2017 13th IEEE Conference on Automation Science and Engineering (CASE)
Xi'an, China, August 20-23, 2017

An L1-Minimization Based Algorithm to Measure the Redundancy of State Estimators in Large Sensor Systems

Vishnu Vijayaraghavan, Kiavash Kianfar, Yu Ding, *Senior Member, IEEE*, and Hamid Parsaei

***Abstract*— Linear models have been successfully used to establish the connections between sensor measurements and system states in sensor networks. Finding the degree of redundancy for structured linear systems is proven to be NP-hard. Previously bound-and-decompose, 0-1 mixed integer programming and hybrid algorithms embedding 0-1 mixed integer feasibility checking within a bound-and-decompose framework have all been proposed and compared in the literature. In this paper, we exploit the computational efficiency of linear programs to present a novel heuristic algorithm which solves a series of l_1-norm minimization problems in a specific framework to find extremely good solutions to this problem in remarkably small runtime.**

I. INTRODUCTION

Sensor systems play an important role in studying physical phenomena which cannot be measured directly. In such systems, the sensor measurements are combined response of various source variables representing these physical phenomena. In a linear sensor system, a linear relationship of the form

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e}, \qquad (1)$$

exists between the sensor measurements $\boldsymbol{y} \in \mathbb{R}^n$ and the system states $\boldsymbol{x} \in \mathbb{R}^n$, with the design matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}, (n \gg p)$ defining this relationship and $\boldsymbol{e} \in \mathbb{R}^n$ accounting for random noise. Equation (1) is, in fact, the observation equation used in a typical linear state-space model [1]. Linear sensor systems have broad applications in various fields of engineering like manufacturing, electric power systems, signal processing, and wireless communications.

The success of a sensor system depends on its ability to estimate the vector of unknown parameters $\boldsymbol{x}$ uniquely. The vector $\boldsymbol{x}$ is commonly estimated using the least square estimator, $\hat{\beta} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{y}$. For any linear sensor system, there exists a limit to the number of sensor failures it can tolerate, beyond which the sensor might lose its capability to identify the system states altogether. The robustness of the estimator depends heavily on this limit, termed as the **degree of redundancy** [3]. This redundancy in the sensor system thus safeguards the system against sensor failures or measurement anomalies thereby aiding successful estimation of system states. The degree of redundancy is formally defined in the literature as

$$d(\boldsymbol{A}) = \min\{\delta - 1 : \exists\, \boldsymbol{A}_{(-\delta)} \text{ s.t. } r(\boldsymbol{A}_{(-\delta)}) < r(\boldsymbol{A})\}, \qquad (2)$$

where $\boldsymbol{A}_{(-\delta)}$ is the reduced matrix after deleting δ rows from the $\boldsymbol{A}$ with $r(\boldsymbol{A})$ denoting the rank of $\boldsymbol{A}$. Based on (2), we can interpret the redundancy degree of a linear system as the minimum number of sensor failures which should happen before the identifiability of any source variable is compromised. We may assume $\boldsymbol{A}$ to have full column rank.

Linear sensor systems are commonly used in many applications in wireless sensor networks [14], array signal processing [13], manufacturing fault diagnostics [2], and sensor systems in electric power systems [4]. In complex multistage assembly processes, linear sensor systems are employed to measure the deviation of locator pins enabling automatic diagnosis of manufacturing process faults. The design matrix is determined by the process design information including product geometry, and tooling layout and the sensors measurements are given in terms of dimensional derivatives from normal. In array signal processing, the location of source variables is estimated using signals received by a given set of sensors organized in patterns. The design matrix is determined by the steering for the sensors towards the source variables.

The degree of redundancy for a linear sensor system with no inherent structure is directly obtained as $d(\boldsymbol{A}) = n - p$, since any of the p row vectors of the design matrix $\boldsymbol{A}$ form a linearly independent set. But most practical systems have inherent structures due to the dependence relationship among the individual components or subsystems (clusters). For these systems, the design matrix $\boldsymbol{A}$ is usually heavily sparse, and hence the degree of redundancy is much smaller than $n - p$. Finding the degree of redundancy for linear systems with structured design matrix is known to be NP-hard [5] making this problem very hard to solve. Despite this challenge, the relevance of systems with structured model matrices in engineering systems makes this problem highly significant.

