STRUCTURE-AWARE RELIABILITY ANALYSIS OF LARGE-SCALE LINEAR SENSOR SYSTEMS

A Dissertation

by

VISHNU VIJAYARAGHAVAN

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Kiavash Kianfar
Committee Members,	Yu Ding
	Anxiao Jiang
	Hamid Parsaei
Head of Department,	Lewis Ntaimo

May 2020

Major Subject: Industrial Engineering

ABSTRACT

A linear sensor system is a system in which the sensor measurements have a linear relationship to the source variables that cannot be measured directly. Linear sensor systems are widely deployed in advanced manufacturing processes, wireless transportation systems, electrical grid systems, and oil and gas pipeline systems to monitor and control various physical phenomena critical to the smooth function of such systems. The source variables capture these complex physical phenomena which are then estimated based on the sensor measurements. Two of the critical parameters to be considered while modeling any linear sensor system are the degree of redundancy and reliability. The degree of redundancy is the minimum number of sensor failures that a system can withstand without compromising the identifiability of any source variables. The reliability of a sensor system is a probabilistic evaluation of the ability of a system to tolerate sensor failures. Unfortunately, the existing approaches to compute the degree of redundancy and estimate the reliability are limited in scope due to their inability to solve problems in large-scale.

In this research, we establish a new knowledge base for computing the degree of redundancy and estimating the reliability of large-scale linear sensor systems. We first introduce a heuristic convex optimization algorithm that uses techniques from compressed sensing to find highly reliable approximate values for the degree of redundancy.

Due to the distributed nature of linear sensor systems often deployed in practical applications, many of these systems embed certain structures. In our second approach, we study these structural properties in detail utilizing matroid theory concepts of connectivity and duality and propose decomposition theorems to disintegrate the redundancy degree problem into subproblems over smaller subsystems. We solve these subproblems using mixed integer programming to obtain the degree of redundancy of the overall system. We further extend these decomposition theorems to help with dividing the reliability evaluation problem into smaller subproblems. Finally, we estimate the reliability of the linear sensor system by solving these subproblems employing mixed integer programming embedded within a recursive variance reduction framework, a technique commonly used in network reliability literature.

We implement and test developed algorithms using a wide range of standard test instances that simulate real-life applications of linear sensor systems. Our computational studies prove that the proposed algorithms are significantly faster than the existing ones. Moreover, the variance of our reliability estimate is significantly lower than the previous estimates.

DEDICATION

To my grandparents

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my advisor, Dr. Kiavash Kianfar, for his continuous support during this research, for his motivation, enthusiasm, and immense knowledge. The good advice, support, and friendship of Dr. Hamid Parsaei has been invaluable on both an academic and a personal level, for whom I am extremely grateful. I would also like to thank my committee members, Dr. Yu Ding and Dr. Anxiao Jiang, for their support, suggestions, and encouragement. I owe my deepest gratitude to my parents, brothers, and sister-in-laws who always supported and believed in me in all my endeavors, let it be personal or professional.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a thesis (or) dissertation committee consisting of Professors Kiavash Kianfar (advisor), Yu Ding, and Hamid Parsaei of the Department of Industrial and Systems Engineering and Professor Anxiao Jiang of the Department of Computer Science and Engineering.

All work conducted for the thesis (or) dissertation was completed by the student independently.

Funding Sources

This work was supported by the Qatar National Research Fund grant NPRP7-953-2-357 and The Department of Industrial and Systems Engineering, Texas A&M Univeristy.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

TABLE Page

1. INTRODUCTION

1.1 Linear Sensor Systems and Applications

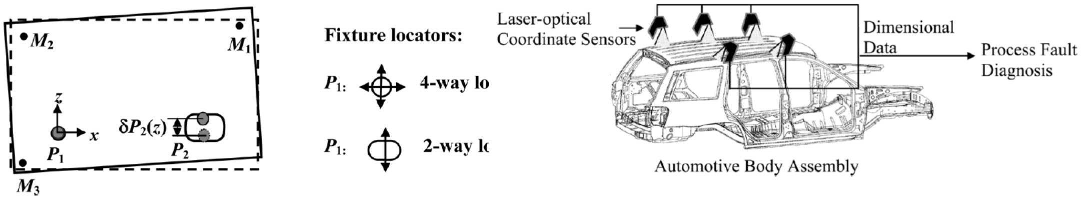
Nowadays, inexpensive smart devices with multiple heterogeneous on-board sensors, networked through wired or wireless links, are distributed in large numbers throughout a physical process or a physical environment, providing real-time measurements, thereby enabling surveillance, monitoring, and fault detection capabilities that could not be imagined a decade ago. Such a system-wide deployment of sensing devices is known as distributed sensing and is considered one of the top emerging technologies today. These high-resolution systems, if properly constructed, can play a crucial role in managing and executing many critical industry-driven activities. In any of these systems, the source variables, whose values trigger various actions, are estimated based on the measurements gathered by the multitude of sensor nodes that monitor and control them. Malfunction of these systems typically result in enormous economic losses and sometimes even endanger critical infrastructure and human lives.

In a *linear sensor system*, a linear model of the form

$$\boldsymbol{u} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e} \tag{1.1}$$

establishes the connection between the *sensor measurements* $\boldsymbol{u} \in \mathbb{R}^n$ and the unknown *source variables* $\boldsymbol{x} \in \mathbb{R}^m$ that cannot be measured directly. The $n \times m$ matrix $\boldsymbol{A}$ called the design matrix captures this linear relationship. We assumed $\boldsymbol{A}$ to be of full column rank, i.e., $r(\boldsymbol{A}) = m$, where $r(.)$ denotes the rank function. The error term $\boldsymbol{e} \in \mathbb{R}^n$ accounts for the sensor noise and is considered to be normally distributed with mean $\mathbf{0}$ and covariance matrix $\sigma^2\boldsymbol{I}$.

Applications of linear sensor systems include diagnosing the process faults in panel assembly processes [4] and multi-stage manufacturing processes [5, 6], estimating the location of source variables in array signal processing [7], improving the stability of electrical power systems, and calibration of wireless sensor systems [8] among others.



(a) Fixture setup for a 2-D workpiece

(b) Optical coordinate measuring machine for automotive body assembly

Figure 1.1: An Application of Linear Sensor System (Reprinted from [1])

In a multi-stage manufacturing process, linear sensor systems are used to identify the process faults via source variables that capture deviations of locating pins during an assembly process. In any such assembly process, the dimensional quality of the finished product is highly dependent on the level of accuracy with which the parts are fixtured during the assembly. No matter how good the fixture design is and how well laid out the locating pins and blocks are, over time the locating elements may be worn, loose, or bent impacting the ability of fixture locators to position the parts accurately during assembly. The impact of this could be a severe deterioration in the capability of the fixture to accurately locate the panel (fixture faults) compromising the dimensional integrity of the finished product [9]. Directly measuring the locating position of the locators during the production process is costly, if not impossible [10]. With the increasing data collection capability of coordinate sensor systems, recent researches have been focusing on utilizing the coordinate sensing data to quickly identify root causes of product or process variations [1].

We can systematically and quickly identify fixture faults by developing a mathematical model off-line from the geometric information about the product and fixtures so that manufacturing downtime is reduced and product quality is improved. To illustrate the model conception, let us consider a simplified panel assembly process as an example. Figure 1.1 (a), presented in [1], shows a 2-D workpiece (a panel) held by a fixture consisting of a four-way locating pin (P_1) and a two-way locating pin (P_2). While P_1 constrains the part motion in both the x- and z- directions, P_2 constrains the part motion in the z- direction. A deviation $\delta P_2(z)$ of locating pin P_2 causes deviation of the coordinate measurement data at points M_1, M_2, and M_3. Although the exact relationship between the deviations of the locating pins and the coordinate measurements is nonlinear, the higher-order terms can be neglected considering the pin displacements are small relative to the overall dimension of the panel. Hence, based on the product and fixture geometry, a linear model can be constructed to capture this relationship, and a diagnostic algorithm based on the least squares estimation theory can then be used to automatically detect fixture faults.

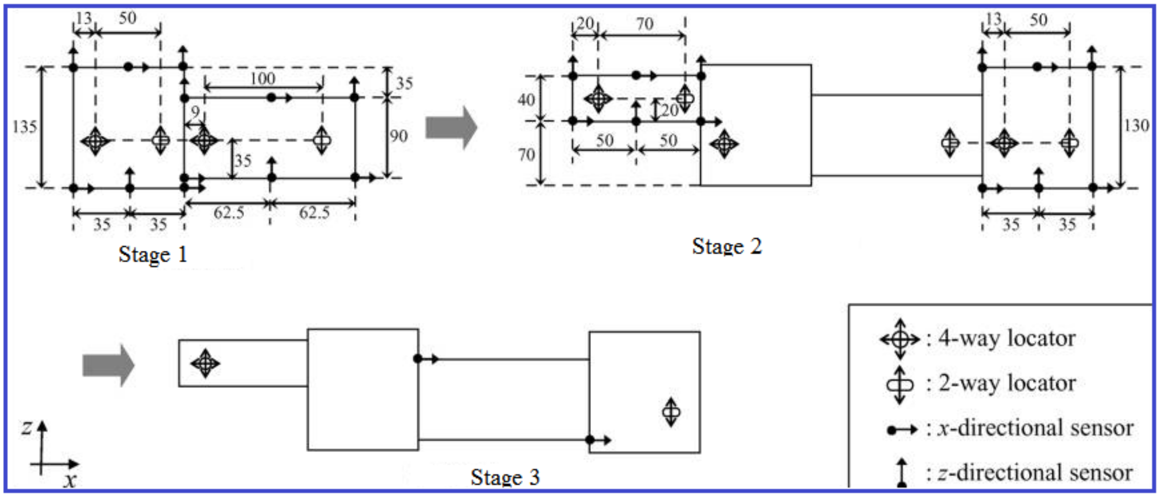


Figure 1.2: A Three Station Assembly Process (Modified from [2])

The automotive body assembly process is an example of a panel assembly process. An in-line Optical Coordinate Measuring Machine (OCMM) used in the quality inspection of automotive body parts is illustrated in Figure 1.1 (b). Quality inspections are commonplace in most assembly processes. To reach high product quality each finished product or sub-assembly is subject to quality inspection, wherein the deviations of product dimensions from the nominal values are measured using coordinate sensors. These coordinate measurement data can be used to identify fixture faults by developing a linear model (1.1) that captures the linear relationship between these measurements $\boldsymbol{x}$ and the deviations due to locating pin errors $\boldsymbol{u}$ (fixture faults). To implement this model the design matrix $\boldsymbol{A}$ needs to be constructed off-line as discussed above.

Most large-scale manufacturing assembly processes happen in various stages. Figure 1.2, from [2], shows such a typical multi-station assembly process consisting of three stages. At Stage 1 two parts are assembled, and the resulting sub-assembly is transferred to Stage 2, where it is further assembled with two more parts. The final assembly is inspected for quality at Stage 3. Coordinate sensors are placed at all the stages to measure the dimensional deviation of parts. With the assembly process happening in multiple states, the design matrix that captures the linear relationship between the coordinate measurements and the fixture faults usually exhibit a bordered block diagonal structure (Explained in detail in Section 2.1). We call such linear sensor systems structured.

1.2 The Degree of Redundancy

Various statistical fixture fault diagnosis algorithms using coordinate sensor systems have been developed in the literature for multi-stage manufacturing processes [11, 12, 13]. In most of these algorithms, the source variables representing the fixture faults are commonly estimated using some linear regression estimators like the Least Squares (LS) estimator. The LS estimate, denoted by $\hat{\boldsymbol{x}}_{LS}$, is given by

$$\hat{\boldsymbol{x}}_{LS} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{u}.$$

Sensor failures are a cause of major concern during any statistical estimation process. Take, for example, the fault diagnosis system in a multi-stage assembly process. If a sensor cannot work properly, the estimated fixture faults based on the linear model can have an uncertainty much larger than the design specification causing false alarms and misdetection of faults eventually compromising the user's confidence in the sensor system.

To obtain statistically accurate estimates for the source variables after accounting for sensor failures, modeling errors in the design matrix $\boldsymbol{A}$ and errors due to sensor measurement noises, linear sensor systems are often designed to have more number of sensor measurements than the number of source variables to be estimated. With more measurements than the source variables, one can always obtain a lower bound on the number of measurements necessary for uniquely estimating $\boldsymbol{x}$. Loosely speaking, the number of measurements beyond this lower bound is termed the degree of redundancy of the linear sensor system.

Definition 1 (Degree or Redundancy [2, 14])**.** *The degree of redundancy of $\boldsymbol{A}$, denoted by $d(\boldsymbol{A})$ is given by*

$$d(\boldsymbol{A}) = \mathit{min}\{d-1 : \exists\ \boldsymbol{A}_{(-d)}\ \textit{s.t.}\ r(\boldsymbol{A}_{(-d)}) < r(\boldsymbol{A})\}, \tag{1.2}$$

where $\boldsymbol{A}_{(-d)}$ is the matrix obtained after deleting some d rows from $\boldsymbol{A}$.

When sensors corresponding to the rows indexed by d for which $r(\boldsymbol{A}_{(-d)}) < r(\boldsymbol{A})$ fail catastrophically, i.e., when no useful information is obtained from these sensors, we run into an ill-posed system with no unique estimation for source variables. The degree of redundancy is thus a measure of the sensitivity of a linear system towards sensor failures. Therefore, it is critical to estimate the degree of redundancy of a linear sensor system during its design phase. The value of degree or redundancy is also essential in developing robust linear estimators. Researchers have been inspired to develop robust regression estimators due to the high sensitivity of traditional LS estimators to sensor failures and measurement anomalies [15]. The study of robust statistical estimation remains a hotbed of research in computational geometry.

We now briefly explain the basics of robust estimation. In robust statistics, the term finite-sample *breakdown point* quantifies the robustness of a regression method. Let

$$\mathcal{Z} = \{(\boldsymbol{a}^1, u_1), ..., (\boldsymbol{a}^n, u_n)\}$$

denote the collection of known information contained in $\boldsymbol{A}$ and $\boldsymbol{u}$. Note that, $\boldsymbol{a}^i$ defines the i^{th} row of $\boldsymbol{A}$. Let $T(\mathcal{Z})$ be a regression estimator over the data set $\mathcal{Z}$. Suppose some k data points of $\mathcal{Z}$ are contaminated such that no useful information can be obtained from these points. Let us denote this contaminated data set by $\mathcal{Z}'_k$. Then, $T(\mathcal{Z}'_k)$ denotes the regression estimator over $\mathcal{Z}'_k$. The maximum difference between $T(\mathcal{Z})$ and $T(\mathcal{Z}'_k)$ is denoted by $bias(k;T)$, defined as

$$bias(k;T) = \underset{\mathcal{Z}'_k}{sup} \|T(\mathcal{Z}) - T(\mathcal{Z}'_k)\|,$$

where the supremum is over all possible $\mathcal{Z}'_k$ for a given k and $||.||$ represents the l_2 norm [16, 17, 2].

The breakdown point of the estimator T, denoted by $\epsilon^*_n(T, \mathcal{Z})$, is defined as

$$\epsilon^*_n(T, \mathcal{Z}) = min\left\{\frac{k}{n} : bias(k; T)\ is\ infinite\right\}.$$

Informally, the breakdown point defines the minimal fraction of gross outliers or missing sensor measurements that could cause an estimator to give completely wrong results [18, 16]. The higher the breakdown point, the more sensor failures an estimator can tolerate, and hence the more robust it is. So, designing estimators with a large breakdown point is fundamental to any robust estimation process. For a linear sensor system, the breakdown point not only depends on the estimator but also on the linear dependence relationship among the rows of the design matrix $\boldsymbol{A}$. Mili and Coakley, in [17], provided an upper bound for the breakdown point of such a system. They proved that the maximum breakdown point that can be attained by any linear regression estimator, represented as $\epsilon^*_{\text{max},n}$, is

$$\epsilon^*_{\text{max},n} = \frac{\lfloor d(\boldsymbol{A})/2 \rfloor + 1}{n},$$

where $\lfloor a \rfloor$ denotes the largest integer $\leq a$.

