heavy-hitters requirement. Now, all coefficients decreasing by 33%.

Table 4: Robustness Scenario 1B – Heavy-Hitters (33% Decrease).

True Model – Coefficients exhibit 33% decrease	
Coefficient	Term
1.71	$F_{29} = v_1$
1.14	$F_{98} = v_3$ & $F_{34} = v_2$
0.76	$F_{50} = v_2$
0.51	$F_{22} = v_1$
0.34	<i>INTERCEPT</i>
0.23	$F_{69} = v_1$ & $F_{23} = v_1$
0.15	$F_{10} = v_2$
0.1	$F_{82} = v_1$

In the case with no noise, Table 5 shows that the analysis now not only recovers all nine factors in the true model, but all significant factors are now found in every model; the tenth factor listed is found only twice. This sharp drop-off that delineates the ninth factor recovered from the tenth factor indicates that the tenth factor is not influential.

When 10% noise is added to the scenario the occurrence counts in the recovery are smaller than those without noise. Furthermore, three factors in the true model (F_{69} , F_{23} , F_{82}) are not recovered; they have the smallest coefficients, and are replaced by two factors that are not in the true model, F_{14} , F_7 . This scenario is affected by noise more than scenario 1A in Table 3; the noise appears to overwhelm terms with coefficients that are small in comparison to the others. Because noise depends on the range of the responses, in a model

Table 5: Robustness Scenario 1B – Recovery.

Occurrence Counts									
Rank	No Noise ($\delta = 1$)			10% Noise ($\delta = 1$)			10% Noise ($\delta = 4$)		
	Count	Factor	True	Count	Factor	True	Count	Factor	True
1	51	F_{69}	✓	58	F_{10}	✓	51	F_{82}	✓
2	51	F_{50}	✓	55	F_{29}	✓	51	F_{34}	✓
3	51	F_{34}	✓	50	F_{98}	✓	51	F_{22}	✓
4	51	F_{23}	✓	50	F_{50}	✓	50	F_{98}	✓
5	51	F_{22}	✓	50	F_{34}	✓	50	F_{69}	✓
6	51	F_{10}	✓	50	F_{22}	✓	50	F_{50}	✓
7	50	F_{98}	✓	34	F_{14}		50	F_{29}	✓
8	50	F_{82}	✓	24	F_7		50	F_{23}	✓
9	50	F_{29}	✓				44	F_{10}	✓
10	2	F_{99}					13	F_6	

that more strongly satisfies the heavy-hitters assumption the range of the responses is larger resulting in more noise in the system when taken as a percentage. Therefore, it is more likely for the noise to prevent recovery of terms with small coefficients.

To determine if higher separation improves recovery, we generate a new locating array with separation $\delta = 4$. When the locating array with higher separation is used in scenario 1B, the recovery in Table 5 ranks all nine true factors as the top nine influential factors. This is a significant improvement over the locating array with $\delta = 1$ used that resulted in only six true factors recovered. It appears that higher separation helps recover more true factors, even in the presence of noise.

We now explore a scenario that satisfies the heavy-hitters assumption even more strongly. This scenario is provided in Table 6 with coefficients decreasing by 50%.

Table 6: Robustness Scenario 1C – Heavy-Hitters (50% Decrease).

True Model – Coefficients exhibit 50% decrease	
Coefficient	Term
12.8	$F_{29} = v_1$
6.4	$F_{98} = v_3$ & $F_{34} = v_2$
3.2	$F_{50} = v_2$
1.6	$F_{22} = v_1$
0.8	<i>INTERCEPT</i>
0.4	$F_{69} = v_1$ & $F_{23} = v_1$
0.2	$F_{10} = v_2$
0.1	$F_{82} = v_1$

Table 7 presents the recovery results for the scenario given in Table 6. Just as in the previous scenario, the analysis recovers all nine significant factors in the true model when no noise is added to the system. When 10% noise is added, now only five factors in the true model are recovered. Again, the four true factors missing from the recovery, F_{69} , F_{23} , F_{10} , and F_{82} , have the smallest coefficients. When the locating array with higher separation $\delta = 4$ is used, Table 7 shows that six true factors are recovered. This is a slight improvement over the recovery by the locating array with separation $\delta = 1$. It is interesting that even though higher separation is used, the true factors are still not recovered even though the true model strongly satisfies the heavy-hitters requirement. Again, we believe this is due to our pessimistic assumption regarding in the

system (dictated by the term with the largest coefficient), resulting in the terms with smaller coefficients being overwhelmed by and therefore indistinguishable from noise.

Table 7: Robustness Scenario 1C – Recovery.