Since most sensor systems in practice are structured, we need to devise efficient methods to determine the degree of redundancy. Most of the algorithms developed so far addresses this problem using mixed integer programming and are ineffective, more so for systems with design matrices in large dimensions. In this work, we use techniques from compressed sensing and matroid theory to propose a heuristic linear programming algorithm to estimate the degree of redundancy. Our computational results show that this heuristic algorithm finds extremely reliable solutions to the degree of redundancy that are most likely optimal. Using our algorithm, we were able to obtain

All authors are with the Department of Industrial and Systems Engineering at Texas A&M University TAMU 3131, College Station, TX 77843-3131 (emails: vishnunitr@tamu.edu; kianfar@tamu.edu; yuding@tamu.edu hamid.parsaei@qatar.tamu.edu).

V. V. and H. P. are also with the Department of Mechanical Engineering at Texas A&M University at Qatar, PO Box 23874, Doha, Qatar.

This work was supported by the Qatar National Research Fund grant NPRP7-953-2-357.

978-1-5090-6781-7/17/$31.00 ©2017 IEEE

an estimated value of the degree of redundancy even for those instances which were not solved by any of the existing approaches. Moreover, our algorithm was able to attain the optimal solution for all the instances that were previously solved optimally using the existing algorithms. For some of these instances, we have reported a runtime reduction of almost 16 times as compared to the previous best reported time.

The rest of the paper is organized as follows. Section II briefly introduces the existing algorithms in the literature to find the degree of redundancy. In Section III, we address this problem using concepts from matroid theory and then briefly explains the l_1-minimization technique from compressed sensing that inspired our solution approach. In Section IV we formally introduce our algorithm 2-StageL1. We then present our computational studies in Section V and conclude with some remarks in Section VI.

II. Related work

A straightforward approach to finding the degree of redundancy is by exhaustive rank testing [3], wherein the rank of each sub-matrix $\boldsymbol{A}_{(-\delta)}$ is calculated, incrementing δ by one at each step until the rank drops. But this approach is very inefficient. As stated in the introduction, the system matrix $\boldsymbol{A}$ usually consists of various interconnected subsystems. In such cases, it is always possible to re-arrange the rows and columns of $\boldsymbol{A}$ using a transformation algorithm [3] to a bordered block diagonal (BBD) form like

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{B}_1 & & & \\ & \boldsymbol{B}_2 & & \\ & & \ddots & \\ & & & \boldsymbol{B}_r \\ \boldsymbol{S}_1 & \boldsymbol{S}_2 & \cdots & \boldsymbol{S}_r \end{bmatrix},$$

where the block $\boldsymbol{B}_t$ is a $n_t \times p_t$ matrix, and the border $\boldsymbol{S}_t$ is a $n_s \times p_t$ matrix for $t = 1, \dots, r$. We have $n_s + \sum_{t=1}^{r} n_t = n$ and $\sum_{t=1}^{r} p_t = p$. The borders usually capture the individual subsystems while the block represents their interconnections. Utilizing this BBD structure of $\boldsymbol{A}$, Cho et al. presented a bound-and-decompose algorithm [5], in which the rank testing is done on reduced matrices obtained by removing certain chosen blocks. But this algorithm is effective only for problems where $\boldsymbol{A}$ has a border block diagonal form with reasonable block sizes and narrow border.

Kianfar et al. [6] proposed a 0-1 mixed integer program (MIP) to solve this problem. Their approach is based on the fact that the redundancy degree problem can be solved by finding the minimum number of rows that if deleted from $\boldsymbol{A}$, the remaining matrix has a nonzero null space (for more details, please refer to [6]). They show that the performance of this algorithm deteriorates as the size of the problem gets larger. Recently, Bansal et al. proposed a hybrid algorithm [11] to calculate the degree of redundancy, denoted by BDMIF, which uses a 0-1 mixed integer feasibility (MIF) checking algorithm embedded within a bound-and-decompose framework. BDMIF makes it possible to simultaneously exploit both the decomposable structure of a BBD matrix $\boldsymbol{A}$ as well as the superiority of MIP over the exhaustive rank testing. Despite its distinct advantage over all the previous algorithms, finding the degree of redundancy still, continues to pose a challenge for larger and denser instances. The approach proposed in this paper is much more effective for harder problems (systems with big and dense blocks and thick borders) compared to previous approaches.