A well-studied robust linear regression estimator is the *Least Trimmed Squares (*LTS*) estimator* developed by Rousseeuw in 1984 [19]. This estimator is much less sensitive to sensor failures and measurement errors compared to the classic LS estimator [20]. An LTS estimator is determined through

$$\min \sum_{i=1}^{h} w^2_{(i)},$$

where h is an integer trimming parameter, and $w^2_{(i)} = (\boldsymbol{u}_i - \boldsymbol{a}^i\hat{\boldsymbol{x}})^2,\ i = 1, ..., n$, are the squared residuals such that $w^2_{(1)} \leq w^2_{(2)} \leq ... \leq w^2_{(n)}$. This estimator can attain the maximum breakdown point, $\epsilon^*_{\text{max},n}$, if we choose the trimming parameter h ideally. More specifically, the maximum breakdown point can be achieved by determining the parameter h such that

$$h_L \leq h \leq h_U,$$

where $h_L = \lfloor n - d(\boldsymbol{A})/2 \rfloor$ and $h_U = \lfloor n - (d(\boldsymbol{A}) - 1)/2 \rfloor$ [17]. The knowledge of $d(\boldsymbol{A})$ is essential in tuning this design parameter. Thus, finding the degree of redundancy is extremely important in designing robust linear sensor systems.

If any m row vectors of $\boldsymbol{A}$ are linearly independent, we can directly find the degree of redundancy as $n - m$. However, when there exists some intrinsic structure in the system, this value is much smaller than $n - m$. Unfortunately, most linear sensor systems in practice inherit specific structure and are often modeled by large design matrices. For general linear sensor systems, finding the degree of redundancy is proven to be $\mathcal{NP}$-hard [21, 22]. This is because the minimum distance problem in binary coding, which was proved to be $\mathcal{NP}$-hard by Vardy [21], can be reduced to a special case of finding the redundancy degree in a linear model [22]. This implies that the existence of a polynomial-time solution method to find the redundancy degree in a general sensor system is highly unlikely.

The complexity of the problem implies that the existence of a polynomial-time solution method to find the redundancy degree in a general sensor system is highly unlikely. However, perhaps because of the difficulty of the problem, previous work reporting computable algorithms to find the redundancy degree is scarce, with a few exceptions. The only known algorithms to solve this problem are a bound and decompose technique [22], a mixed integer programming (MIP) formulation [14], and a hybrid algorithm that integrates the MIP scheme within the bound and decompose framework [30]. However, none of these algorithms are applicable in large-scale problems due to the general intractability of MIPs and the inefficiency of the bound and decompose approach.

In our research, we developed algorithms to find the degree of redundancy of large-scale linear sensor systems, with an emphasis on solving for systems that exhibit a bordered block diagonal structure (detailed in 2.1). We implemented and tested all the proposed algorithm over a wide range of test instances inspired by real-life applications of linear sensor systems. Our main contributions on the degree of redundancy problem can be summarized as follows:

- First, we propose a *heuristic convex optimization algorithm* to compute approximate values of the degree of redundancy. Our heuristic algorithm, denoted as 2-STAGEL1, utilizes compressed sensing fundamentals of sparse representation of signals to formulate a series of l_1*-norm minimization* problems in a specific framework reformulated as linear programs to find extremely good solutions to the degree of redundancy problem (Section 3.1).

- Our computational experiments show that the average running time of 2-STAGEL1 was $\sim$0.2 times that of the fastest existing algorithm in the literature. Moreover, for over 95% of the tested instances, the degree of redundancy value from 2-STAGEL1 was also the optimal solution (Section 7.1).

- We get the solution from 2-STAGEL1 to establish a starting solution (warm start solution) to the existing MIP formulation [14] (denoted as MIP_{CO}) and use *warm start* options in CPLEX 12.9 solver to boost the running time of MIP_{CO}. The overall running time of the resultant algorithm was $\sim$10% less than that of MIP_{CO} (Section 3.2).

- To find the optimal solution to the degree of redundancy problem for large-scale structured linear sensor systems, we utilize the equivalence of the degree of redundancy of the design matrix $\boldsymbol{A}$ and the *cogirth* (explained in Section 2.2) of the vector matroid $M[\boldsymbol{A}^T]$ defined over the columns of $\boldsymbol{A}^T$. We exploit the structural properties of linear sensor systems to derive a *decomposition theorem for the cogirth* problem based on the properties of connectivity of matroids (Section 4.1).

- We introduce a decomposition algorithm, referred to as DMIP_{CO}, that uses the decomposition theorem to break the cogirth problem into subproblems and then solve these subproblems using MIP_{CO} (The algorithm MIP_{CO} finds the cogirth of a given vector matroid) to obtain optimal solution to the degree of redundancy problem (Section 4.2).

- We establish the equivalence of the cogirth problem over $M[\boldsymbol{A}^T]$ to the *girth* problem over its dual matroid, and then develop a dual *decomposition theorem for the girth problem* utilizing the properties of connectivity and duality of matroids (Section 5.1). We then formulate an MIP, denoted as MIP_{CIR}, to solve the girth problem. With the help of the dual decomposition theorem, we disintegrate the girth problem into smaller subproblems and solve these subproblems with MIP_{CIR} to compute the degree of redundancy. We refer to this dual decomposition algorithm as DMIP_{CIR} (Section 5.2).

- We present a detailed computational analysis of our decomposition algorithms by comparing their performance against the existing algorithms. Our algorithms report a running time reduction of more than 50% for ∼75% of the tested instances with an average decrease of more than 80% for some of the large instances (matrices with > 500 rows) compared to the best running times reported by the other algorithms (Section 7.1).

1.3 The Reliability

The reliability is another important parameter that needs to be considered while designing a linear sensor system. The reliability of a linear sensor system is the probabilistic evaluation of the ability of the system to estimate source variables with a desired level of statistical efficiency sustaining sensor failures. In this dissertation, we study the reliability of a linear sensor system under catastrophic sensor failure. Under this assumption, each sensor is assumed to take only two states, functional or failed. To define the reliability, we consider a $(k$ out of $n)-$system, a well-studied system in the literature, which comprises an n component system that works only if at least k of the n components work [23].

Definition 2 (Reliability [10])**.** *Consider a $(k$ out of $n)-$linear sensor system with a design matrix $\boldsymbol{A}$ and having n sensors. Assume the sensors fail (work) independently of each other with known probabilities. Then, the reliability of the system is the probability that:*

(a) $r(\boldsymbol{A}_{(-d)}) = m$ where d denotes the row indices of failed sensors, and

(b) the number of rows in $\boldsymbol{A}_{(-d)}$ is greater than or equal to k, where $k \geq m$.

The system is said to work, if conditions (a) and (b) in Definition 2 are met. While condition (a) guarantees a unique state estimation under catastrophic sensor failure, (b) ensures that a desired level of statistical efficiency is achieved during the estimation process. Unfortunately, computing the reliability of a general linear sensor system belongs to a class of $\#\mathcal{P}$-complete problems, a family of $\mathcal{NP}$-hard problems not known to be in $\mathcal{NP}$ [10]. Estimation approaches has been used in the literature to solve this problem.

A naive approach to estimate the reliability is to use the crude Monte Carlo method. This approach is highly impractical and inefficient. In the network reliability literature, improved Monte Carlo methods have been developed to increase the efficiency of system reliability evaluation, among which the Recursive Variance Reduction (RVR) method offers the best performance [24, 25]. In its essence, the RVR method iteratively reduces the sample space of all possible sensor states by obtaining combinations of states that guarantee either a system failure or a success and then apply the crude Monte Carlo method over these reduced spaces. Yang and Chen [3] applied the RVR technique to estimate the reliability of linear sensor systems. In their algorithm, which we denote by RVR_{RREF}, the reduction in the sample space is achieved by finding the minimal cut sets (defined in Section 2.4) of the linear sensor system [10, 3]. However, the approach used in RVR_{RREF} to find the minimal cut sets is a row reduction method which is unsuitable for systems with large design matrices. The inefficiency of this rank reduction method limits the scope of reliability estimation problems that can be solved using RVR_{RREF}.

Estimating the reliability of a linear sensor system is a crucial problem in data analytics with a direct impact on the design and operation of such systems. As with the degree of redundancy problem, our algorithms to estimate the reliability focuses on solving for linear sensor systems that exhibit bordered block diagonal structure. To study the performance our algorithms, we implemented and tested them over numerous linear sensor systems with structured design matrices developed based on practical applications of such systems and compared then against the existing RVR_{RREF} algorithm proposed in [10]. The contributions on the reliability estimation problem of this dissertation can be summarized as follows:

- We extend our decomposition theorem for cogirth to derive another decomposition theorem that proposes a strategy to disintegrate the problem of finding all cocircuits (detailed in Section 2.2) over the matroid $M[\boldsymbol{A}^T]$. We call this theorem, the *decomposition theorem for* *cocircuits* (Section 6.1).

- We re-formulate the existing MIP$_{\text{CO}}$ algorithm for cogirth to find the minimal cut set of a linear sensor system for which cetain sensor states are fixed as working or failed. We then follow the RVR framework developed in [25, 3], and develop an algorithm, referred to as RVR$_{\text{CO}}$, that decomposes the re-formulated MIP$_{\text{CO}}$ algorithm into smaller subproblems utilizing our decomposition theorem for cocircuits. RVR$_{\text{CO}}$ iteratively reduces the sensor state space by obtaining minimal cut sets by solving these decomposed MIPs and then generates samples over these reduced spaces to estimate the reliability.

- We propose a decomposition theorem to find all circuits over the dual matroid of $M[\boldsymbol{A}^T]$ by extending the decomposition theorem for girth. We call this dual theorem, the *decomposition theorem for circuits* (Section 6.2).

- We present a dual decomposition algorithm, denoted by RVR$_{\text{CIR}}$, that embeds a re-formulated MIP$_{\text{CIR}}$ algorithm within a decomposition framework based on the decomposition theorem for circuits and then integrates this within the RVR framework to find minimal cut sets and iteratively reduce the sensor state space thereby estimating the reliability of the given system.

- We show that, with a sample size of 1,000,000, our reliability evaluation algorithms improve over the RVR$_{\text{RREF}}$ algorithm with a reduction in the variance of the estimated system reliability by an average of $\sim$120 and an average decrease in the running time of $\sim$35% for the tested instances (Section 7.2).

1.4 Dissertation Structure

The dissertation is organized as follows: In Chapter 2, we present a brief review of structured linear sensor systems, matroid theory, and the existing algorithms for the degree of redundancy and the reliability evaluation problems to the extent required for the results in this dissertation. We introduce our heuristic convex optimization algorithm to to solve the degree of redundancy problem in Chapter 3. In Chapters 4 and 5, we present our cocircuit and circuit based decomposition algorithms, respectively, to find the degree of redundancy utilizing the structural properties of linear sensor systems. We present our algorithms to estimate the reliability of linear sensor systems in Chapter 6. The focus of Chapter 7 is on the detailed results of our computational experiments studying the performance of all the proposed algorithms. We conclude the dissertation in Chapter 8 along with some future research plans.

2. NECESSARY BACKGROUND

2.1 Structured Linear Sensor Systems

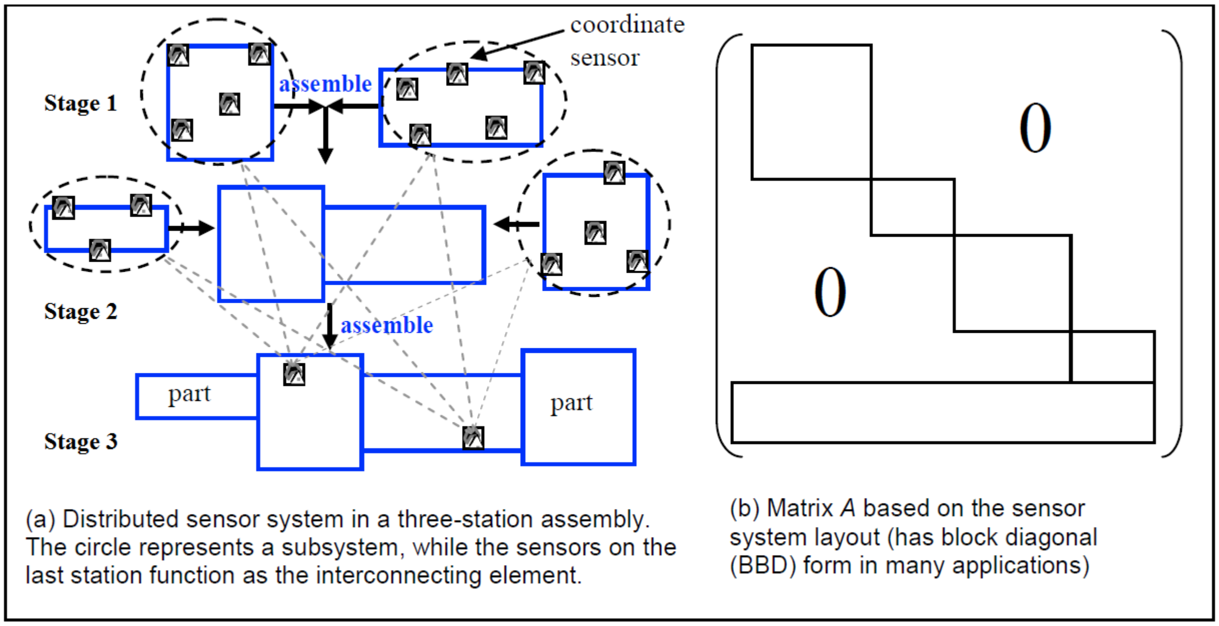
Structured linear systems are pervasive in engineering applications. In many of these applications, the design matrix is not only structured but also relatively sparse (meaning that it has many zero entries) [2] making its rows highly dependent. For a structured linear system, the design matrix $\boldsymbol{A}$ commonly manifests a *bordered block diagonal* (BBD) structure given by

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

where the nonzero submatrices $\boldsymbol{A}_i,\ i = 1, ..., r$ form the blocks, and the submatrices $\boldsymbol{S}_i,\ i = 1, ..., r$ represent the border rows. Each block submatrix $\boldsymbol{A}_i$ is of size $n_i \times m_i$, and border submatrix $\boldsymbol{S}_i$ is of size $n_s \times m_i$, where n_s is the number of border rows.

Many structured linear systems, especially those comprising of subsystems or clusters linked through an interconnecting element are likely to have a BBD design matrix. The design matrices used in fault diagnosis of multi-stage assembly processes and robust calibration for localization in clustered wireless sensor systems are all examples of systems with BBD structures. The physical interpretation is that the blocks represent the subsystems (or clusters) and the border rows represent the interconnecting elements (or between-cluster links); see Figure 2.1 for illustrations.

In many linear sensor systems, the BBD form of the design matrix may not be readily apparent even if it is inherently a BBD matrix. In [2], a graph theory based procedure is presented to transform such design matrices into a BBD form, given the matrix is sufficiently sparse. For the



(a) Distributed sensor system in a three-station assembly. The circle represents a subsystem, while the sensors on the last station function as the interconnecting element.

(b) Matrix A based on the sensor system layout (has block diagonal (BBD) form in many applications)

Figure 2.1: Relationship Between a Structured System and the BBD Form Matrix

scope of this dissertation, we assume that the design matrices are already expressed in the BBD form.

Before moving ahead, let us introduce some of the notations that we use throughout this dissertation. We denote by $\boldsymbol{A}[U]$, the reduced matrix obtained by removing from $\boldsymbol{A}$ those rows and columns containing $\boldsymbol{A}_t$, for $t \in \{1, 2, ..., r\} \setminus U$. If $|U| = t$, then we call $\boldsymbol{A}[U]$ a t-block submatrix of $\boldsymbol{A}$. We also assume that $\boldsymbol{A}[U]$ is of size $n_u \times m_u$. Also, by $\boldsymbol{A}_{(D)}$, we denote the reduced matrix from $\boldsymbol{A}$ after removing rows which are not indexed by D, and by $\boldsymbol{A}_{(-D)}$, the reduced matrix after removing rows indexed by D.

2.2 Matroids: A Brief Review

In this section, we present a brief review of matroids. For more detailed understanding of matroids, please refer [26]. Matroids are combinatorial structures that generalizes the notion of linear dependence in a vector space.