Occurrence Counts									
Rank	No Noise ($\delta = 1$)			10% Noise ($\delta = 1$)			10% Noise ($\delta = 4$)		
	Count	Factor	True	Count	Factor	True	Count	Factor	True
1	51	F_{69}	✓	68	F_{29}	✓	53	F_{34}	✓
2	51	F_{50}	✓	60	F_{50}	✓	53	F_{22}	✓
3	51	F_{34}	✓	52	F_{34}	✓	52	F_{29}	✓
4	51	F_{23}	✓	50	F_{98}	✓	51	F_{98}	✓
5	51	F_{22}	✓	50	F_{22}	✓	50	F_{50}	✓
6	51	F_{10}	✓	32	F_{92}		49	F_3	
7	50	F_{98}	✓	21	F_{42}		45	F_{78}	
8	50	F_{82}	✓	20	F_{44}		36	F_{10}	✓
9	50	F_{29}	✓	14	F_{14}		19	F_{59}	
10	2	F_{99}		13	F_{64}				

The three scenarios 1A, 1B, and 1C show an interesting trend. As the heavy-hitters assumption is more strongly satisfied in scenarios with noise, the true factors with smallest coefficients (least affecting the true model) becomes difficult to recover. This is interesting because when these same models are not affected by noise, these factors are recovered. Yet this is due to higher noise levels in true models that more strongly satisfy the heavy-hitters requirement. As a result, even small amounts of noise overwhelm the factors with small coefficients. More specifically, when a coefficient is not greater than the noise in the system (dictated by the term with the largest coefficient), then its effects are likely unrecoverable.

5 USING SEPARATION TO COPE WITH NOISE

In the previous section, the results indicate that separation is helpful in recovery. This section studies the use of separation in more detail. We construct locating arrays for 10 factors, F_1, F_2, \dots, F_{10} , each with three levels, $\{v_1, v_2, v_3\}$. The first locating array has separation $\delta = 1$ and has size 28, while the second locating array has separation $\delta = 3$ and has size 58. The true model along with the analysis parameters are given in scenario 2A in Table 8. This scenario satisfies the heavy-hitters requirement since most coefficients are well differentiated. The analysis parameters used in this section are $nModels = 25$ and $nTerms = 5$.

Table 8: Robustness Scenario 2A – Separation to Cope With Noise.

True Model	
Coefficient	Term
4.02	$F_9 = v_1$
2.49	$F_8 = v_2$ & $F_4 = v_1$
0.99	$F_5 = v_2$ & $F_2 = v_1$
0.78	$F_4 = v_1$
0.5	<i>INTERCEPT</i>

The analysis is first executed with no noise and the results are shown in Table 9. In this case, both locating arrays succeed in recovering all terms of the true model. Table 10 shows the same results for 10% noise.

Table 9: Separation $\delta = 1$ and $\delta = 3$ – No Noise.

Occurrence Counts (No Noise)						
	$\delta = 1$			$\delta = 3$		
Rank	Count	Factor	True	Count	Factor	True
1	49	F_4	✓	51	F_4	✓
2	28	F_8	✓	29	F_8	✓
3	27	F_9	✓	28	F_9	✓
4	24	F_2	✓	15	F_5	✓
5	23	F_5	✓	14	F_2	✓
6	3	F_{10}		5	F_7	

Table 10: Separation $\delta = 1$ and $\delta = 3$ – Noise at 10%.

Occurrence Counts (10% Noise)						
	$\delta = 1$			$\delta = 3$		
Rank	Count	Factor	True	Count	Factor	True
1	48	F_4	✓	47	F_4	✓
2	28	F_9	✓	28	F_9	✓
3	26	F_8	✓	28	F_8	✓
4	23	F_5	✓	26	F_5	✓
5	23	F_2	✓	26	F_2	✓
6	6	F_3		4	F_{10}	

Table 11 shows results for noise increasing from 20% to 40%. In most cases, all terms except for F_5 are recovered by each locating array. F_5 is part of an interaction in a term with a smaller coefficient and is therefore more susceptible to noise.

Now we see how much noise the locating array with higher separation can tolerate before its ability to recover begins to break down. Table 12 shows the recovery results for 50%, 60%, and 70% noise. For 50% noise, all terms of the model are recovered. A false factor (F_7) is recovered with 60% noise, but the five true factors are also recovered, though not ranked in the top five. The recovery for 70% still shows all five true factors but only three rank in the top five recovered. Two false factors (F_7 and F_3) are recovered and are ranked second and third. As expected, the true factors are ranked lower as more noise is added to the system. Furthermore, the drop-off in the occurrence counts becomes less distinct as more noise is added.

While the locating array with separation $\delta = 1$ begins to fail with around 20% noise, the locating array with separation $\delta = 3$ still recovers true model terms well through 50% noise. This supports the intuition that higher separation, at least in some cases, leads to more accurate recovery.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we present an analysis technique for screening designs based on locating arrays. These arrays are often highly unbalanced requiring the development of new analysis techniques. An orthogonal matching pursuit (OMP) based recovery relies on a heavy-hitters assumption, hence the use of a $(\bar{1}, \bar{2})$ -locating array suffices to locate a term in each iteration of the algorithm.