III. Redundancy degree problem as a girth problem

In this paper, we utilize concepts from the matroid theory literature to solve the redundancy degree problem by finding the smallest circuit over a vector matroid. Matroid theory started from the algebraic theory of linear independence. Let $M = (E, \mathcal{F})$ be an ordered pair consisting of a ground set E and a collection $\mathcal{F}$ of independent subsets of E. In order for M to be a matroid it should satisfy the independent augmentation axiom; i.e., if I_1 and I_2 are in $\mathcal{F}$ and $|I_1| < |I_2|$, then there exists an element e of $I_1 - I_2$ such that $I_1 \cup e \in \mathcal{F}$. Vector matroids are matroids that are defined over a matrix. For the vector matroid $M[\boldsymbol{A}^T] = (E, \mathcal{F})$ obtained from the $p \times n$ matrix $\boldsymbol{A}^T$, E will be the set of column labels of $\boldsymbol{A}^T$ with $\mathcal{F}$ being the set of subsets of E such that the subsets are linearly independent. M remains unchanged by elementary row operations. Any minimal linearly dependent set of columns of E defines a **circuit** of $M[\boldsymbol{A}^T]$. For each matroid M on the set E, there exists an associated dual matroid M^* specified on the same set E, defined usually using its basis. A **basis** of a matroid is any maximal independent set of columns in E. If we denote the collection of bases of M by $\mathcal{B}(M)$, then $\mathcal{B}(M^*) = \{E - B : B \in \mathcal{B}(M)\}$, where $\mathcal{B}(M^*)$ is the collection of all basis of M^* [12]. The **cocircuit** of a vector matroid $\boldsymbol{M}[\boldsymbol{A}^T]$ is the minimal subset of columns that when removed from $\boldsymbol{A}^T$ reduces its rank. By the definition of duality, any circuits of $M^*[\boldsymbol{A}^T]$ forms the cocircuits of $\boldsymbol{M}[\boldsymbol{A}^T]$. The terms girth and cogirth define the cardinality of the smallest circuit and the smallest cocircuit, respectively. For detailed understanding of matroid theory, please see [12].

The matrix $\boldsymbol{A}^T$ can be transformed into a reduced row echelon form (RREF) $[\boldsymbol{I}_p|\boldsymbol{D}]$ using elementary row operations, where $\boldsymbol{I}_p$ is the $p \times p$ identity matrix and $\boldsymbol{D}$ is a $p \times (n - p)$ matrix. Since M remains unchanged by these row operations, so do its circuits and cocircuits. It is a well-known result that the dual of a vector matroid over $[\boldsymbol{I}_p|\boldsymbol{D}]$, is the vector matroid over $[-\boldsymbol{D}^T|\boldsymbol{I}_{n-p}]$, where $\boldsymbol{I}_{n-p}$ is the $(n-p) \times (n-p)$ identity matrix, assuming that both $[\boldsymbol{I}_p|\boldsymbol{D}]$ and $[-\boldsymbol{D}^T|\boldsymbol{I}_{n-p}]$ have their columns labeled in the same order indexed by $E = \{c_1, c_2, \dots, c_n\}$ [12]. This result allows us to find a dual matroid for any given vector matroid.