Definition 3 (Matroid)**.** *A matroid M is an ordered pair $(E, \mathcal{I})$ consisting of a ground set $E(M)$ and a collection $\mathcal{I}(M)$ of subsets of $E(M)$ such that* *(i1)* $\emptyset \in \mathcal{I}$,

(i2) if $I_1 \subset I_2$ *and* $I_2 \in \mathcal{I}$, *then* $I_1 \in \mathcal{I}$, *and*

(i3) if $I_1,\ I_2 \in \mathcal{I}$ *and* $|I_1| < |I_2|$, *then there exists an element* e *of* $I_2 - I_1$ *such that* $I_1 \cup e \in \mathcal{I}$.

The members of $\mathcal{I}$ are defined as the independents (independent sets) of M. Two specific examples of matroids are the vector matroid and the graphic matroid. A vector matroid $M[\boldsymbol{B}]$ is defined over the matrix $\boldsymbol{B}$, with E being the set of column labels of $\boldsymbol{B}$ and $\mathcal{I}$ being the set of subsets of E that are are linearly independent. A graphic matroid $M(G)$ is defined over an undirected graph $G = (V, E)$, not necessarily simple, having a non-empty set $V(G)$ of vertices and a multiset $E(G)$ of edges. The set $E(G)$ forms the ground set of $M(G)$, with $\mathcal{I}$ consisting of the subsets of $E(G)$ that does not form a cycle.

Any maximal independent set of a matroid forms its *base*. We denote, by $\mathcal{B}(M)$, the collection of all bases of $M(E, \mathcal{I})$. For every matroid there is an associated dual matroid M^* defined over the same ground set E with bases

$$\mathcal{B}(M^*) = \{E - B : B \in \mathcal{B}(M)\}$$

The independents $\mathcal{I}^*$ of M^* is given by

$$\mathcal{I}^* = \{X \mid X \subset E \text{ such that } \exists\ B \in \mathcal{B}(M) \text{ with } X \cap B = \emptyset\}.$$

The independents and bases of M^* are called the *coindependents* and *cobases* of M, respectively. X is a *spanning set* in M if there exists a $B \in \mathcal{B}(M)$ such that $B \subseteq X$.

The *rank* of a matroid M is the cardinality of its bases. Let $X \subseteq E$, then the rank of X is given by

$$r(X) = \text{ max } \{|Y| : Y \subseteq X, Y \in \mathcal{I}\}.$$

Thus, every base of M has a cardinality equal to its rank. The rank of M^*, denoted by $r(M^*)$, is given by

$$r(M^*) = |E| - r(M).$$

The two most important operations over matroids that we use in this dissertation are the restriction and contraction of matroids.

Definition 4 (Restriction (Deletion))**.** *Given an $X \subseteq E(M)$ such that $\mathcal{I}|X = \{I \subseteq X : I \in \mathcal{I}\}$, the pair $(X, \mathcal{I}|X)$ forms a matroid (denoted by $M|X$ or $M \setminus E - X$), defined as the* restriction *of M to X or the* deletion *of $E - X$ from M.*

Definition 5 (Contraction)**.** *Given a matroid M and an $X \subseteq E(M)$, the contraction of X from M (the contraction of M to $E - X$), denoted as M/X, is defined as $(M^* \setminus X)^*$.*

In summary, the contraction is the dual of deletion.

Now, let us present two fundamental lemmas which are direct results of the independent augmentation axiom and form the basis of all our decomposition theorems. The proofs are presented in [26, 22].

Lemma 1. *Let I be an independent set in M. Then there exists a base B containing I in M.*

Lemma 2. *An $X \subset E$ is an independent set in M if and only if $E - X$ is a spanning set in M^*.*

Lemma 2 links the independent sets of a matroid to the spanning sets of its dual.

Definition 6 (Circuit)**.** *A* circuit *is a minimal dependent set in M.*

We represent the collection of circuits in M by $\mathcal{C}(M)$.

Definition 7 (Cocircuit)**.** *A* cocircuit *is a minimal codependent set in M (a minimal dependent set in M^*).*

By $\mathcal{C}^*(M)$ (or $\mathcal{C}(M^*)$) we denote the collection of cocircuits in M.

Definition 8 (Girth)**.** *The cardinality of the smallest circuit in M is termed as the* girth*, denoted as $g(M)$.*

Definition 9 (Cogirth)**.** *The cogirth, deonted as $g^*(M)$, is the cardinality of the smallest cocircuit in M.*

The following lemma, derived from Lemma 2, helps to establish the equivalence between the degree of redundancy and cogirth.

Lemma 3. *If C^* is a cocircuit in M, then*

i. $r(M \setminus C^*) < r(M)$*, and*

ii. for any $X \subset C^*$, $r(M \setminus X) = r(M)$.

Proof. C^*, being a cocircuit, is a minimal codependent set in M. Hence, by Lemma 2, $E(M) - C^*$ is not a spanning set in M. Any $X \subset C^*$ is a coindependent set in M. Therefore, $E(M) - X$ is a spanning set in M. □

Based on Lemma 3, the degree of redundancy, $d(\boldsymbol{A})$, can now be defined as

$$\begin{aligned} d(\boldsymbol{A}) &= g^*(M[\boldsymbol{A}^T]) - 1 \\ &= \underset{D \in \mathcal{C}^*(M[\boldsymbol{A}^T])}{min} |D| - 1. \end{aligned} \tag{2.1}$$

In (2.1), the matroid is defined over $\boldsymbol{A}^T$ instead of $\boldsymbol{A}$. This is because we are interested in the linear dependence relationship among the rows of the design matrix $\boldsymbol{A}$. $g^*(M[\boldsymbol{A}^T])$ denotes the cogirth of $M[\boldsymbol{A}^T]$, and $\mathcal{C}^*(M[\boldsymbol{A}^T])$ denote the collection of cocircuits in M. Hence, the degree of redundancy of $\boldsymbol{A}$ is equal to the cogirth over $M[\boldsymbol{A}^T]$ minus one.

Two other operations that we need to discuss are the truncation and elongation of matroids.

Definition 10 (Truncation)**.** *The k-*truncation *of a matroid $M = (E, \mathcal{I})$ is a matroid $M^{\bar{k}} = (E, \mathcal{I}^{\bar{k}})$ such that for any $V \subseteq E$, $V \in \mathcal{I}^{\bar{k}}$ if and only if $|V| \leq k$ and $V \in \mathcal{I}$.*

If M is an $m \times n$ vector matroid, then $M^{\bar{k}}$ is a $k \times n$ matroid such that for every $I \subseteq \{1, 2, ..., n\}$ of size at most k, the set of columns corresponding to I in M has rank $|I|$ if and only if the corresponding columns in $M^{\bar{k}}$ has rank $|I|$.

Definition 11 (Elongation)**.** *The* l-elongation *of* $M = (E, \mathcal{I})$*, where* $l > r(M)$*, is the matroid* $M^l = (E, \mathcal{I}^l)$ *such that* $V \subseteq E$ *is a base of* M^l *if and only if it contains a base of* M *and* $|V| = l$*.*

We can easily see that the rank of the matroid M^l is l. Note that, the ground sets of $M^{\bar{k}}$ and M^l are the same as that of M. The following remark connects these two notions of a matroid.

Remark 1. *The* l*-elongation of* M *is the dual of the* $(n-l)$*-truncation of* M^**, where* n *is the size of the ground set of* M*.*

The proof of Remark 1 directly follows from the definitions of truncation and elongation. For details, please see [27]

Let us detail some of the properties of structured matroids, which are extensively used in this dissertation. A matroid M is disconnected if and only if, for some proper non-empty subset T of $E(M)$,

$$\mathcal{I}(M) = \big\{I_1 \cup I_2 : I_1 \in \mathcal{I}(M|T), I_2 \in \mathcal{I}(M|(E-T))\big\}.$$

This property implies that

$$r(M) = r(M|T) + r(M|E-T), \text{ and} \tag{2.2}$$

$$r^*(M|T) = r((M|T)^*) = |T| - r(M) + r(M|E-T). \tag{2.3}$$

Naturally, (2.2) and (2.3) imply

$$r(M|T) + r^*(M|T) = |T|. \tag{2.4}$$

Hence, connectivity is self-dual. We can thus conclude that M is connected if and only if M^* is connected [22].

Let $M_1, M_2, ..., M_r$ be matroids on disjoint ground sets $E_1, E_2, ..., E_r$, respectively. Let $\mathcal{I}_1, \mathcal{I}_2, ..., \mathcal{I}_r$ be the collection of independents in $M_1, M_2, ..., M_r$, respectively. Suppose $M(E, \mathcal{I})$ is the matroid with the ground set

$$E = E_1 \cup E_2 \cup ... \cup E_r,$$

and independents

$$\mathcal{I} = \{I_1 \cup I_2 \cup ... \cup I_r : I_j \in \mathcal{I}_j \text{ for all } j \in \{1, 2, ..., r\}\}.$$

Then, we call $M_1, M_2, ..., M_r$ as the direct sum components of M and represent M by

$$M = M_1 \oplus M_2 \oplus ... \oplus M_r.$$

The direct sum has the following properties:

$$\mathcal{C}(M_1 \oplus M_2 \oplus ... \oplus M_r) = \mathcal{C}(M_1) \cup \mathcal{C}(M_2) \cup ... \cup \mathcal{C}(M_r), \tag{2.5}$$

$$(M_1 \oplus M_2 \oplus ... \oplus M_r)^* = M_1^* \oplus M_2^* \oplus ... \oplus M_r^*, \tag{2.6}$$

$$\mathcal{C}^*(M_1 \oplus M_2 \oplus ... \oplus M_r) = \mathcal{C}^*(M_1) \cup \mathcal{C}^*(M_2) \cup ... \cup \mathcal{C}^*(M_r). \tag{2.7}$$

It is a straightforward task to prove (2.5) and (2.6). Then, (2.7) directly follows from (2.5) and (2.6). These equations show that the girth and cogirth of a disconnected matroid is the minimum of the girths and cogirths over its direct sum components, respectively.

Now consider a BBD structured design matrix $\boldsymbol{A}$, with S being the row indices of the border submatrix $[\boldsymbol{S}_1\ \boldsymbol{S}_1\ ...\ \boldsymbol{S}_r]$. By notation $\boldsymbol{A}^T \setminus S$, let us denote the rest of the matrix $\boldsymbol{A}^T$ after the removal of columns indexed by S. Then the deletion of S from $M[\boldsymbol{A}^T]$,

$$M[\boldsymbol{A}^T] \setminus S = M[\boldsymbol{A}^T \setminus S] = M[\boldsymbol{A}_1^T] \oplus M[\boldsymbol{A}_2^T] \oplus ... \oplus M[\boldsymbol{A}_r^T]. \tag{2.8}$$

The first inequality in (2.8) can be found in [26], and the second inequality follows directly from the direct sum property of a connected matroid. We call S a separating set because the removal of indices S from $M[\boldsymbol{A}^T]$ makes the resultant matroid disconnected. For a BBD structured matrix $\boldsymbol{A}$

with no border rows, the cogirth, $g^*(M[\boldsymbol{A}^T])$, is merely the smallest value of the cogirth among the direct sum components of $M[\boldsymbol{A}^T]$. However, most BBD structured design matrices have border rows, the presence of which makes the computation of $g^*(M[\boldsymbol{A}^T])$ complicated.

The matroid representation of a matrix is not unique. The RREF of $\boldsymbol{A}^T$, denoted as $\boldsymbol{A}_s^T = [\boldsymbol{I}_m|\boldsymbol{D}]$, where $\boldsymbol{I}_m$ is the $m \times m$ identity matrix and $\boldsymbol{D}$ is some $m \times (n-m)$ matrix, provides the standard representative matrix for $M[\boldsymbol{A}^T]$. In theory, $M[\boldsymbol{A}^T] = M[\boldsymbol{A}_s^T]$. $\boldsymbol{A}_s^T$ can be obtained by following a Gaussian elimination approach with a partial pivoting which runs in $O(m^2n)$ time [28]. Partial pivoting, in general, generate standard representative matrices free of significant round-off errors so that $M[\boldsymbol{A}^T] = M[\boldsymbol{A}_s^T]$ still holds in a practical setting [29].

For a vector matroid, we can construct an explicit representation for its dual using the standard representative matrix as follows:

Given the standard representative matrix $\boldsymbol{A}_s^T$ of $M[\boldsymbol{A}^T]$, the standard representative matrix of $M^*[\boldsymbol{A}^T]$ is given by the $(n-m) \times n$ matrix $\boldsymbol{H} = [-\boldsymbol{D}^T|\boldsymbol{I}_{n-m}]$.

Note that, the dual matroid $\boldsymbol{H}$ is defined over the same We can now associate the degree or redundancy of $\boldsymbol{A}$ to the girth of the dual matroid $M^*[\boldsymbol{A}^T]$ as:

$$d(\boldsymbol{A}) = g^*(M[\boldsymbol{A}^T]) - 1 = g(M[\boldsymbol{H}]) - 1. \tag{2.9}$$

The second equality in 2.9 is based on the fact that the cogirth of a matroid is also the girth over its dual.

Let us assume that the separating set S of the BBD matrix $\boldsymbol{A}^T$ (since $\boldsymbol{A}$ is BBD structured, $\boldsymbol{A}^T$ will also be BBD structured) is coindependent, i.e., $r(M[\boldsymbol{A}^T \setminus S]) = r(M[\boldsymbol{A}^T])$. Then, the dual $\boldsymbol{H} = [-\boldsymbol{D}^T|\boldsymbol{I}_{n-m}]$ manifests a BBD structure like

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{H}_1 & & & & & & \\ & \boldsymbol{H}_2 & & & & & \\ & & \cdot & & & & \\ & & & \cdot & & & \\ & & & & \cdot & & \\ & & & & & \boldsymbol{H}_r & \\ \boldsymbol{R}_1 & \boldsymbol{R}_2 & \cdot & \cdot & \cdot & \boldsymbol{R}_r & \boldsymbol{I}_{n_s} \end{bmatrix},$$

with nonzero 1-block submatrices $\boldsymbol{H}_i,\ i = 1, ..., r$ and a border submatrix $[\boldsymbol{R}_1\ ...\ \boldsymbol{R}_r\ \boldsymbol{I}_s]$, where $\boldsymbol{I}_{n_s}$ is the $n_s \times n_s$ identity matrix. Each $\boldsymbol{H}_i$ is of size $(n_i - m_i) \times n_i$ and $\boldsymbol{R}_i$ is of size $n_s \times n_i$. Assuming that the column indices of $\boldsymbol{A}^T$ and $\boldsymbol{H}$ are labeled in the same order, the separating set S of $\boldsymbol{A}^T$ also gives the column indices of $\boldsymbol{I}_{n_s}$ in $\boldsymbol{H}$.

We can easily dualize (2.8) using duality of matroids to decompose $\boldsymbol{H}$ into its connected components as

$$M[\boldsymbol{H}]/S = M[\boldsymbol{H}/S] = M[\boldsymbol{H}_1] \oplus M[\boldsymbol{H}_2] \oplus ... \oplus M[\boldsymbol{H}_r], \tag{2.10}$$

where $\boldsymbol{H}/S$ represents the matrix obtained by contracting columns indexed by S from $\boldsymbol{H}$. Hence, the contraction of S from $M[\boldsymbol{H}]$ makes it disconnected. However, unlike deletion, finding the contraction of a vector matroid is more involved.

Let e be the label of a non-zero column of $M[\boldsymbol{H}]$. Then $M[\boldsymbol{H}]/e$ is obtained as follows:

(i) *transform $\boldsymbol{H}$ by pivoting the column indexed by e to a unit vector.*

(ii) *delete the row and column containing the unique non-zero entry in e from $\boldsymbol{H}$.*

It is easy to see that applying the above approach to contract S from the BBD structured matrix $\boldsymbol{H}$ makes it disconnected.

2.3 Existing Approaches to Find the Degree of Redundancy

Let us briefly review the existing algorithms to find the degree of redundancy, $d(\boldsymbol{A})$, of a linear sensor system. A naive approach for finding the degree of redundancy is to exhaustively test the rank of all matrices obtained by deleting some d number of rows from $\boldsymbol{A}$, starting with a value of 1 until a rank reduction is obtained. Then, the value $d-1$ obtained upon termination gives $d(\boldsymbol{A})$. This "brute force" procedure is undoubtedly impractical.