Our interest in this paper was a study of the robustness of the OMP-based recovery, *i.e.*, the impact of the heavy-hitters assumption under varying levels of noise in the system. When there is no noise in the system, the analysis recovers all terms of the model.

Table 11: Separation $\delta = 1$ and $\delta = 3$ – Noise at 20%, 30%, and 40%.

Occurrence Counts (20% Noise)						
	$\delta = 1$			$\delta = 3$		
Rank	Count	Factor	True	Count	Factor	True
1	57	F_4	✓	41	F_4	✓
2	34	F_8	✓	30	F_8	✓
3	26	F_9	✓	27	F_5	✓
4	26	F_2	✓	26	F_9	✓
5	6	F_1		26	F_2	✓
6				6	F_3	
Occurrence Counts (30% Noise)						
	$\delta = 1$			$\delta = 3$		
Rank	Count	Factor	True	Count	Factor	True
1	48	F_4	✓	59	F_4	✓
2	33	F_9	✓	40	F_8	✓
3	29	F_8	✓	26	F_9	✓
4	20	F_5	✓	18	F_2	✓
5	16	F_2	✓	11	F_5	✓
6	8	F_7		6	F_{10}	
Occurrence Counts (40% Noise)						
	$\delta = 1$			$\delta = 3$		
Rank	Count	Factor	True	Count	Factor	True
1	49	F_4	✓	38	F_5	✓
2	31	F_8	✓	27	F_9	✓
3	28	F_9	✓	26	F_8	✓
4	17	F_6		26	F_4	✓
5	17	F_2	✓	26	F_2	✓
6	8	F_3		10	F_6	

Table 12: Separation $\delta = 3$ – Noise at 50%, 60%, and 70%.

Occurrence Counts (50%-70% Noise)									
	50% Noise			60% Noise			70% Noise		
Rank	Count	Factor	True	Count	Factor	True	Count	Factor	True
1	34	F_9	✓	34	F_5	✓	44	F_9	✓
2	26	F_4	✓	29	F_9	✓	32	F_7	
3	24	F_5	✓	26	F_8	✓	25	F_3	
4	24	F_2	✓	26	F_4	✓	22	F_8	✓
5	23	F_8	✓	15	F_7		15	F_4	✓
6	16	F_7		14	F_2	✓	13	F_5	✓
7							11	F_2	✓
8							11	F_1	
9							2	F_{10}	

In the absence of knowledge of noise, we made a pessimistic assumption that noise depends on the largest difference in the responses. Specifically, uniform random noise is added to our output responses with the magnitude of the noise characterized as a percentage of the range of all output responses. This assumption on noise resulted affected models that more heavily satisfied the heavy-hitters requirement. Specifically, the factors with small coefficients became indistinguishable from the noise and therefore were not recoverable. Higher separation improves recovery with noise in the system but can still suffer from the same issue.

Hence our future work will study less pessimistic models for noise in the system. As well, the recovery algorithm should be augmented. At present it recovers $nTerms$ term, regardless of impact on the adjusted R^2 of the resulting model. This may impact the counting of occurrences of terms and lead to more accurate rankings of recovered terms.

Nevertheless, the results of our study provide some confidence that when our analysis technique is applied to locating array-based screening results gathered from simulated or physical systems where the true system model is unknown, such as our motivating engineered network systems, the influential factors in the system will be recovered accurately.

ACKNOWLEDGMENTS

This work is supported in part by the U.S. National Science Foundation under Grant No. 1421058.