We can now redefine our redundancy degree problem using these matroid theory concepts. It is easy to see that the redundancy degree problem on $\boldsymbol{A}$ given by (2) is simply the cogirth problem on $M[\boldsymbol{A}^T]$ and one less than the cogirth gives the optimal degree of redundancy. Now we can apply the duality concepts to find the cogirth of $M[\boldsymbol{A}^T]$ by solving a girth problem over its dual matroid $M[-\boldsymbol{D}^T|\boldsymbol{I}_{n-p}]$. More specifically, the definitions of circuit and cocircuit directly imply that a set C *is the index set of a cocircuit of* $M[\boldsymbol{I}_p|\boldsymbol{D}]$ *if and only if it is the index set of a circuit of* $M[-\boldsymbol{D}^T|\boldsymbol{I}_{n-p}]$. Thus the problem of finding $d(\boldsymbol{A})$ can now be redefined as the problem of finding the girth of $M[-\boldsymbol{D}^T|\boldsymbol{I}_{n-p}]$. Moreover, if C^* is the index set of the smallest circuit of $M[-\boldsymbol{D}^T|\boldsymbol{I}_{n-p}]$, then $r(\boldsymbol{A}[-C^*]) < r(\boldsymbol{A})$, where $\boldsymbol{A}[-C^*]$ denotes the reduced matrix obtained by removing from $\boldsymbol{A}$ all rows $\boldsymbol{a}_i, i \in C^*$.

The girth problem can be formulated as an l_0-norm minimization problem, defined as,

$$\min \|\boldsymbol{x}\|_0 \text{ s.t. } \boldsymbol{Bx} = \boldsymbol{0},\ \boldsymbol{x} \neq \boldsymbol{0}, \tag{3}$$

with $\|\boldsymbol{x}\|_0 = |supp(\boldsymbol{x})|$ being the l_0-norm, where the $supp(\boldsymbol{x}) = \{j: x_j \neq 0\}$ called the support of $\boldsymbol{x}$ with $\boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$. Solving l_0-minimization problems is NP-hard in general. An alternative approach used in compressed sensing [9] is by replacing the l_0-norm with an l_1-norm. The l_1-norm can be considered as a convex approximation of the l_0-norm. This technique is applied to a signal reconstruction problems of the form

$$\min \|\boldsymbol{z}\|_0 \text{ s.t. } \boldsymbol{y} = \boldsymbol{\Phi z}, \tag{4}$$

where $\boldsymbol{\Phi}$ is a $m \times N$ sensing matrix with $m < N$ where one tries to find the sparsest $N \times 1$ vector $\boldsymbol{z}$ as the reconstructed signal based on the measured data $\boldsymbol{y}$. The conditions under which this approach works were extensively studied in [8]-[10]. Initially, the equality of the solutions for the two methods was established only for a limited set of sensing matrices. Subsequently, Donoho in [10] proved that under the conditions that the columns of $\boldsymbol{\Phi}$ are normalized to the l_2-norm and with a uniform measure placed on $\boldsymbol{\Phi}$, the solution to (4) is unique and is equal to the solution to its corresponding l_1-norm relaxation for large m and for all $\boldsymbol{\Phi}$'s except a negligible fraction, when the number of non-zeros in $\boldsymbol{z}$ reduces below a threshold.

The value of girth gives a condition to ascertain the uniqueness of the signal reconstruction process itself. Consider two k-sparse vectors $\boldsymbol{z}$ and $\boldsymbol{z}'$ such that $\boldsymbol{z}, \boldsymbol{z}' \in S_k$, where $S_k = \{\boldsymbol{z}: \|\boldsymbol{z}\|_0 \leq k\}$. If the measurements $\boldsymbol{\Phi z}$ and $\boldsymbol{\Phi z}'$ are the same, then $\boldsymbol{z}$ and $\boldsymbol{z}'$ should be the same or else the signal reconstruction is impossible solely based on the measurements $\boldsymbol{y}$. When $\boldsymbol{\Phi z} = \boldsymbol{\Phi z}'$, $\boldsymbol{\Phi}(\boldsymbol{z} - \boldsymbol{z}') = \boldsymbol{0}$ where $\boldsymbol{z} - \boldsymbol{z}' \in S_{2k}$. If the girth of $\boldsymbol{\Phi}$ is more than $2k$, i.e., the null space of $\boldsymbol{\Phi}$ does not contain any $2k$-sparse vectors other than the zero vector, then $\boldsymbol{\Phi z} = \boldsymbol{\Phi z}'$ implies the uniqueness of the vectors $\boldsymbol{z}$ and $\boldsymbol{z}'$. To obtain the girth of $\boldsymbol{\Phi}$, a problem similar to (3), i.e., $\min\{\|\boldsymbol{z}\|_0: \boldsymbol{\Phi z} = \boldsymbol{0},\ \boldsymbol{z} \neq \boldsymbol{0}\}$, must be solved. This problem differs from the signal reconstruction problem in compressed sensing since the right-hand side vector is always zero and the set of constraints $\boldsymbol{z} \neq \boldsymbol{0}$ make this problem non-convex even with the l_1-norm relaxation. To estimate the solution of this problem, Donoho et al. in [8] proposed solving a sequence of l_1-minimization problems instead, for $t = 1, \ldots, N$:

$$(L_t) \qquad \min \|\boldsymbol{z}\|_1 \quad s.t.\ \boldsymbol{\Phi z} = \boldsymbol{0}, z_t = 1 \tag{5}$$

where $\|\boldsymbol{z}\|_1 = \sum_{t=1}^{N} |z_t|$ is the **l_1-norm**. If $\boldsymbol{z}_t^*$ denote the optimal solution of the problem L_t, then an upper bound on girth is obtained as $\min_{1 \leq t \leq N} \|\boldsymbol{z}_t^*\|_0$. The constraint $z_t = 1$ makes the null vector infeasible to (5).

In our computational experience, we have noticed although this approach leads to relatively good solutions, there are quite a number of instances where the solution from this approach can be improved by the improved heuristic algorithm we propose in this paper. The improvement in solution is particularly significant for problems with denser blocks and thick borders.

IV. A Heuristic Algorithm: 2-StageL1

In this section, we present our heuristic algorithm for the redundancy degree problem, referred to as 2-StageL1. The 2-StageL1 algorithm is as follows:

2-StageL1 Algorithm

Given a $n \times p$ design matrix $\boldsymbol{A}$, use RREF to transform $\boldsymbol{A}^T$ to $\boldsymbol{H} = [\boldsymbol{I}_p|\boldsymbol{D}]$. Let $\boldsymbol{B} = [-\boldsymbol{D}^T|\boldsymbol{I}_{n-p}]$. We assume that the columns of $\boldsymbol{A}^T$, $\boldsymbol{H}$, and $\boldsymbol{B}$ are indexed in the order, $1, \ldots, n$. Set $\boldsymbol{b}_j \leftarrow \boldsymbol{b}_j / \|\boldsymbol{b}_j\|_2$, where $\boldsymbol{b}_j$ is the jth column of $\boldsymbol{B}, j = 1, \ldots, n$.

Let $LP(k)$ define the minimization problem

$$\min \|\boldsymbol{x}\|_1 \ s.t.\ \ \boldsymbol{B}\,\boldsymbol{x} = \boldsymbol{0},\ x_k = 1$$
$$-1 \leq x_j \leq 1,\ j = 1, .., n$$

and $LP(k, l)$, $k \neq l$ define the minimization problem

$$\min \|\boldsymbol{x}\|_1 \ s.t.\ \ \boldsymbol{B}\,\boldsymbol{x} = \boldsymbol{0},\ x_k = 1,\ x_l = 0$$
$$-1 \leq x_j \leq 1,\ j = 1, .., n$$

$d^1 := n, d^2 := n,\ d^t := n$

for $1 \leq k \leq n$ **do**

Solve $LP(k)$ and set the optimal solution to $\boldsymbol{x}^t$
Set $d^t = \|\boldsymbol{x}^t\|_0$ and $C^t = supp(\boldsymbol{x}^t) \backslash k$
Let $C^t = \{k_1, k_2, \ldots, k_{d^t - 1}\}$

if $d^t < d^1$ **then**
$d^1 = d^t$ and $C^1 = supp(x^t)$
end if

if $d^1 < d^2$ **then**
$d^2 = d^1$ and $C^2 \leftarrow C^1$
end if

for $1 \leq l < d^t$ **do**

Solve $LP(k, k_l)$ and set the optimal solution to $\boldsymbol{x}^t$ and get $d^t = \|\boldsymbol{x}^t\|_0$

if $d^2 < d^t$ **then**
$d^2 = d^t$ and $C^2 = supp(\boldsymbol{x}^t)$
end if

end for

end for

end function

The degree of redundancy after Stage 1 is $d^1 - 1$
The estimated degree of redundancy $\hat{d}(\boldsymbol{A}) = d^2 - 1$

Also $r(\boldsymbol{A}[-C^2]) < r(\boldsymbol{A})$, where $\boldsymbol{A}[-C^2]$ is the resultant reduced matrix once removing the columns indexed by C^2.