Cho et al. [2] took advantage of the BBD structure of $\boldsymbol{A}$ to decompose it into disjoint submatrices and performed rank-testing on these smaller matrices to find $d(\boldsymbol{A})$. They proved the following decomposition theorem using properties of the matroids to achieve this decomposition. We call this theorem, the bound and decompose theorem.

Theorem 1 (Bound and Decompose Theorem [2]). *For any $t \in \{1, ..., r\}$,*

$$\text{if } g^*(M[\boldsymbol{A}^T]) \geq \frac{t+1}{t} n_s - 1, \text{ then}$$

$$g^*(M[\boldsymbol{A}^T]) = \mathit{min}\big\{g^*(M[\boldsymbol{A}[U]^T]) : U \subseteq \{1, 2, ..., r\} \text{ and } |U| = t\big\}.$$

According to this theorem, if the cogirth, $g^*(M[\boldsymbol{A}^T])$, is known to be greater than or equal to the so-called "decomposition bound" given by $\frac{t+1}{t} n_s - 1$, its value can be obtained by finding the minimum of the cogirths over all the t-block submatroids of $M[\boldsymbol{A}^T]$. The bound and decompose algorithm, presented in [2], takes advantage of Theorem 1 to decompose the design matrix into the smallest possible submatrices allowable based on the decomposition bound. Then, the degree of redundancy of $\boldsymbol{A}$ can be computed by rank testing these submatrices. The algorithm (denoted by BDNEW) is presented below.

BDNEW*: Compute the degree of redundancy of* $\boldsymbol{A}$

Input: Matrix $\boldsymbol{A} \in \mathbb{R}^{n \times m}$ with blocks $\boldsymbol{A}_1, ..., \boldsymbol{A}_r$, and a separating set S.

Step 0. $d \leftarrow 1$

```
Step 1. if $d < \lceil \frac{r}{r-1} n_s - 1 \rceil$,
                $t \leftarrow r$
        else $t \leftarrow \lceil \frac{n_s}{d-n_s+1} \rceil$
Step 2. for all $U \subseteq \{1, 2, ..., r\}, |U| = t$:
                if $\exists$ an index $D \subseteq \{1, 2, ..., n\}, |D| = d$ such that $r(\boldsymbol{A}[U]_{(-D)}) < r(\boldsymbol{A}[U])$
                        stop, and return $d - 1$
Step 3. $d \leftarrow d + 1$ and go to Step 1
```

The bound and decompose algorithm tests the ranks of matrices $\boldsymbol{A}_{(-D)}$'s until d reaches $\lceil \frac{r}{r-1} n_s - 1 \rceil$, the bound that permits rank testing over $(r-1)$-block submatrices. As the value of d increases, the bound allows for rank testing smaller and smaller submatrices. At each iteration, we increment d by 1 and continue rank testing the smallest possible submatrices based on the decomposition bound until a rank reduction is observed. The value returned by the BDNEW algorithm is the degree of redundancy $d(\boldsymbol{A})$. In the original version of this algorithm (presented in [2]), the value of t computed in Step 1 gives the minimum number of matrices to be rank-tested based on the decomposition property. In [30], Bansal et al. modified this algorithm to find a t that minimizes the size of submatrices, $\boldsymbol{A}[U]$s', to be rank-tested. We presented this modified version of the bound and decompose algorithm as it is computationally efficient compared to the previous one presented in [2]. Hence, we denote this algorithm by BDNEW. Note that, the condition in Step 1 of BDNEW finds a t that minimizes the submatrix size.

Despite its clear advantage over the exhaustive testing approach, the BDNEW algorithm has many limitations. For one, the number of submatrices that need to be rank tested can get exponentially large for matrices with large cogirths. For another, if the number of border rows (n_s) is large, the size of the submatrices to be rank tested becomes considerably large. As an example, consider a design matrix with just three border rows and six blocks. Until $d \geq 3$, no decomposition is possible. Therefore, a total of $n + \binom{n}{2}$ rank testings need to be performed over the matrix $\boldsymbol{A}$

until d reaches 3, provided no rank reduction is observed. When d reaches 3, 3-block submatrices can be tested based on the decomposition bound. But, there are 20 such 3-block submatrices. And for each of them $\binom{n_u}{3}$ rank testings need to be done in the worst case, where n_u is the number of rows in a chosen $\boldsymbol{A}[U]$. 1-block matrices can be rank tested only when d reaches 5. Evidently, this decomposition approach is not effective when dealing with large scale problems.

Later, Kianfar et al. [14] formulated the redundancy degree problem as a 0-1 MIP. Their formulation, denoted as MIP$_{\text{CO}}$ where ‘co’ stands for the cocircuit, finds the smallest cocircuit of an input matrix $\boldsymbol{A}$. The MIP$_{\text{CO}}$ algorithm is based on the fact that any rank deficient matrix has a nonzero null space. Their formulation is presented below.

MIP$_{\text{CO}}$: *Smallest Cocircuit Problem ([14])*

Given a linear sensor system defined by the design matrix $\boldsymbol{A}$

$$g^*(M[\boldsymbol{A}^T]) = \min \mathbf{1}\boldsymbol{\alpha} \tag{2.11}$$

$$\text{subject to} \qquad -\boldsymbol{\alpha} \leq \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{\alpha} \tag{2.12}$$

$$-\mathbf{1} + 2\boldsymbol{z} \leq \boldsymbol{x} \leq \mathbf{1},\ \mathbf{1}\boldsymbol{z} = 1 \tag{2.13}$$

$$\boldsymbol{x} \in \mathbb{R}^m, \boldsymbol{\alpha} \in \{0,1\}^n,\ \boldsymbol{z} \in \{0,1\}^m \tag{2.14}$$

In the MIP$_{\text{CO}}$ formulation, $\mathbf{1}$ stands for a vector of all ones. The algorithm finds the cogirth $g^*(M[\boldsymbol{A}^T])$ as follows. The constraint set (2.12) searches for a nonzero $\boldsymbol{x} \in \mathbb{R}^m$ such that the number of rows, $\boldsymbol{a}^i$’s, for which $\boldsymbol{a}^i\boldsymbol{x} \neq 0$ is minimized. The constraints (2.13) guarantee a nonzero solution $\boldsymbol{x}$. The objective function (2.11) minimizes the number of α_i’s set to 1. Any α_i corresponding to the constraint $\boldsymbol{a}^i\boldsymbol{x} \neq 0$ needs to be set to 1. The set of indices i for which $\alpha_i = 1$ gives the smallest cocircit of $M[\boldsymbol{A}^T]$, and hence $d(\boldsymbol{A}) = \mathbf{1}\boldsymbol{\alpha} - 1$. The row vectors ($\boldsymbol{a}^i$’s) are scaled such that $\|\boldsymbol{a}^i\|_1 = \sum_{j=1}^m |a_{ij}| = 1$, to ensure the feasibility of constraints (2.12). Although MIP$_{\text{CO}}$ ignores any structure inherent in $\boldsymbol{A}$, it is shown in [14] that for moderate instances with large decomposition bounds MIP$_{\text{CO}}$ outperforms BDNEW considerably. However, MIPs can be computationally expensive, particularly for large-scale problems. (For more on mixed integer programming, please refer [31].)

In [30], Bansal et al. proposed a hybrid algorithm, represented as BDMIF, by integrating a mixed integer feasibility (MIF) checking algorithm based on MIP$_{\text{CO}}$ with the bound-and-decompose framework presented in [2]. BDMIF capitalizes on the benefits of both bound-and-decompose approach and MIP. In essence, BDMIF uses Theorem 1 to break the problem into subproblems and replaces the rank testing of submatrices, $\boldsymbol{A}[U]$s', with an MIF component that executes MIP$_{\text{CO}}$ over these submatrices. The formulation for MIF differs from MIP$_{\text{CO}}$ in that the objective function value (2.11) is fixed to a constant integer value d. This algorithm is presented below.

BDMIF*: Compute the degree of redundancy of* $\boldsymbol{A}$

Input: Matrix $\boldsymbol{A} \in \mathbb{R}^{n \times m}$ *with blocks* $\boldsymbol{A}_1, ..., \boldsymbol{A}_r$*, and a separating set* S*.*

```
Step 0.  $d \leftarrow 1$
Step 1.  if $d < \lceil \frac{r}{r-1} n_s - 1 \rceil$,
                 $t \leftarrow r$
         else $t \leftarrow \lceil \frac{n_s}{d - n_s + 1} \rceil$
Step 2.  for all $U \subseteq \{1, 2, ..., r\}, |U| = t$:
    Step 2.1  Solve the MIF problem defined as follows:
```

$$
\begin{aligned}
& & \min \mathbf{1}\boldsymbol{\alpha} = d & \qquad (2.15)\\
& \text{subject to} & -\boldsymbol{\alpha} \le \boldsymbol{A}[U]\boldsymbol{x} \le \boldsymbol{\alpha} & \\
& & -\mathbf{1} + 2\boldsymbol{z} \le \boldsymbol{x} \le \mathbf{1},\ \mathbf{1}\boldsymbol{z} = 1 & \\
& & \boldsymbol{x} \in \mathbb{R}^{m_u}, \boldsymbol{\alpha} \in \{0,1\}^{n_u},\ \boldsymbol{z} \in \{0,1\}^{m_u} &
\end{aligned}
$$

```
  Step 2.2  if $\exists$ some $\boldsymbol{A}[U]$ for which MIF has a feasible solution
                stop; return $d - 1$
Step 3.  $d \leftarrow d + 1$ and go to Step 1
```

As with the BDNEW algorithm d is initially chosen as 1 and is incremented until a submatrix $A[U]$ is found for which MIF has a feasible solution. The feasibility is checked by forcing the objective function (2.15) to take a value equal to d. Until the first decomposition bound $\lceil \frac{r}{r-1} n_s - 1 \rceil$ is reached, the MIF component is solved for the entire design matrix $\boldsymbol{A}$. As d increases, we solve MIF over the smallest permissible submatrices based on Theorem 1. For any d, if MIF finds a feasible solution, we terminate with $d(\boldsymbol{A}) = d - 1$. The row vectors ($\boldsymbol{a}[U]^i$'s of $\boldsymbol{A}[U]$) are normalized so that $\|\boldsymbol{a}[U]^i\|_1 = \sum_{j=1}^{m_u} |a[U]_{ij}| = 1$. Clearly, BDMIF provides some computational advantage by capitalizing on the benefits of both BDNEW and MIP_{CO}. However, the size and the number of subproblems to be solved still dependents on Theorem 1. This dependence, coupled with the inefficiency of MIPs when executed over large inputs, makes this algorithm less effective for design matrices with thick borders, or large blocks, or even matrices with a large number of blocks.

The inefficiency of the existing algorithms in solving many large-scale problems of practical significance has inspired us to design new and improved algorithms to solve this problem.

2.4 Existing Approaches for Reliability Evaluation

The reliability of a linear sensor system is a probabilistic evaluation of the ability of the system to withstand sensor failures. We study the reliability of the well-known (k out of n)-linear sensor system $\mathcal{S}$ consisting of a set of sensors, $s_1, s_2, ..., s_n$, designed over the matrix $\boldsymbol{A}$. We assume each sensor s_i to function (or fail) independently of each other. We also assume that the sensor failures are catastrophic, meaning no useful information can be derived from the failed sensors.

Each sensor s_i has only two states: 0 if the sensor works, and 1 if the sensor fails. Let p_i denote the working probability of sensor s_i. Define $\mathfrak{I}_{(.)}$ to be the indicator function: $\mathfrak{I}_{(\text{True})} = 1$

and $\mathfrak{I}_{(\text{False})} = 0$. If the r.v. v_i denotes the state of the sensor s_i, then

$$v_i = \mathfrak{I}_{(s_i \text{ works})}.$$

Now, the *component (node) state vector* $\boldsymbol{v} = (v_1, v_2, ..., v_n)$ determines the state of the system S. Define the r.v. ϕ_S as

$$\phi_S = \mathfrak{I}_{(\mathcal{S} \text{ works})}.$$

ϕ_S can be determined based on the reliability definition in Section 2.4. If we define D as

$$D = \{i : v_i = 0, i = 1, ..., n\},$$

then, by the reliability definition, $\phi_S = 1$ if and only if $r(\boldsymbol{A}_{(-D)}) = m$ and $n - |D| \geq k$. The reliability r of the system S can be computed as

$$r = Pr\{\phi_S = 1\} = E(\phi_S), \tag{2.16}$$

where $Pr(.)$ denotes the probability and $E(.)$ denotes the expectation of a r.v.

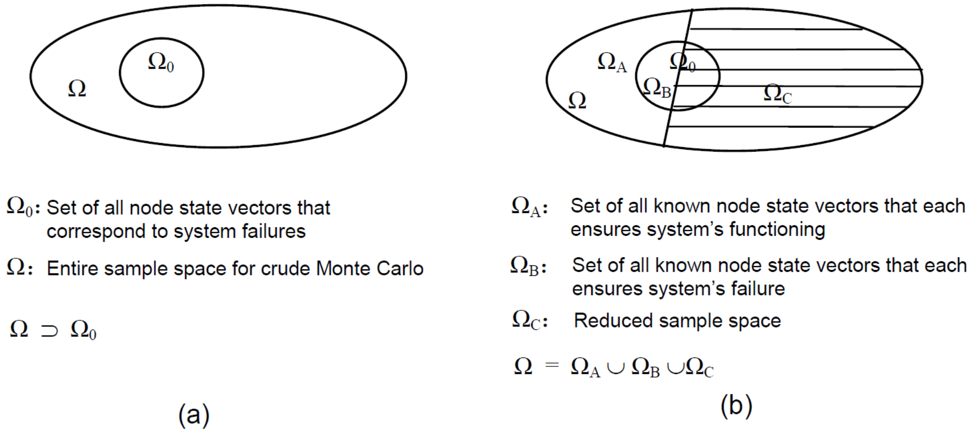

Ω₀
Ω
Ω₀: Set of all node state vectors that correspond to system failures
Ω: Entire sample space for crude Monte Carlo
Ω ⊃ Ω₀
(a)
Ω_A
Ω₀
Ω_B
Ω_C
Ω
Ω_A: Set of all known node state vectors that each ensures system's functioning
Ω_B: Set of all known node state vectors that each ensures system's failure
Ω_C: Reduced sample space
Ω = Ω_A ∪ Ω_B ∪Ω_C
(b)

Figure 2.2: (a) Sample Space for Crude Monte Carlo Method; (b) Reduced Sample Space (Reprinted from [10])

Let Ω be the set of all possible component state vectors $\boldsymbol{v}$. To find the exact value of the reliability, one needs to find all the component state vectors that result in the system failure (or success). For a large-scale system, the sample space Ω of all component state vectors is very large because it consists of 2^n elements. The only practical option is to estimate the reliability. A simple approach for estimating the reliability r of system $\mathcal{S}$ is to use the Crude Monte Carlo (CMC) method. The unbiased CMC estimate $\hat{r}_c$ of r can be obtained by generating N independent samples $\boldsymbol{v}^{(j)}, j = 1, ..., N$, of $\boldsymbol{v}$, and then calculating the sample mean $\hat{r}_c$ as

$$\hat{r}_c = \frac{\sum_{j=1}^{N} \phi_S^{(j)}}{N},$$

where $\phi_S^{(j)}$ is the system state for sample j. The variance of the estimator $\hat{r}_c$ is given by

$$V_c = \frac{Var(\phi_S)}{N} = \frac{r(1-r)}{N}.$$

The unbiased estimator of this variance is

$$\hat{V}_c = \frac{1}{N(N-1)} \sum_{j=1}^{N} (\phi_S^{(j)} - \hat{r}_c)^2.$$

A well known drawback of CMC is the large number of samples required to have an accurate estimate of r when the sensor system is highly reliable due to the extremely small percentage of samples that contribute to system failure. This is clearly depicted in Figure 2.2 (a). In the figure, Ω_0 represents the set of all component state vectors that correspond to sensor failure. Ω_0 is only a small portion of Ω if the system is highly reliable. As a result, one will have to take a very large number of random samples from Ω to get sufficient number of samples in Ω_0.