REFERENCES

- Akaike, H. 1974, December. “A new look at the statistical model identification”. *IEEE Transactions on Automatic Control* vol. 19 (6), pp. 716–723.
- Aldaco, A. N., C. J. Colbourn, and V. R. Syrotiuk. 2015, January. “Locating Arrays: A New Experimental Design for Screening Complex Engineered Systems”. *SIGOPS Operating Systems Review* vol. 49 (1), pp. 31–40.
- Athanasiou, G., T. Korakis, O. Ercetin, and L. Tassiulas. 2009, January. “A Cross-Layer Framework for Association Control in Wireless Mesh Networks”. *IEEE Transactions on Mobile Computing* vol. 8 (1), pp. 65–80.
- Cáceres, R., and L. Iftode. 1995, June. “Improving the Performance of Reliable Transport Protocols in Mobile Computing Environments”. *IEEE Journal on Selected Areas in Communications* vol. 13 (5), pp. 850–857.
- Cammarano, A., F. L. Presti, G. Maselli, L. Pescosolido, and C. Petrioli. 2015, September. “Throughput-Optimal Cross-Layer Design for Cognitive Radio Ad Hoc Networks”. *IEEE Transactions on Parallel and Distributed Systems* vol. 26 (9), pp. 2599–2609.
- Chiang, M. and N. Rao 2012, September. “Networking and Information Technology Research and Development (NITRD) Large Scale Networking (LSN) Workshop Report on Complex Engineered Networks”.
- Colbourn, C. J., and D. W. McClary. 2008. “Locating and Detecting Arrays for Interaction Faults”. *Journal of Combinatorial Optimization* vol. 15, pp. 17–48.
- Compton, R., M. T. Mehari, C. J. Colbourn, E. De Poorter, and V. R. Syrotiuk. 2016. “Screening Interacting Factors in a Wireless Network Testbed Using Locating Arrays”. In *Proceedings of the International Workshop on Computer and Networking Experimental Research Using Testbeds (CNERT’16)*, in conjunction with *IEEE International Conference on Computer Communications (INFOCOM’16)*.

- Efazati, S., and P. Azmi. 2016, July. “Cross Layer Power Allocation For Selection Relaying and Incremental Relaying Protocols Over Single Relay Networks”. *IEEE Transactions on Wireless Communications* vol. 15 (7), pp. 4598–4610.
- Gilmour, S. 2006. “Factor Screening via Supersaturated Designs”. In *Screening: Methods for Experimentation in Industry, Drug Discovery and Genetics*, edited by A. M. Dean and S. M. Lewis, Chapter 8, pp. 169–190. Springer-Verlag.
- Hartman, A. 2005. “Software and hardware testing using combinatorial covering suites”. In *Interdisciplinary Applications of Graph Theory, Combinatorics, and Algorithms*, edited by M. C. Golumbic and I. B.-A. Hartman, pp. 237–266. Norwell, MA, Springer.
- Li, R., and D. K. J. Lin. 2003. “Analysis Methods for Supersaturated Designs: Some Comparisons”. *Journal of Data Science* (1), pp. 249–260.
- Mann, H. B., and D. R. Whitney. 1947. “On a test of whether one of two random variables is stochastically larger than the other”. *Annals of Mathematical Statistics* vol. 18 (1), pp. 50–60.
- Montgomery, D. C. 2017. *Design and Analysis of Experiments*. 9th ed. John Wiley and Sons, Inc.
- 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 International Workshop on Computer and Networking Experimental Research Using Testbeds (CNERT)*.
- Seidel, S. A., K. Sarkar, C. J. Colbourn, and V. R. Syrotiuk. 2018. “Separating Interaction Effects Using Locating and Detecting Arrays”. In *Combinatorial Algorithms [IWOC 2018], Lecture Notes in Computer Science*, edited by C. Iliopoulos, H. W. Leong, and W.-K. Sung, Volume 10979, pp. 349–360, Springer International Publishing.
- Tropp, J. A., and A. C. Gilbert. 2007, December. “Signal recovery from random measurements via orthogonal matching pursuit”. *IEEE Transactions on Information Theory* vol. 53 (12), pp. 4655–4666.
- Wang, Y., M. C. Vuran, and S. Goddard. 2012, January. “Cross-Layer Analysis of the End-to-End Delay Distribution in Wireless Sensor Networks”. *IEEE/ACM Transactions on Networking* vol. 20 (1), pp. 305–318.
- Wilcoxon, F. 1945, December. “Individual Comparisons by Ranking Methods”. *Biometrics Bulletin* vol. 1 (6), pp. 80–83.

AUTHOR BIOGRAPHIES

STEPHEN A. SEIDEL earned his M.S. in 2018 in Computer Engineering from Arizona State University. He is now employed by American Express. His e-mail address is sseidel1@asu.edu.

CHARLES J. COLBOURN earned his Ph.D. in 1980 from the University of Toronto, and is a Professor of Computer Science and Engineering at Arizona State University. He is the author of *The Combinatorics of Network Reliability* (Oxford), *Triple Systems* (Oxford), and over 350 refereed journal papers focussing on combinatorial designs and graphs with applications in networking, computing, and communications. In 2004, he was awarded the Euler Medal for Lifetime Research Achievement by the Institute for Combinatorics and its Applications. His e-mail address is charles.colbourn@asu.edu.

VIOLET R. SYROTIUK earned her Ph.D. in Computer Science from the University of Waterloo (Canada). She is an Associate Professor of Computer Science and Engineering at Arizona State University. Her research has been supported by grants from NSF, ONR, and DSTO, and numerous contracts. She serves on the editorial boards of *Computer Networks* and *Computer Communications*, as well as on the technical program and organizing committees of several major conferences. Her e-mail address is syrotiuk@asu.edu.