In 2-StageL1, we start off by finding the dual matroid $M[-\boldsymbol{D}^T|\boldsymbol{I}_{n-p}]$ corresponding to the given vector matroid $M[\boldsymbol{A}^T]$ and then normalizing its columns such that each column has an l_2-norm equal to 1. Then we solve n l_1-norm minimization problems $LP(k)$'s which are similar to the ones proposed in [10] but with the additional constraints $-1 \leq x_j \leq 1,\ j = 1,..,n$. We call this Stage 1. For each of the solution to the optimization problem $LP(k)$ in Stage 1, we get the support set, i.e., the indices at which the solution vector is non-zero. We then solve another set of linear programs $LP(k, k_l)$'s in Stage 2, with an additional constraint forcing each of the non-zero components of the Stage 1 solution to zero. We exclude the component x_k since this is already set to 1 in the first stage. One less than the l_0-norm of the sparsest one among all the optimal solutions to $LP(k)$'s and $LP(k, k_l)$'s is taken as the estimated degree of redundancy $\hat{d}(\boldsymbol{A})$. The minimization problems $LP(k)$ and $LP(k, l)$ defined above can easily be reformulated as linear programs making this algorithm time-efficient.

The intuition behind this approach relies on the highly non-linear structure of the set of all k-sparse vectors in a given Euclidian space. Consider the n-dimensional Euclidian space $\mathbb{R}^n$ and let $\boldsymbol{e}_1, \boldsymbol{e}_2, \dots \boldsymbol{e}_n$ be the unit vectors that form a basis for this space. The set of all k-sparse vectors in $\mathbb{R}^n$ is given by $U = \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x}\|_0 \leq k\}$. This set consists of a union of $\binom{n}{k}$ linear subspaces, where each subspace is spanned by a unique choice of k out of these n unit vectors. This union operation makes the set U highly non-linear. By forcing nonzero components of a solution vector from Stage 1 to zero, we are searching for solutions with perhaps a larger l_1-norm, but hopefully a smaller l_0-norm, either within the same subspace or other subspaces within the union. Our computational experiments showed that by introducing this second stage, we are now able to find exact optimal solutions to many instances of the girth problem with known solutions which were otherwise not solved optimally by Stage 1.

V. Computational Experiments

In this section, we present the results of our computational experiments and then compare and contrast the performance of our algorithm 2-StageL1 against the integer programming based algorithms MIP and BDMIF. From now on we will refer MIP and BDMIF as exact algorithms since they find the optimal solutions to our problem. We have implemented all the algorithms in C++, and the optimization problems were solved using CPLEX 12.6.1 solver [7]. The computations were carried out using randomly generated instances of design matrices grouped into 7 categories, with each category having five instances of equal dimensions. All the instances have BBD with the instances in the same category having a similar structure. We have set a time limit of 10 hours on the runtime of each instance.

Tables I, II and II summarizes the results of our computational experiments. Each row of these tables corresponds to an instance category. The size, number of border rows and the number of blocks of each category is shown in Table I. The last column in this table shows the optimal average value of the degree of redundancy of the five instances in each category. These values were obtained from one of the exact algorithms. For instance categories that were not solved by any of the exact algorithms, the best upper and lower bounds attained are listed. The upper bounds were obtained from the best integer solution found after 10 hours. Table II summarizes the performance of the 2-StageL1 algorithm. We have reported the average runtimes over all the five instances in each category along with the estimated average value of the degree of redundancy after Stage 1 and Stage 2. The final column in this table gives the standard deviation (run time SD) of the runtimes over each category. In Table III, we have included the performance of the exact algorithms MIP and BDMIF. The average runtime along with the runtime standard deviation over each category is reported against the corresponding category. An entry with "> 10 hrs." under any category indicates that the corresponding algorithm failed to find the degree of redundancy for all the instances in that category after 10 hours of runtime.