In the network reliability literature, an improved Monte Carlo method called the Recursive Variance Reduction (RVR) technique is developed to estimate the reliability. This RVR method has been shown to outperform many other enhanced Monte Carlo methods for network reliability [25]. The main idea of the RVR method is to obtain sets of component state vectors that guarantee a system failure or success before Monte Carlo sampling. This is illustrated in Figure 2.2 (b). The set Ω_A in Figure 2.2 (b) contains known component state vectors that ensure the system is working. Another set Ω_B contains component state vectors that result in system failure. If we can predetermine Ω_A and Ω_B , then the effective sample space for Monte Carlo sampling is reduced to $\Omega_C = \Omega - (\Omega_A + \Omega_B)$. As we will see later, the larger the probability of such predetermined component state vectors, the lesser the variance of the reliability estimate. In the RVR approach, we recursively reduce the sample space and then perform Monte Carlo sampling over these smaller sample spaces to estimate the reliability until a desired level of accuracy is achieved. Yang and Chen ([3]) extended the RVR approach to the linear sensor system reliability estimation problem. We now discuss the basics of the RVR method, however, with a more rigorous mathematical approach than in [3].

When applying the RVR framework to a linear sensor system, predetermined component state vectors that ensure a system failure are obtained by finding the *minimal cut sets* of the system.

Definition 12 (Minimal Cut Set [10, 3]). *Given a linear sensor system with some sensor states fixed (either working or failed), a minimal cut set is defined as the minimal set of sensors among the remaining ones whose simultaneous failure results in the system failure.*

A minimal cut set of a system S with no fixed sensor states can be obtained by finding a cocircuit over the design matrix $\boldsymbol{A}^T$, as detailed in the following remark. This remark follows directly from the reliability definition in Section 2.3.

Remark 2. *Let C^* be a cocircuit over $M[\boldsymbol{A}^T]$. If $|C^*| \leq n - k + 1$, then C^* is a minimal cut set of S. Otherwise, each subset of C^* with a cardinality of $n - k + 1$ is a minimal cut set.*

Proof. Assume that $|C^*| \leq n - k + 1$ and let $D \subset C^*$. Then, $r(\boldsymbol{A}_{(-D)}) = r(\boldsymbol{A})$ and the number of rows in $\boldsymbol{A}_{(-D)} \geq k$. Consequently, the simultaneous failure of sensors corresponding to any proper subset of C^* cannot result in the system failure and hence C^* is a minimal cut set of S.

Now consider the case when $|C^*| > n - k + 1$. Then, any $D' \subset C^*$ with cardinality equal to $n - k + 1$ is such that the number of rows in $\boldsymbol{A}_{(-D')} < k$, causing system failure. Also, any each proper subset of D' will not result in a system failure. This makes D' a minimal cut set. □

Then, Remark 3 follows directly from the property of elongation of matroids.

Remark 3 ([10])**.** *For a linear sensor system S over $\boldsymbol{A}$, each of its minimal cut sets corresponds to a cocircuit of $M^k[\boldsymbol{A}^T]$, the k-elongation of matroid $M[\boldsymbol{A}^T]$.*

From Section 2.2, we know that the rank of a k-elongated matroid $M^k[\boldsymbol{A}^T]$ is k. Then, one can easily interpret why any D' with $|D'| = n - k + 1$ as defined above will be a cocircuit in $M^k[\boldsymbol{A}^T]$. For a detailed proof, please refer to [10]. Now, consider a system S in which some sensors are fixed as working or failed. For such a system, the remark below helps to find its minimal cut sets.

Remark 4 ([10])**.** *For a linear sensor system S over $\boldsymbol{A}$, with W indicating the sensors fixed as working and F indicating those fixed as failed, each of its minimal cut sets corresponds to a cocircuit of the updated system defined over $M^{k-|W|}[(\boldsymbol{A}^T/W) \setminus F]$.*

$M[(\boldsymbol{A}^T/W) \setminus F]$ is the vector matroid obtained by contracting and deleting columns indexed by W and F, respectively, from $M[\boldsymbol{A}^T]$. Also, $M^{k-|W|}[(\boldsymbol{A}^T/W) \setminus F]$ is the $(k - |W|)$-elongation of $M[(\boldsymbol{A}^T/W) \setminus F]$. Remark 4 says that the minimal cut sets of a (k out of n)-system S over $M[\boldsymbol{A}^T]$ with certain fixed sensors is also the minimal cut sets of a ($k - |W|$ out of n)- system over $M[(\boldsymbol{A}^T/W) \setminus F]$. Therefore, to account for failed sensors, one needs to delete the corresponding columns from $\boldsymbol{A}^T$, and to account for working sensors, those columns needs to be contracted from $\boldsymbol{A}^T$. For a more detailed understanding of Remark 4, please refer [10].

Now, let us understand the mechanics behind the RVR approach. Consider a minimal cut set in $\mathcal{S}$ given by $G = \{s_{<1>}, s_{<2>}, ..., s_{<h>}\}$, with $\{v_{<1>}, v_{<2>}, ..., v_{<h>}\}$ being the corresponding component state vector. Define

$$\Omega_0 = \{\boldsymbol{v} : v_{<t>} = 0,\ t = 1, ..., h\}, \text{ and} \tag{2.17}$$

$$\Omega_t = \{\boldsymbol{v} :\ v_{<l>} = 0,\ l = 1, ..., t-1, v_{<t>} = 1\} \text{ for } t = 1, ..., h. \tag{2.18}$$

Then $\Omega = \bigcup_{t=0}^{h} \Omega_t$. Let

$$Q_0 = Pr\{\Omega_0\} = \prod_{t=1}^{h} (1 - p_{<t>}). \tag{2.19}$$

Define f_G as the discrete r.v. with pmf

$$\begin{aligned} f_G(t) &= \frac{Pr\{s_{<1>}, ..., s_{<t-1>} \text{ is failed, and } s_{<t>} \text{ is working}\}}{1 - Q_0},\ t = 1, ..., h \\ &= Pr\{\Omega_t | \Omega_{0^c}\} = \frac{p_{<t>} \prod_{l=1}^{t-1} (1 - p_{<t>})}{1 - Q_0} \end{aligned}$$

Let S_{G_t}, $t = 1, ..., h$, be the system derived from S in which the columns corresponding to sensors $s_{<1>}, ..., s_{<t-1>}$ in $\boldsymbol{A}^T$ are deleted, and the column corresponding to $s_{<t>}$ is contracted. Hence, the system S_{G_t} is defined over the matroid $M[(\boldsymbol{A}^T/W) \setminus F]$, where W denotes the indices of deleted sensors and F that of the contracted sensors. By $\boldsymbol{A}^T_{(-F)/W}$, let us denote the reduced matrix that is represented by the matroid $M[(\boldsymbol{A}^T/W) \setminus F]$. We define $p(S_{G_t}) = Pr\{\Omega_t | \Omega_{0^c}\}$.

For a system S, let the r.v. $\Xi(S)$ be defined as:

$$\Xi(S) = \begin{cases} 1 & \text{if } S \text{ works} \\ 0 & \text{if } S \text{ fails} \\ Q_0^c . \sum_{t=1}^{h} \mathfrak{I}_{(f_G = t)} . \Xi(S_{G_t}) & \text{otherwise} \end{cases} \tag{2.20}$$

Then,

$$\begin{aligned} E(\Xi(S)) &= r, \text{ and} \\ Var(\Xi(S)) &= r(1 - r - Q_0) \leq r(1 - r) = Var(\phi_S). \end{aligned} \tag{2.21}$$

Hence, $E(\Xi(S))$ is also an unbiased estimator of r, however, with a smaller variance. From (2.21), we can easily realize that a large value of Q_0, i.e., a minimal cut set with large failure probability, results in a more accurate estimate of reliability.

Now, let us consider generating a sample $\boldsymbol{v}^{(j)}$ from the reduced space $(\Omega - \Omega_0)$. Note that, $\boldsymbol{v}^{(j)}$ belongs to some $\Omega_t, t = 1, ..., h$, defined in (2.18). Generate the trial $\Xi(S)^{(j)}$ based on (2.20) as:

$$\Xi(S)^{(j)} = Q_0^c.\Xi(S_{G_t})^{(j)}, \tag{2.22}$$

where $\Xi(S_{G_t})^{(j)}$ is a trial of $\Xi(S_{G_t})$. Let

$$\phi_{S_{G_t}} = \mathcal{I}\{S_{G_t} \text{ works}\}.$$

If the system S_{G_t} is deterministic, i.e., if $\phi_{S_{G_t}} = 1$ or 0, then (2.22) returns with $\Xi(S)^{(j)} = Q_0^c.\phi_{S_{G_t}}$. Now consider generating N independent trials $\Xi(S)^{(1)}, ..., \Xi(S)^{(N)}$ of $\Xi(S)$. Then,

$$\widehat{\Xi(S)} = \frac{\sum_{j=1}^{N} \Xi(S)^{(j)}}{N}$$

is an unbiased estimator of r. We can represent this recursive structure by a tree where the root corresponds to the system S under study, each internal node corresponds to a recursive call, and the leaf nodes correspond to a deterministic system. Since the number of sensors in the system gets reduced by at least one at each recursive call, the height of the tree is $O(n)$ [25]. However, this recursive approach to estimate relaibility is not workable for large systems.

A better implementation of the RVR approach is presented in [25] and is used by Yang and Chen in [3]. Let $N(S_{G_t})$ be the binomial r.v. with parameters N and $p(S_{G_t})$. Since $\sum_{t=1}^{h} N(S_{G_t}) = N$ and $\sum_{t=1}^{h} p(S_{G_t}) = 1$, the random vector $\big(N(S_{G_1}), ..., N(S_{G_h})\big)$ has a multinomial distribution given by

$$\mathcal{M}\Big(N, p(S_{G_1}), ..., p(S_{G_h})\Big).$$

Then, the recursive r.v.

$$\psi_1(S,N) = \begin{cases} 0 & \text{if } N = 0 \\ N & \text{if } S \text{ works} \\ 0 & \text{if } S \text{ fails} \\ Q_0^c.\sum_{t=1}^{h} \psi(S_{G_j}, N_{S_{G_j}}) & \text{otherwise} \end{cases}$$

has the same distribution as $\sum_{j=1}^{N} \Xi(S)^{(j)}$. Now we can use the unbiased estimator $(\psi_1(S,N))/N$ instead of $\widehat{\Xi(S)}$ to estimate r.

The unbiased estimator of $Var\big(\widehat{\Xi(S)}\big)$:

$$\begin{aligned} \hat{V}_{\Xi(S)} &= \frac{1}{N.(N-1)} \sum_{j=1}^{N} \Big(\widehat{\Xi(S)} - \Xi(S)^{(j)}\Big)^2 \\ &= \frac{1}{(N-1)} \bigg(\frac{1}{N} \sum_{j=1}^{N} \big(\Xi(S)^{(j)}\big)^2 - \big(\widehat{\Xi(S)}\big)^2 \bigg). \end{aligned}$$

It can be shown that the recursive r.v.

$$\psi_2(S,N) = \begin{cases} 0 & \text{if } N = 0 \\ N & \text{if } S \text{ works} \\ 0 & \text{if } S \text{ fails} \\ (Q_0^c)^2.\sum_{t=1}^{h} \psi_2(S_{G_t}, N_{S_{G_t}}) & \text{otherwise} \end{cases}$$

has the same distribution as $\sum_{j=1}^{N} \big(\Xi(S)^{(j)}\big)^2$. Then an unbiased estimator of $Var\big(\widehat{\Xi(S)}\big)$ is given by:

$$\hat{V}_{\psi,\Xi(S)} = \frac{1}{N-1} \left[\frac{\psi_2(S,N)}{N} - \left(\frac{\psi_1(S,N)}{N} \right)^2 \right].$$

This modified RRV approach provides a more viable alternative to estimate the reliability and the associated variability. In practice, $Var\big(\widehat{\Xi(S)}\big)$ is commonly estimated using the sample variance given by

$$\frac{1}{N-1}\left[\psi_2(S,N) - \frac{\psi_1(S,N)^2}{N}\right].$$

We now present the RVR approach in [3] that applies the above RVR framework to the linear sensor system reliability evaluation problem. At first, let us detail the approach by which minimal cut sets are obtained in [3]. Given the indices W and F of the working and failed sensors, respectively, the minimal cut sets of S are computed in [3] as follows. We denote this approach by RREF_{CO}.

$\mathbf{RREF_{CO}}$: *Find the minimal cut set of a linear sensor system*

Step 1. ***get*** $\boldsymbol{A}^T_{(-F)/W}$ *with* $\bar{n} = n - |W| - |F|$ *and* $\bar{k} = k - |W|$

Step 2. ***find*** *the RREF of* $\boldsymbol{A}^T_{(-F)/W}$

Step 3. ***find*** *a set of sensors* G *that correspond to the nonzero elements in a nonzero row of* $\boldsymbol{A}^T_{(-F)/W}$

Step 4. ***if*** $|G| \leq \bar{n} - \bar{k} + 1$, ***then*** G *is a minimal cut set of* S

Step 5. ***else*** *each subset of* G *with a cardinality of* $\bar{n} - \bar{k} + 1$ *is a minimal cut set*

The critical step in $\mathbf{RREF_{CO}}$ is the Step 3 that finds a cocircuit of $M[(\boldsymbol{A}^T/W) \setminus F]$. This Step applies the following observation to find a cocircuit.

Observation 1. *The set of sensors that correspond to the nonzero elements in any nonzero column of a standard representation (RREF) of a vector matroid gives a cocircuit of the matroid.*

Please refer [3] for the proof. Specifically, Step 3 of $\mathbf{RREF_{CO}}$ finds a cocircuit over $\boldsymbol{A}^T_{(-F)/W}$. Steps 4 and 5 finds a minimal cut set based on the cocircuit from Step 3. Let us now present the RVR algorithm algorithm, which we denote as RVR_{RREF}, proposed in [3].

$\mathbf{RVR_{RREF}}$*: Estimate the System Reliability [3]*

Given a sensor system defined by the matrix $\boldsymbol{A}$ *having* n *sensors* $S = (s_1, s_2, ..., s_n)$ *with working probabilities* p_i, $i = 1, ..., n$. *A sample size of* N *is chosen. Set* $W = F = \emptyset$.

```
Step 1. Execute the following procedure to calculate $\alpha$ and $\beta$

    Procedure $P(W, F, N)$:

      1. if $N = 0$
             $\alpha = \beta = 0$; return
      2. else if $r(\boldsymbol{A}_{(W)}) = m$ and $|W| \geq k$
             $\alpha = \beta = N$; return
      3. else if $r(\boldsymbol{A}_{(-F)}) < m$ or $n - |F| < k$
             $\alpha = \beta = 0$; return

      4. find a minimal cut set $G = \{s_{<1>}, s_{<2>}, ..., s_{<h>}\}$ of $S$ using $\mathrm{RREF_{co}}$

      5. divide $\Omega$ into $\Omega_0, \Omega_1, ..., \Omega_h$ based on (2.17) and (2.18). Define $Q_0 = Pr\{\Omega_0\}$

      6. generate a trial $\big(N(S_{G_t}), ..., N(S_{G_h})\big)$ of the r.v. with multinomial distribution
             $\mathcal{M}\Big(N, p(S_{G_1}), ..., p(S_{G_h})\Big)$, where $p(S_{G_t}) = Pr\{\Omega_t|\Omega_0^c\}$

      7. for each $t \in \{1, 2, ..., h\}$:
         7.1 set $W_t = W \cup s_{<t>}$ and $F_t = F \bigcup_{l=1}^{t-1} s_{<l>}$
         7.2 call procedure $\Big(P(W_t, F_t, N(S_{G_t}))\Big)$ to calculate $\alpha_t$ and $\beta_t$

Step 2. get $\alpha = (1 - Q_0) \sum_{t=1}^{h} \alpha_t$ and $\beta = (1 - Q_0)^2 \sum_{t=1}^{h} \beta_t$

Step 3. calculate $\hat{r}_{rvr} = \frac{\alpha}{N}$ and $\hat{V}_{rvr} = \frac{1}{N-1}\left[\beta - \frac{\alpha^2}{N}\right]$
```

The RVR algorithm works as follows: At first, we call the procedure $P(W, F, N)$ to obtain a minimal cut set of the system S having no fixed sensors. We divide the sample space Ω into disjoint sets $\Omega_0, \Omega_1, ..., \Omega_h$ and then generate a trial $\big(N(S_{G_t}), ..., N(S_{G_h})\big)$ based on a multinomial distribution that captures the conditional probabilities of each of the Ω_i's, $i = 1, 2, ..., h$, given Ω_0^c. Note that, we are sampling from a reduced space $\Omega - \Omega_0$, where Ω_0 represents one set of component state vectors that result in sensor failure. We proceed by iteratively calling the procedure $(P(W_t, F_t, N(S_{G_t}))$ with the updated set W_t and F_t of fixed sensors. W_t captures those sensors that are fixed as working and F_t takes those that are fixed as failed. The values of α and β are calculated from the values of α_t and β_t obtained from the recursive calls of the procedure $\big(P(W_t, F_t, N(S_{G_t})\big)$. Finally, Step 3 finds the estimate $\hat{r}_{rvr}$ of reliability with a variance estimate $\hat{V}_{rvr}$.