TABLE I
CHARACTERISTICS OF INSTANCE CATEGORIES AND AVERAGE $d(\boldsymbol{A})$ OF INSTANCES IN EACH CATEGORY

Model Matrix $\boldsymbol{A}$				
No.	$n \times p$	# border rows	# blocks	Average $d(\boldsymbol{A})$
1	66 × 27	3	9	7
2	154 × 72	2	2	5
3	222 × 55	2	11	14
4	1009 × 252	1	42	17
5	2018 × 504	2	84	18
6	501 × 384	9	4	≥ 4, ≤ 34
7	650 × 350	10	6	≥ 9, ≤ 35

Category 1 instances are the smallest with size 66×27, while category 5 instances are the largest with size 2018×504. The two large instance categories, i.e., 4 and 5, have a large number of small blocks with comparatively smaller borders. On the other hand, instance categories 6 and 7 have only a few blocks which are much larger and denser than the instances in other categories along with larger borders. All instances except those in category 6 and 7 were completely solved within 10 hours by at least one of the exact algorithms. Therefore, the exact value of $d(\boldsymbol{A})$ for all instances in each of categories 1 through 5 were obtained, and the average value is reported in Table I. But our heuristic algorithm 2-StageL1, as expected, was able to solve all the instance categories well within the specified time limit. More importantly, the estimates of $d(\boldsymbol{A})$ from 2-StageL1 are also optimal for all the instances in categories 1 through 5. However, due to the unavailability of the exact values of $d(\boldsymbol{A})$ for the instances in the last two categories, we cannot confirm whether the estimates from 2-StageL1 for these instances are optimal.

The estimate of $d(\boldsymbol{A})$ reported under Stage 1 in Table II gives the solution obtained for the naive l_1-minimization approach by Donoho et al. in [8]. A comparison of Stage 1 solution with that of the Stage 2 solution (final estimated solution) of 2-StageL1 indicates the effectiveness of Stage 2 in finding improved estimates. For four of the seven instance categories, Stage 2 was able to find a better solution than the one obtained by Stage 1 and comparing the optimal solutions found by the exact algorithms in Table I, Stage 2 solutions are indeed all optimal for categories 1 to 5 and perhaps for 6 and 7 as well, while the solutions from Stage 1 are definitely not optimal for categories 1, 2, 6 and 7. With the help of this additional Stage 2 in our algorithm, we were able to significantly improve the time efficiency of finding the optimal degree of redundancy compared to exact approaches.

TABLE II
RUNTIMES OF 2-STAGEL1

No.	Average $\hat{d}(\boldsymbol{A})$		average runtime	runtime SD
	Stage 1	Stage 2		
1	7.4	7	7.491 sec.	0.31 sec.
2	6	5	8.787 sec.	0.23 sec.
3	14	14	14.225 sec.	0.53 sec.
4	17	17	2.684 min.	0.26 min.
5	18	18	7.716 min.	0.28 min.
6	35	33	3.983 min.	0.64 min.
7	36	34	5.027 min.	0.52 min.

TABLE III
RUNTIMES OF MIP AND BDMIF

No.	MIP		BDMIF	
	average run time	runtime SD	average runtime	runtime SD
1	18.49 sec.	4.23 sec.	46.82 sec.	3.44 sec.
2	2.96 sec.	3.59 sec.	21.73 sec.	4.65 sec.
3	47.95 sec.	4.17 sec.	2.05 min.	0.08 min.
4	> 10 hrs.		21.17 min.	4.59 min.
5	> 10 hrs.		1.66 hrs.	0.04 hrs.
6	> 10 hrs.		> 10 hrs.	
7	> 10 hrs.		> 10 hrs.	

BDMIF performs significantly better than MIP for large instances, though for smaller instances MIP has a slight advantage over BDMIF. 2-StageL1 reports far superior runtimes as compared to both these exact algorithms. None of the instances in categories 6 or 7 were solved by any of the exact algorithms. These instances have much thicker blocks with a large border. But 2-StageL1 solves category 6 instances within an average runtime of around 4 minutes and category 7 instances within an average runtime of around 5 minutes giving the estimates 33 and 34 respectively for the degree of redundancy. Hence 2-StageL1 is a highly reliable alternative approach for finding the degree of redundancy for such instances. Stage 2 estimates were two less (the most improvement we have observed among categories) than the Stage 1 estimates for all these instances, highlighting the significance of Stage 2. The runtime of 2-StageL1 does not depend much on the structure of the design matrix but rather on the number of linear programs solved in each case. In fact, all the instances that we have tested were solved by 2-StageL1 within a few minutes.

VI. CONCLUDING REMARKS

We started this paper by establishing the equivalence between redundancy degree problem and the girth problem in matroid theory. Exploring our options to solve the latter problem took us to l_1-minimization approach, a successful technique applied to similar problems in compressed sensing. Building on this idea, we proposed a new heuristic algorithm that finds extremely good near-optimal solutions to the degree of redundancy. This algorithm gives highly reliable estimates to the degree of redundancy in remarkably faster time than all the exact algorithms. The l_1-minimization technique thus takes us a step closer to the long-held goal of finding the degree of redundancy of instances with large blocks and borders, but the challenge of solving such instances to optimality remains.

REFERENCES

[1] W. J. Rugh, Linear System Theory, Prentice-Hall, Upper Saddle River, NJ, 1996.

[2] Y. Ding, D. Ceglarek, and J. Shi, "Fault diagnostics of multistage manufacturing process by using state space approach", Journal of Manufacturing Science and Engineering, Vol. 124, 313-322, 2002.

[3] J.J. Cho, Y. Chen, and Y. Ding, "Calculating the breakdown point condition of sparse linear models", Technometrics, Vol. 51, 34-46, 2009.

[4] L. Mili, M.G. Cheniae, and P.J. Rousseeuw, "Robust state estimation of electric power systems", IEEE Transactions on Circuits and Systems, Vol. 41, 349-358, 1994.

[5] J.J. Cho, Y. Chen, and Y. Ding "On the (co)girth of connected matroids", Discrete Applied Mathematics, Vol. 155, 2456 – 2470, 2007.

[6] K. Kianfar, A. Pourhabib, and Y. Ding "An Integer Programming Approach for Analyzing the Measurement Redundancy in Structured Linear Systems", IEEE Transactions on Automation Science and Engineering, Vol. 8, 447-450, 2011.

[7] IBM ILOG CPLEX 12.6.1 documentation, available at: https://www.ibm.com/support/knowledgecenter/en/SSSA5P_12.6.1/ilog.odms.studio.help/Optimization_Studio/topics/COS_home.html.

[8] D.L. Donoho, M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via l_1 minimization", PNAS, Vol. 100, 2003.

[9] S. Foucart, H. Rauhut, A Mathematical Introduction to Compressed Sensing, Springer, New York, 2013.

[10] D.L. Donoho, "For most large underdetermined systems of linear equations the minimal $l_1$norm solution is also the sparsest solution", Communications on Pure and Applied Mathematics, Vol. 109, 797-829, 2006.

[11] M. Bansal, K. Kianfar, Y. Ding, E. Moreno-Centeno, "Hybridization of Bound-and-Decompose and Mixed Integer Feasibility checking to measure redundancy in structured linear systems," IEEE Transactions on Automation Science and Engineering, Vol. 10, 1151-1157, 2013.

[12] J.G. Oxley, Matroid Theory, Oxford University Press, New York, 1992.

[13] D. H. Johnson and D.E. Dudgeon, Array signal processing: concepts and techniques, Prentice Hall, New Jersey, 1993.

[14] J.J. Cho, Y. Ding, Y. Chen, and J. Tang, "Robust calibration for localization in clustered wireless sensor networks," IEEE Transactions on Automation Science and Engineering, Vol. 7, 81-95, 2010.