The crucial step in the RVR$_{\text{RREF}}$ algorithm is the RREF$_{\text{CO}}$ procedure employed in finding the minimal cut sets. One way to improve the accuracy of RVR$_{\text{RREF}}$ algorithm is by finding minimal cut sets with the maximum failure probability at each iteration of Step 4. However, a quick analysis of the RVR$_{\text{RREF}}$ algorithm shows that the RREF$_{\text{CO}}$ method is not an ideal approach for finding such minimal cut sets. For example, when all the sensors have the same working probability, we prefer a minimal cut set (also a cocircuit over $M[(\boldsymbol{A}^T/W) \setminus F]$) with the smallest cardinality. RREF$_{\text{CO}}$ often fails to obtain a minimal cut set with the smallest cardinality.

Theoretically, an ideal alternative is to use MIP$_{\text{CO}}$ over $\boldsymbol{A}^T_{(-F)/W}$ at each iteration to find its smallest cocircuit (or equivalently MIP$_{\text{CIR}}$ over its dual to find its smallest circuits) and then integrate this within the RVR framework. These circuits and cocircuits provide minimal cut sets with large failure probabilities compared to the RREF$_{\text{CO}}$ approach even when applied to systems having sensors with different working probabilities. This approach is however not recommended for systems with large design matrices because of the difficulties associated with solving MIPs. Hence, we need more sophisticated decomposition approaches to obtain a more accurate estimate of reliability without compromising on the running time of the overall algorithm.

3. A HEURISTIC APPROACH TO FIND THE DEGREE OF REDUNDANCY

3.1 An l_1 Minimization Approach

In this Section, we present the heuristic approach that we proposed to find the degree or redundancy. In Section 2.2, we re-formulated the degree of redundancy problem as a girth problem over the dual matroid $M[\boldsymbol{H}]$, the dual matroid of $M[\boldsymbol{A}^T]$. Our heuristic approach utilizes the techniques from compressed sensing to reformulate this dual girth problem into an l_0 minimization problem. Compressed sensing is a field that studies signal reconstruction building on the fact that we can represent many signals using only a few nonzero coefficients in certain basis or dictionaries [32]. It means that we can design sensing technologies that acquire some types of data using far fewer measurements than classical systems allow. Rather than first sampling in high rate and then compressing the sampled data, compressed sensing aims for directly sampling data in a compressed form thereby reducing the complexity and the computational cost of the data acquisition stage. In recent years, compressed sensing has attracted substantial attention in areas of applied mathematics, signal processing, statistics, and computer science. [32] gives a thorough introduction to the mathematics behind compressed sensing.

A large part of the theory of compressed sensing studies finding the sparsest representation of random signals in a given dictionary. This requires solving an l_0-minimization problem as with finding the girth of a matroid. This problem is highly non-linear and non-convex. Let us first formulate the girth problem as an l_0-minimization problem as given below.

$$
\begin{aligned}
g(M[\boldsymbol{H}]) &= \min \|\boldsymbol{y}\|_0 \\
\text{subject to} \qquad \boldsymbol{H}\boldsymbol{y} &= \boldsymbol{0} \\
\boldsymbol{y} &\neq \boldsymbol{0}
\end{aligned}
$$

The l_0-norm $\|\boldsymbol{y}\|_0 = |supp(\boldsymbol{y})|$, where $supp(\boldsymbol{y}) = \{i : y_i \neq 0\}$ ($supp$ stands for support). This optimization problem finds the sparsest nonzero vector $\boldsymbol{y}$ in the null space of H, which is nothing but the smallest circuit over $M[\boldsymbol{H}]$. Solving l_0-minimization problems is NP-hard in general. An alternate approach proposed in compressed sensing is to replace the l_0-norm with the l_0-norm. The l_1-minimization problem is as follows.

$$\begin{aligned} & \min \|\boldsymbol{y}\|_1 \\ \text{subject to} \quad & \boldsymbol{H}\boldsymbol{y} = \boldsymbol{0} \\ & \boldsymbol{y} \neq \boldsymbol{0} \end{aligned}$$

Note that, $\|\boldsymbol{y}\|_1 = \sum_{i=1}^{n} |y_i|$. This l_1-minimization problem can be easily reformulated as a linear programming problem. The l_1-norm can be considered as a convex approximation of the l_0-norm. However, the solutions to both these optimization algorithms may nit be the same. The conditions under which the solution to l_1-minimization problem is also the solution to the l_0-minimization problem is extensively studied in [33, 34]. These conditions require the columns of the matrix $\boldsymbol{H}$ to follow some normal distribution with a uniform measure. However, the design matrix that models a linear sensor system generally does not satisfy these conditions.

In [33], Donoho et al. proposed solving a series of l_1-minimization problems to obtain good estimates for the l_0-minimization problem. They proposed solving the optimization problem

$$\begin{aligned} & d_i = \min \|\boldsymbol{y}\|_1 \\ \text{subject to} \quad & \boldsymbol{H}\boldsymbol{y} = \boldsymbol{0} \\ & y_i = 1 \end{aligned}$$

n times, each with the constraint $y_i = 1, i = 1, 2, ..., n$. Then,

$$g(M[\boldsymbol{H}]) \leq \min_{1 \leq i \leq n} d_i. \tag{3.1}$$

(3.1) gives an upper bound to the value of $\|\boldsymbol{y}\|_0$ and hence the girth $g(M[\boldsymbol{H}])$. In our computational experience, we have noticed although this approach leads to relatively good approximate solutions, in most cases, the optimal l_0-norm solution is not obtained.

We extended the approach introduced in [33] solving additional l_1-minimization problems to strengthen the upper bound obtained by (3.1). We call this algorithm 2-STAGEL1, because we solve a series of l_1-minimization problems in two stages. Motivation for this technique was first obtained empirically when the solutions to this problem were also the solutions to the l_0-minimization problem. We present the algorithm below.

2-STAGEL1*: Estimate the degree of redundancy of* $\boldsymbol{A}$

Input: Matrix $\boldsymbol{A} \in \mathbb{R}^{n \times m}$

Use RREF to transform $\boldsymbol{A}^T$ *to* $[\boldsymbol{I}_m|\boldsymbol{D}]$ *and get the dual* $\boldsymbol{H} = [-\boldsymbol{D}^T|\boldsymbol{I}_{n-m}]$. *Set* $\boldsymbol{h}_i = \boldsymbol{h}_i/\|\boldsymbol{h}_i\|_2$, *where* $\boldsymbol{h}_i$ *is the* i^{th} *column of* $\boldsymbol{H}$.

Let $LP(i)$ define the optimization problem below:

$$\min \|\boldsymbol{y}\|_1 \tag{3.2}$$

$$\text{subject to} \quad \boldsymbol{H}\boldsymbol{y} = \boldsymbol{0} \tag{3.3}$$

$$y_i = 1 \tag{3.4}$$

$$-1 \leq \boldsymbol{y} \leq \boldsymbol{1} \tag{3.5}$$

Let $LP(i, j), i \neq j$ define the optimization problem below:

$$\min \|\boldsymbol{y}\|_1 \tag{3.6}$$

$$\text{subject to} \quad \boldsymbol{H}\boldsymbol{y} = \boldsymbol{0} \tag{3.7}$$

$$y_i = 1, y_j = 0 \tag{3.8}$$

$$-1 \leq \boldsymbol{y} \leq \boldsymbol{1} \tag{3.9}$$

```
Step 0. $d^1 \leftarrow n$, $d^2 \leftarrow n$, $d^* \leftarrow n$
Step 1. for $i \leq i \leq n$
```

```
        solve $LP(i)$ to get the optimal solution $\boldsymbol{x}^*$
        $d^* \leftarrow \|\boldsymbol{x}^*\|_0$ and $D^* \leftarrow supp(x^*) \setminus i$
        if $d^* < d^1$
            $d^1 \leftarrow d^*$ and $C^1 \leftarrow supp(x^*)$
        if $d^1 < d^2$
            $d^2 \leftarrow d^1$ and $C^2 \leftarrow C^1$
Step 2. for each $j \in D^*$
        solve $LP(i,j)$ to get the optimal solution $\boldsymbol{x}^*$
        $d^* \leftarrow \|\boldsymbol{x}^*\|_0$
        if $d^* < d^2$
            $d^2 \leftarrow d^*$ and $C^2 \leftarrow supp(x^*)$
Step 3. get $d(\hat{\boldsymbol{A}}) = d^2 - 1$
```

The algorithm 2-STAGEL1 obtains the dual matroid $\boldsymbol{H}$ and solves two sets of optimization problems. The first one, $LP(i)$, is defined in (3.2) - (3.5), and the other, $LP(i,j)$ is defined in (3.6) - (3.9). $LP(i)$ solved for all $i \in \{1, 2, ..., n\}$ is the same as the one solved in [33] with (3.5) added as an additional constraint. We call this as Stage 1 (given in Step 1). For each optimization problem $LP(i)$, we solve another set of optimization problems $LP(i,j)$ which we call as Stage 2 (given in Step 2). $LP(i,j)$ is a modified version of $LP(i)$, with an additional constraint $y_j = 0$ as given in (3.8). The values that j is set to are the indices of the nonzero components of the optimal solution to the $LP(i)$ problem. The set D^* is set to these indices. Note that, i is not included in D^*.

C^1 tracks the support of the smallest l_0-norm solution among all the Stage 1 optimization problems. C^1 obtains the smallest l_0-norm (sparsest) solution among all the Stage 1 and Stage 2 optimization problems. Then, d^1 gives the l_0-norm of the sparsest solution after Stage 1 and d^2 the one after Stage 1 and Stage 2. Hence, $d(\hat{\boldsymbol{A}}) = d^2 - 1$ is an estimate of the degree of redundancy of

$\boldsymbol{A}$. Also, $d^1 - 1$ gives the redundancy estimate from the Stage 1. Note that, C^1 and C^2 contains at least one circuit over $M[\boldsymbol{H}]$, and hence a cocircuit over $M[\boldsymbol{A}^T]$. The optimization problems $LP(i)$ and $LP(i, j)$ can easily be re-written as LPs and hence computationally very efficient. From the way we defined C^1 and C^2, $|C^2| \leq |C^2|$.

The idea for 2-STAGEL1, conceived based on empirical results, relies on the highly non-linear structure of the set of all k-sparse vectors in a given Euclidian space. Since no theoretical guarantees of optimality of $d(\boldsymbol{A})$ can be provided for the algorithm, we briefly explain our intuition behind this approach. Consider the n-dimensional Euclidian space $\mathbb{R}^n$ with unit vectors $\boldsymbol{e}_1, \boldsymbol{e}_2, ..., \boldsymbol{e}_n$ as basis. The set of all k-sparse vectors in $\mathbb{R}^n$ is given by $\mathfrak{U} = \{\boldsymbol{y} \in \mathbb{R}^n : \|\boldsymbol{y}\|_0 \leq k\}$. This set consists of a union of a union of $\binom{n}{k}$ subspaces, where each subspace is spanned by a unique choice of k out of these n unit vectors. This union operation makes the set $\mathfrak{U}$ highly non-linear. Stage 1 finds a vector in the null space of $\boldsymbol{H}$ with the smallest l_1-norm. This vector gives an upper bound solution to the l_0-norm optimization problem. 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 shows that by introducing this second stage, we are able to find exact optimal solutions to many instances of the girth problem (degree of redundancy problem) with known solutions which were otherwise not solved optimally by Stage 1.

3.2 From Hueristic to Optimal Solution

The heuristic solution to $d(\boldsymbol{A})$ does not guarantee an optimal solution. Empirical evidence based on our experiments suggest that the solution is extremely close to the optimal solution. Advanced optimization solvers allows for providing starting solutions to warm-start an MIP problem. Warm-starting, i.e., providing a good initial solution to an MIP solver, can have significant impact on the solution time of MIP as it can potentially provide much stronger upper bounds for the minimization problem from the beginning. We can easily provide a starting solution to MIP$_{\text{CO}}$ based on the set C^2 that contains a cocircuit over $M[\boldsymbol{A}^T]$. However, an initial solution to MIP$_{\text{CO}}$ requires variables $(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{z})$. We can provide an initial solution to $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_n)$ as $\alpha_j = 1$

if $j \in C^2$, otherwise $\alpha_j = 0$. So, 2-STAGEL1 gives a partial starting solution to MIP$_{\text{CO}}$. To obtain a complete solution, we need to find an $\boldsymbol{x} \in \mathbb{R}^m$ and a $\boldsymbol{z} \in \{0, 1\}^m$ corresponding to this partial solution that has only values for $\boldsymbol{\alpha}$. For this, we use the capabilities of our MIP solver, CPLEX. When warm-starting an MIP using CPLEX, one can identify the values for part of the variables and ask CPLEX to form a complete feasible solution by finding the proper values for the other variables. In our computations, we used this capability by providing an $\boldsymbol{\alpha}$ as calculated above and letting CPLEX to find the proper $\boldsymbol{x}$ and $\boldsymbol{z}$ to form a warm-start solution.

The algorithm MIP$_{\text{CO-W}}$ (where ‘w’ stands for warm-starting) presented below warm-starts MIP$_{\text{CO}}$ with the solution $\boldsymbol{\alpha}$ obtained from 2-STAGEL1.

MIP$_{\text{CO-W}}$*: Compute the degree of redundancy of* $\boldsymbol{A}$

Input: Matrix $\boldsymbol{A} \in \mathbb{R}^{n\times m}$ *with blocks* $\boldsymbol{A}_1, ..., \boldsymbol{A}_r$*, and separating set* S.

Step 1. *Solve* 2-STAGEL1 *to find an upper bound solution* C^2 *to the cocircuit over* $M[\boldsymbol{A}^T]$.

Step 2. *Get an initial solution* $\boldsymbol{\alpha}^0$ *to* MIP$_{\text{CO}}$ *as* $\boldsymbol{\alpha} = (\alpha_1^0, \alpha_2^0, ..., \alpha_n^0)$ *as* $\alpha_j^0 = 1$ *if* $j \in C^2$*, otherwise* $\alpha_j^0 = 0$.

Step 3. *Solve* MIP$_{\text{CO}}$ *using CPLEX with* $\boldsymbol{\alpha}^0$ *as a partial warm-start solution to find* $d(\boldsymbol{A})$.

Note that, providing a partial solution is not as efficient as providing a complete solution. Although we observed computational gains of the order of 10% when tested over some instances, this approach is still inefficient to solving large-scale problems.

4. COMPUTING THE DEGREE OF REDUNDANCY: A COGIRTH BASED DECOMPOSITION APPROACH

4.1 A Decomposition Theorem for Smallest Cocircuit

We start by considering a matroid $M(E, \mathcal{I})$, with a separating set $S \subset E(M)$. We assume $M \setminus S$ to be disconnected with direct sum components $M_1, M_2, ..., M_r$ such that $M \setminus S = M_1 \oplus M_2 \oplus ... \oplus M_r$.

For $i = 1, ..., r$, let

$$g^*(M) = \text{ min } \{|D| : D \in \mathcal{C}^*(M)\}, \text{ and} \tag{4.1}$$

$$g^*(M_j) = \text{ min } \{|D_j| : D_j \in \mathcal{C}^*(M_j)\}. \tag{4.2}$$

$g^*(M)$ and $g^*(M_j)$ gives the values of the cardinality of the smallest cocircuits in M and M_j respectively. Hence, by definition of cogirth, $g^*(M)$ and $g^*(M_j)$ are the cogirths of M and M_j, respectively.

We now present the Lemma 4 that gives an upper bound on the cogirth of a connected matroid based on the cogiths of its direct sum components.

Lemma 4. *Suppose $M(E, \mathcal{I})$ be such that $M \setminus S = M_1 \oplus M_2 \oplus ... \oplus M_r$. Define*

$$g^*(M_{(1)}) = \underset{j \in \{1,2,...,r\}}{\textit{min}} g^*(M_j).$$

Then, $g^*(M) \leq g^*(M_{(1)}) + |S|$.

Proof. Let $D_j \in \mathcal{C}^*(M_j)$. We first show that $D_j \cup S$ is a codependent set in M.

Assume $D_j \cup S$ to be a coindependent set in M. Then, $E(M) - (D_j \cup S)$ will be a spanning set in M. Hence, there exists a base B in M such that

$$B \subseteq E(M) - (D_j \cup S).$$

Also, since $B \subseteq E(M \setminus S)$, B is a base in $M \setminus S$. Then, by the properties of direct sum, there exists respective bases $B_1, B_2, ..., B_r$ in $M_1, M_2, ..., M_r$, such that $B = B_1 \cup B_2 \cup ... \cup B_r$.

Now consider the base B_j in M_j. Indeed $B_j \subseteq E(M_j) - D_j$ since $B \subseteq E(M \setminus S) - D_j$. But this contradicts our assumption that D_j is a cocircuit in M_j, since that makes $E(M_j) - D_j$ a non spanning set in M_j. Therefore, $D_j \cup S$ is a codependent set in M. Then, by definition of cogirth,

$$g^*(M) \leq |D_j \cup S| = |D_j| + |S|$$

Since this is true for any $D_j \in \mathcal{C}^*(M_j)$,

$$g^*(M) \leq g^*(M_j) + |S|, \ j = 1, 2, ..., r \text{ and hence}$$

$$g^*(M) \leq g^*(M_{(1)}) + |S|.$$

□

Before we prove the next lemma, let us define some of the notations that we use throughout this report.

For $J \subseteq \{1, ..., r\}$, let

$$\mathcal{C}^*_J(M) = \{D \in \mathcal{C}^*(M) : D \subseteq \bigcup_{j \in J} E(M_j) \cup S\},$$

$$\tilde{c}^*_J(M) = \min \{|D| : D \in C^*_J(M)\}, \text{ and}$$

$$c^*_J(M) = \min \{|D| : D \in C^*_J(M) \text{ and } D \cap E(M_j) \neq \emptyset, \text{ for all } j \in J\}.$$

For simplicity, we sometimes refer to $\tilde{c}^*_J(M)$ and $c^*_J(M)$ as $\tilde{c}^*_J$ and c^*_J, respectively.

Let $\mathcal{J}_t$ be the collection of all subsets of $\{1, 2, ..., r\}$ containing exactly t elements. Also, let $\mathcal{J}_{\leq t}$ and $\mathcal{J}_{\geq t}$ be the collection of subsets of $\{1, 2, ..., r\}$ containing at most and at least t elements, respectively.

We now present another lemma that connects the cociruits over a vector matroid to the cocircuits over its direct sum components.

Lemma 5. *Suppose $M \setminus S = M_1 \oplus M_2 \oplus ... \oplus M_r$ and D be a cocircuit in M where $D \in \mathcal{C}^*_J(M)$ for some $J \subseteq \{1, ..., r\}$. Then, for any $j \in J$ with $D \cap E(M_j) \neq \emptyset$,*

$$|D \cap E(M_j)| \geq g^*(M_j).$$

Proof. Let $Q \in D$ be chosen at random such that $Q \cap E(M_j) \neq \emptyset$, for all $j \in J$. Get $Q_j = Q \cap E(M_j)$ and $Q_s = Q \cap S$. Then, it is trivial that

$$\left| \bigcup_{j \in J} Q_j \cup Q_s \right| = \sum_{j \in J} |Q_j| + |Q_S|.$$

We now show that

$$\sum_{j \in J} |Q_j| + |Q_s| \geq \sum_{j \in J} g^*(M_j). \tag{4.3}$$

Assume that the above inequality does not hold. Then, there should be atleast one $l \in J$ such that $|Q_l| < g^*(M_l)$. Hence, Q_l has to be a coindependent set in M_l, and thus $E(M_l) - Q_l$ is a spanning set in M_l. We can now find a base

$$B_l \subseteq E(M_l) - Q_l \text{ in } M_l.$$

Let $P = \bigcup_{j \in J \setminus \{l\}} Q_j \cup Q_s$. We know, B_l is an independent set in M_l. Therefore, B_l is an independent set in $M \setminus P$. Now, by Lemma 2 we can find a base $\tilde{B}$ in $M \setminus P$, such that $B_l \subseteq \tilde{B}$.

Q_l is a non-empty set, and for this reason P is a coindependent set in M. Therefore, $E(M) - P$ is a spanning set in M. Clearly, this makes $\tilde{B}$ a base in M.

Then, the condition $Q_l \cap \tilde{B} \neq \emptyset$ should be satisfies, as otherwise

$$\tilde{B} \subseteq E(M) - (P \cup Q_l) = E(M) - Q,$$

which is not a spanning set in M.

It follows that there exists an $e \in Q_l \cap \tilde{B}$ such that $e \notin B_l$. We now have a contradiction. This is because $B_l \cup \{e\}$ is indeed a dependent set, which disputes our claim that $\tilde{B}$ is a base. Therefore, Q_l is a codependent set in M_l and evidently, $|Q_l| \geq g^*(M_l)$.

Consequently, $|Q_j| \geq g^*(M_j)$ for all j and (4.3) holds. Since (4.3) is true for any Q, by definition of c^*_J we can conclude that

$$c^*_J \geq \sum_{j \in J} g^*(M_j).$$

It is obvious that we can extend (4.3) to any $D \in \mathcal{C}^*_J(M)$, by excluding all $D_j = D \cap E(M_j)$ for which $D_j = \emptyset$ from either sides of the summation and hence proves the lemma. □

Using Lemma 4 and Lemma 5, we now prove Theorem 2, the decomposition theorem that forms the basis of our cocircuit based decomposition algorithm to find the degree of redundancy.

Theorem 2 (Decomposition Theorem for Cogirth). *Given a matroid $M \setminus S = M_1 \oplus M_2 \oplus ... \oplus M_r$. Let the cogirths $g^*(M_j)$, $j = 1, ..., r$, be ordered such that*

$$g^*(M_{(1)}) \leq g^*(M_{(2)}) \leq ... \leq g^*(M_{(r)}).$$

Get $t = \underset{l}{min} \left\{ l - 1 : \sum_{j=1}^{l} g^*(M_{(j)}) \geq g^*(M_{(1)}) + |S| \right\}$. *Then:*

(a) $g^*(M) = \underset{J \in \mathcal{J}_t}{min} \tilde{c}^*_J$, *and*

(b) for any $J \in \mathcal{J}_t$, *if* $\sum_{j \in J} g^*(M_j) > g^*(M_{(1)}) + |S|$, *then* $g^*(M) < c^*_J$.

Proof. By the definition of cogirth,

$$g^*(M) = \tilde{c}^*_{\{1,2,...,r\}}.$$

The value of t is chosen such that for any $J \in \mathcal{J}_{t+1}$,

$$\sum_{j \in J} g^*(M_{(j)}) \geq g^*(M_{(1)}) + |S|.$$

Then, for any $Y \in \mathcal{J}_{\geq t+1}$,

$$c_Y^* \geq \sum_{j \in Y} g^*(M_j) \geq \sum_{j=1}^{t+1} g^*(M_{(j)}) \geq g^*(M_{(1)}) + |S|,$$

where the first inequality follows from Lemma 5.

From Lemma 4,

$$\underset{i \in \{1,2,\ldots,r\}}{min} c_i^* \leq g^*(M_{(1)}) + |S|.$$

Hence, there exists some $X \in \mathcal{J}_{\leq t}$ such that

$$c_X^* \leq g^*(M_{(1)}) + |S|.$$

Therefore,

$$g^*(M) = \min_{J \in \mathcal{J}_{\leq t}} c_J^* = \min_{J \in \mathcal{J}_t} \tilde{c}_J^*.$$

Part (b) follows directly from Lemma 5. □

4.2 The Decomposition Algorithm: DMIP$_{\textbf{CO}}$

Consider the BBD structured design matrix $\boldsymbol{A}$ with a separating set S and direct sum components $\boldsymbol{A}_1, ..., \boldsymbol{A}_r$. As discussed in Section 2.2, the degree of redundancy $d(\boldsymbol{A}) = g^*(M[\boldsymbol{A}^T]) - 1$. Lemma 6, presented in [22], connects $\tilde{c}_J^*(M[\boldsymbol{A}^T])$ with the cogirth of the matriod over the $|J|$-block submatrix $\boldsymbol{A}[J]^T$, i.e., the submatrix consisting of all the blocks $\boldsymbol{A}_j^T$'s, $j \in J$, along with the respective border blocks $\boldsymbol{S}_j^T$'s.

Lemma 6. *If* $r(M[\boldsymbol{A}^T \setminus S]) = r(M[\boldsymbol{A}^T]),$ *then* $\tilde{c}_J^*(M[\boldsymbol{A}^T]) = g^*(M[\boldsymbol{A}[J]^T]).$

The above lemma can be easily proved from the fact that any cocircuit in $M[\boldsymbol{A}[J]^T]$ is a codependent set in $M[\boldsymbol{A}^T]$. Corollary 1 applies Theorem 2 to $M[\boldsymbol{A}^T]$ and uses Lemma 6.

Corollary 1. *Given a matroid $M[\boldsymbol{A}^T]$ with $M[\boldsymbol{A}^T \setminus S] = M[\boldsymbol{A}_1^T] \oplus M[\boldsymbol{A}_2^T] \oplus ... \oplus M[\boldsymbol{A}_r^T]$. Let the cogirths $g^*(M[\boldsymbol{A}_i^T])$, $i \in \{1, ..., r\}$ be ordered such that*

$$g^*(M[\boldsymbol{A}_{(1)}^T]) \leq g^*(M[\boldsymbol{A}_{(2)}^T]) \leq ... \leq g^*(M[\boldsymbol{A}_{(r)}^T]).$$

Get $t = \ min \left\{ l-1 : \sum_{i=1}^{l} g^*(M[\boldsymbol{A}_{(i)}^T]) \geq g^*(M[\boldsymbol{A}_{(1)}^T]) + |S| \right\}$. *Then:*

(a) $g^*(M[\boldsymbol{A}^T]) = \min_{J \in \mathcal{J}_t} g^*(M[\boldsymbol{A}[J]^T])$, *and*

(b) for any $J \in \mathcal{J}_t$, *if* $\sum_{j \in J} g^*(M[\boldsymbol{A}_j^T]) > g^*(M[\boldsymbol{A}_{(1)}^T]) + |S|$, *then* $g^*(M[\boldsymbol{A}^T]) < c_J^*(M[\boldsymbol{A}^T])$.

The proof is omitted as it is straightforward from Theorem 2 and Lemma 6.

In our first decomposition algorithm, denoted by DMIP$_{\text{CO}}$, we decompose the problem of finding the cogirth over $M[\boldsymbol{A}^T]$ into subproblems over t-block submatrices of $\boldsymbol{A}^T$ with the value of t obtained from Corollary 1. The subproblems are then solved using MIP$_{\text{CO}}$ ([14]), thereby obtaining the cogirth over $M[\boldsymbol{A}^T]$, and hence, the degree of redundancy of $\boldsymbol{A}$. We now present this algorithm.

DMIP$_{\text{CO}}$: *Compute the degree of redundancy of* $\boldsymbol{A}$

Input: Matrix $\boldsymbol{A} \in \mathbb{R}^{n \times m}$ with blocks $\boldsymbol{A}_1, ..., \boldsymbol{A}_r$, and separating set S.

```
1. find $g^*(M[\boldsymbol{A}_i^T])$ for all $i \in \{1, 2, ..., r\}$
2. sort the cogirths such that $g^*(M[\boldsymbol{A}_{(1)}^T]) \leq g^*(M[\boldsymbol{A}_{(2)}^T]) \leq ... \leq g^*(M[\boldsymbol{A}_{(k)}^T])$
```

```
3. get $t = min\left\{ l-1 : \sum_{i=1}^{l} g^*(M[\boldsymbol{A}^T_{(i)}] \geq g^*(M[\boldsymbol{A}^T_{(1)}]) + |S| \right\}$
4. set $\mathcal{F} = \{J : J \in \mathcal{J}_t\}$, $d = g^*(M[\boldsymbol{A}^T_{(1)}]) + |S|$, $\mathcal{L} = \emptyset$
5. if $\mathcal{F} = \emptyset$, stop; return $d - 1$
6. for each $J \in \mathcal{F}$,
       if $J \in \mathcal{L}$, $\mathcal{F} = \mathcal{F} \setminus \{J\}$
       else if $\sum_{j \in J} g^*(M[\boldsymbol{A}^T_j]) > d$,
           update $\mathcal{F} = \mathcal{F} \bigcup_{j \in J} \{J - \{j\}\} \setminus \{J\}$
       else solve $\text{MIP}_{\text{CO}}$ over $\boldsymbol{A}[J]^T$, get $d_J = g^*(M[\boldsymbol{A}[J]^T])$;
           update $\mathcal{L} = \{J : J \in \mathcal{J}_{\leq |J|}\}$, if $d_J < d$, set $d = d_J$
7. go back to step 5
```

In algorithm DMIP_{CO}, we first obtain the cogirths of $M[\boldsymbol{A}^T_1], ..., M[\boldsymbol{A}^T_r]$, then find the value of t using Theorem 2. Now, as Corrollary 1 presents, we can find the cogirth of $M[\boldsymbol{A}^T]$ by finding the minimum of the cogirths over all vector matroids defined over the t-block submatrices of $\boldsymbol{A}^T$. In DMIP_{CO}, we solve MIP_{CO} over these submatrices to find their cogirths, and then continuously update d with the smallest cogirth obtained so far. The set $\mathcal{J}$, initialized with all subsets of $\{1, 2, ..., r\}$ of cardinality t, includes all the submatrices that need to be solved for. However, instead of finding the cogirths over all t-block submatrices, we use proposition (b) in Corrollary 1 to avoid solving for any submatrix J for which

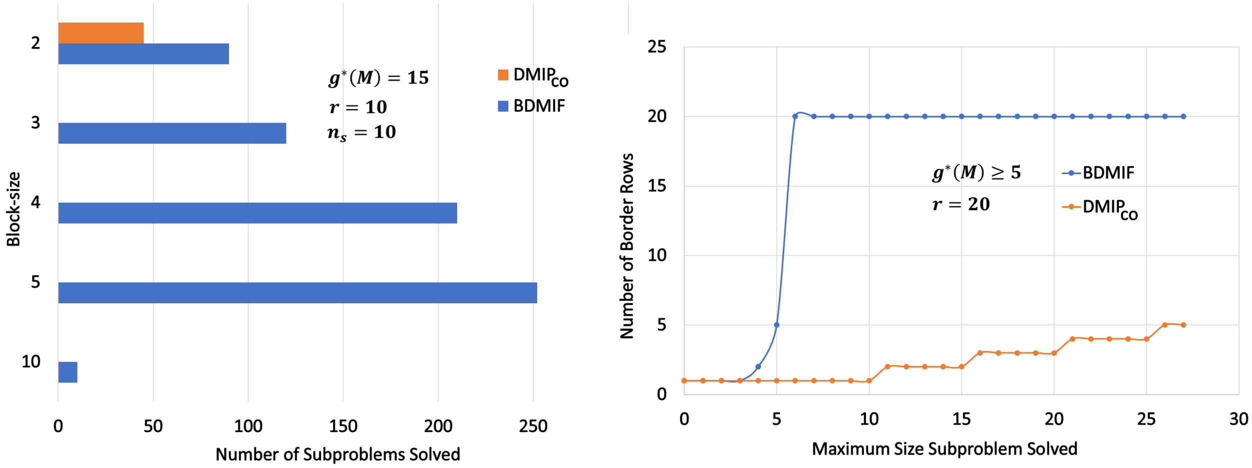
$$\sum_{j \in J} g^*(M[\boldsymbol{A}^T_j]) > d.$$

We remove any such J from $\mathcal{F}$, and then update it with all subsets of J with $|J| - 1$ elements. This allows us to further decompose the subproblems into even smaller ones.

The following remark helps to further reduce the number of instances of MIP$_{\text{CO}}$ that we need to solve.

Remark 5. *For two submatrices $\boldsymbol{A}[X]^T$ and $\boldsymbol{A}[Y]^T$ of $\boldsymbol{A}^T$, where $X, Y \subseteq \{1, 2, .., r\}$, and $X \subset Y$, then $g^*(M[\boldsymbol{A}[Y]^T]) \leq g^*(M[\boldsymbol{A}[X]^T])$.*

As Remark 5 indicates, we can avoid solving for any submatrix $\boldsymbol{A}[J]$ for which another MIP$_{\text{CO}}$ instance over a matrix having $\boldsymbol{A}[J]$ as a sub-block has already been solved for. We use the set $\mathcal{L}$ to include the submatrices that has already been solved for directly, or indirectly based on Remark 5, which are then excluded from $\mathcal{F}$ in Step 6.



(a) Number of subproblems of each block-size solved when $g^*(M[\boldsymbol{A}^T]) = 15$

(b) A sample analysis of DMIP$_{\text{CO}}$ against BDMIF

Figure 4.1: A Comparison of Decomposition Theorems

The number of subproblems and the size of the subproblems that needs to be solved for any algorithm based on our decomposition theorem, Theorem 1, is far less than those that are based on the bound and decompose theorem ([2]). A simple analysis of DMIP$_{\text{CO}}$ against BDMIF demonstrates this advantage. Figure 4.1(a) shows the number of subproblems of each size that needs to be solved for each of the algorithms when a design matrix $\boldsymbol{A}$ with 10 blocks and border rows is considered with $g^*(M[\boldsymbol{A}^T]) = 15$. For any design matrix with these parameters, the number of subproblems that need to be solved for BDMIF is the same except for 2-block submatrices which can be anywhere between 46 to 90. The reason is as follows: When MIF checking procedure is executed with $d = 14$, all 45 2-block submatrices (the smallest possible based on the decomposition bound) are tested without finding a feasible solution. When $d = 15$ is tested, we still need to solve for the 2-block submatrices, and atleast one of these 45 submatrices give a feasible solution for MIF.

The number of subproblems to be solved and their block-sizes for MIP_{CO} depends on the cogirths of each of the $M[\boldsymbol{A}_i^T]$'s. For example, if $g^*(M[\boldsymbol{A}_{(1)}^T]) = 5$ and $g^*(M[\boldsymbol{A}_{(2)}^T]) = 10$, then we need to solve only 1-block submatrices, a total of 10, to find the degree of redundancy using MIP_{CO}. Based on (b) in Corollary 1 we might even be able to further reduce the number of subproblems solved. One of the worst cases occur when $g^*(M[\boldsymbol{A}_{(i)}^T]) = 5$ for all i's. In such a case, we need to solve a maximum of 45 2-block subproblems, which is much less that the total number of subproblems that needs to be solved for BDMIF. Notice that, the larger the value of $M[\boldsymbol{A}_{(i)}^T]$'s, the smaller the value of t obtained from Corrollary 1, and smaller the size of the subproblems that we need to solve. This result is extremely significant, because design matrices with larger denser blocks tend to have larger values of $g^*(M[\boldsymbol{A}_{(i)}^T])$'s thereby reducing the block-sizes of subproblems that need to be solved.

Another important parameter that needs to be considered is the dependence of the block-size of the subproblems to that of the number of border rows n_s. Figure 4.1(b) shows the maximum block-size of the subproblem that needs to be solved for a BBD matrix $\boldsymbol{A}$ with 20 blocks and assuming $g^*(M[\boldsymbol{A}^T]) \geq 5$. As before, the values for MIP_{CO} changes based on the cogirths of $g^*(M[\boldsymbol{A}_{(i)}^T])$'s and the worst case values are included in Figure 4.1(b). As the number of border rows increases, the maximum size subproblems that needs to be solved for both the algorithms increases. However, the values for MIP_{CO} are much smaller than those for BDMIF even in the worst case. Also, as the value of n_s increases, the maximum subproblem size to be solved for MIP_{CO} increase much slowly as compared to BDMIF. These analysis shows why our algorithm is well-suited for problems in large scale.

5. COMPUTING THE DEGREE OF REDUNDANCY: A GIRTH BASED DECOMPOSITION APPROACH

5.1 A Decomposition Theorem for Smallest Circuit

Finding the girth of the dual matroid $M^*[\boldsymbol{A}^T]$, defined as $M[\boldsymbol{H}]$, as we discussed in Section 2.2, solves for the redundancy degree problem over $\boldsymbol{A}$. The girth problem can be formulated as an MIP problem. We call this MIP formulation MIP$_{\text{CIR}}$, where 'cir' stands for circuit. A dual approach to compute the degree of redundancy is to solve for the girth over $M[\boldsymbol{H}]$, the dual of $M[\boldsymbol{A}^T]$, as given by (2.9).

MIP$_{\text{CIR}}$: *Smallest Circuit Problem*

$$g(M[\boldsymbol{H}]) = \min \mathbf{1}\boldsymbol{\beta} \tag{5.1}$$

$$\text{subject to} \quad \boldsymbol{H}\boldsymbol{y} = \mathbf{0} \tag{5.2}$$

$$-\boldsymbol{\beta} + 2\boldsymbol{z} \leq \boldsymbol{y} \leq \boldsymbol{\beta} \tag{5.3}$$

$$\mathbf{1}\boldsymbol{z} = 1 \tag{5.4}$$

$$\boldsymbol{y} \in \mathbb{R}^n, \ \boldsymbol{\beta} \in \{0,1\}^n, \ \boldsymbol{z} \in \{0,1\}^n \tag{5.5}$$

The formulation MIP$_{\text{CIR}}$ looks for a nonzero $\boldsymbol{y}$ with the smallest l_0 norm in the null space of $\boldsymbol{H}$. The constraints (5.3) and (5.4) make sure that $\boldsymbol{y}$ is nonzero. The objective function (5.1) minimizes the number of y_i's for which $y_i \neq 0$ thereby finding a $\boldsymbol{y}$ with the smallest l_0 norm. The set of indices i for which $\beta_i = 1$ gives the smallest circit of $M[\boldsymbol{H}]$, and hence $d(\boldsymbol{A}) = \mathbf{1}\boldsymbol{\beta} - 1$. Solving MIP$_{\text{CIR}}$ for large-scale problems is challenging. However, we can employ the dual version of the decomposition theorem, Theorem 1, to develop a decomposition approach for this dual problem. This dual theorem is presented over the dual matroid M^*.

For $J \subseteq \{1, ..., r\}$, let

$$\begin{aligned}
&\mathcal{C}_J(M^*) = \{D \in \mathcal{C}(M^)) : D \subseteq \bigcup_{j \in J} E(M_j^*) \cup S\},\\
&\tilde{c}_J(M^*) = \text{ min } \{|D| : D \in C_J(M^*)\}, \text{ and}\\
&c_J(M^*) = \text{ min } \{|D| : D \in C_J(M^*) \text{ and } D \cap E(M_j^*) \neq \emptyset, \text{ for all } j \in J\}.
\end{aligned}$$

For simplicity, we sometimes refer to $\tilde{c}_J(M^*)$ and $c_J(M^*)$ as $\tilde{c}_J$ and c_J, respectively. These definitions are same as the ones that we defined for cocircuits over M.

Given the matroid M with a separating set S, $M^*/S = M_1^* \oplus M_2^* \oplus ... \oplus M_r^*$. Now, we present the decomposition theorem for girth over the dual matroid M^*. This theorem directly follows from Theorem 1 based on the duality properties of a matroid. Hence, the proof is omitted.

Theorem 3 (Decomposition Theorem for Girth)**.** *Given a matroid $M \setminus S = M_1 \oplus M_2 \oplus ... \oplus M_r$. Let the girths $g(M_j^*)$, $j = 1, ..., r$, be ordered such that*

$$g(M_{(1)}^*) \leq g(M_{(2)}^*) \leq ... \leq g(M_{(r)}^*).$$

Get $t = \underset{l}{min} \left\{ l - 1 : \sum_{j=1}^{l} g(M_{(j)}^*) \geq g(M_{(1)}^*) + |S| \right\}$. *Then:*

$$\begin{aligned}
&(a)\ g(M^*) = \underset{J \in \mathcal{J}_t}{min}\, \tilde{c}_J, \textit{ and}\\
&\textit{(b) for any } J \in \mathcal{J}_t, \textit{ if } \sum_{j \in J} g(M_j^*) > g(M_{(1)}^*) + |S|, \textit{ then } g(M^*) < c_J.
\end{aligned}$$

Now, we present a lemma that hepls to re-write Theorem 3 with respect to the vector matroid $M[\boldsymbol{H}^T]$.

Lemma 7. *If $r(M[\boldsymbol{H}^T/S]) = r(M[\boldsymbol{H}^T])$, then $\tilde{c}_J(M[\boldsymbol{H}^T]) = g(M[\boldsymbol{H}[J]^T])$.*

The above lemma can be easily proved from the fact that any circuit in $M[\boldsymbol{H}[J]^T]$ is a dependent set in $M[\boldsymbol{H}^T]$. The condition $r(M[\boldsymbol{H}^T/S]) = r(M[\boldsymbol{H}^T])$ directly follows from our assumption that S is a coindependent set in $M[\boldsymbol{A}^T]$. Now, we present a corollary to Theorem 7

Corollary 2. *Given a matroid $M[\boldsymbol{A}^T]$ with a coindependent separating set S and the dual $M[\boldsymbol{H}^T]$ such that $M[\boldsymbol{H}/S] = M[\boldsymbol{H}_1] \oplus M[\boldsymbol{H}_2] \oplus ... \oplus M[\boldsymbol{H}_r]$. Let the girths $g(M[\boldsymbol{H}_i])$, $i \in \{1, ..., r\}$ be ordered such that*

$$g(M[\boldsymbol{H}_{(1)}]) \leq g(M[\boldsymbol{H}_{(2)}]) \leq ... \leq g(M[\boldsymbol{H}_{(r)}]).$$

Get $t = \ min \left\{ l - 1 : \sum_{i=1}^{l} g(M[\boldsymbol{H}_{(i)}]) \geq g(M[\boldsymbol{H}_{(1)}]) + |S| \right\}$. *Then:*

(a) $g(M[\boldsymbol{H}]) = \underset{J \in \mathcal{J}_t}{min}\, g(M[\boldsymbol{H}[J]])$, *and*

(b) for any $J \in \mathcal{J}_t$, *if* $\sum_{j \in J} g(M[\boldsymbol{H}[J]]) > g(M[\boldsymbol{H}_{(1)}]) + |S|$, *then* $g(M[\boldsymbol{H}]) < c_J(M[\boldsymbol{H}])$.

The proof of Corollary 2 is omitted as it directly results from applying Lemma 7 Theorem 3.

5.2 The Decomposition Algorithm: DMIP$_{\text{CIR}}$

The algorithm DMIP$_{\text{CIR}}$ applies Theorem 3 over the dual matrix $\boldsymbol{H}$ and solves the decomposed subproblems using MIP$_{\text{CIR}}$ to find the girth of $\boldsymbol{H}$, and then gets the degree of redundancy $d(\boldsymbol{A}) = g(M[\boldsymbol{H}]) - 1$. The algorithm is presented below.

DMIP$_{\text{CIR}}$: *Computing the degree of redundancy for $\boldsymbol{A}$ using the dual $\boldsymbol{H}$*

```
Input: Matrix $\boldsymbol{A} \in \mathbb{R}^{n \times m}$ with blocks $\boldsymbol{A}_1, ..., \boldsymbol{A}_r$, and separating set $S$.
1. get the dual $\boldsymbol{H} = [-\boldsymbol{D}^T | \boldsymbol{I}_{n-m}]$ from $RREF(\boldsymbol{A}^T)$ given by $[\boldsymbol{D} | \boldsymbol{I}_m]$
2. find $g(M[\boldsymbol{H}_i])$ for all $i \in \{1, 2, ..., r\}$
3. sort the girths such $g(M[\boldsymbol{H}_{(1)}]) \leq g(M[\boldsymbol{H}_{(2)}]) \leq ... \leq g(M[\boldsymbol{H}_{(k)}])$
4. get $t = \ min \left\{ l - 1 : \sum_{i=1}^{l} g(M[\boldsymbol{H}_{(i)}] \geq g(M[\boldsymbol{H}_{(1)}]) + |S| \right\}$
5. set $\mathcal{F} = \{J : J \in \mathcal{J}_t\}$, $d = g(M[\boldsymbol{H}_{(1)}]) + |S|$, $\mathcal{L} = \emptyset$

6. if $\mathcal{F} = \emptyset$, stop; return $d - 1$

7. for each $J \in \mathcal{F}$,

        if $J \in \mathcal{L}$, then $\mathcal{F} = \mathcal{F} \setminus \{J\}$

        else if $\sum_{j \in J} g(M[\boldsymbol{H}[J]]) > d$,

            update $\mathcal{F} = \mathcal{F} \bigcup_{j \in J} \{J - \{j\}\} \setminus \{J\}$

        else solve MIP$_{\text{CIR}}$ over $\boldsymbol{H}[J]$, get $d_J = g(M[\boldsymbol{H}[J]])$;

            update $\mathcal{L} = \{J : J \in \mathcal{J}_{\leq |J|}\}$, if $d_J < d$, then set $d = d_J$

7. go back to step 5
```

DMIP$_{\text{CIR}}$ follows the same steps as that of DMIP$_{\text{CO}}$, except that it solves MIP$_{\text{CIR}}$ over the submatrices of $\boldsymbol{H}$ to find $d(\boldsymbol{A})$. Since Theorem 3 is the dual of Theorem 2, the number of subproblems solved in DMIP$_{\text{CIR}}$ and their blocks-sizes are exactly same as that of DMIP$_{\text{CO}}$. The use of matrix $\boldsymbol{H}$ which is of different dimension than that of $\boldsymbol{A}^T$ along with the use of DMIP$_{\text{CIR}}$ instead of DMIP$_{\text{CO}}$ will impact the running time of DMIP$_{\text{CIR}}$.

6. RELIABILITY EVALUATION: AN ENHANCED RECURSIVE VARIANCE REDUCTION APPROACH

6.1 A Cocircuit Based Approach

We discussed in Section 2.4 that the minimal cut sets obtained by RVR_{RREF} are often inferior which severely affects the accuracy of the reliability estimate. However, we can gain a significant improvement in the quality of the reliability estimate by replacing RREF_{CO} with the MIP algorithms. Let us consider a hypothetical RVR framework that finds minimal cut sets using the smallest cocircuits over $M[(\boldsymbol{A}^T/W) \setminus F]$ obtained by solving MIP_{CO}. Since we need to execute MIP_{CO} at every iteration, this approach will undoubtedly increase the running time of RVR_{RREF} algorithm. So, the improvement in the accuracy of the reliability estimate will be at the heavy cost of computational time. We also need to account for the additional computational time required to obtain $\boldsymbol{A}^T_{(-F)/W}$ at each iteration. Let us look into the deletion operation for sensors in F and the contraction operation for sensors in W, required to get $(\boldsymbol{A}^T/W) \setminus F$ from $\boldsymbol{A}^T$. Deletion operation, which can be performed in $O(1)$ time, has the added advantage of reducing the overall size of the problem. However, the pivot operation in contraction is more involved.

A better approach to account for fixed sensors is to include additional constraints in the formulation for MIP_{CO} as given below:

i. *for each sensor $s_i \in F$, set $\alpha_i = 1$,*

ii. *for each sensor $s_i \in W$, set $\alpha_i = 0$.*

To understand this approach, consider $(\boldsymbol{x}^*, \boldsymbol{\alpha}^*, \boldsymbol{z})$ the solution to MIP_{CO} obtained after solving this algorithm with α_i's fixed as above. Let D be such that

$$D = \{s_i : \alpha_i^* = 1, s_i \notin W\}.$$

We can now find the minimal cut sets over $M[(\boldsymbol{A}^T/W) \setminus F]$ as below:

i. *if* $|D| \leq n - |F| - k + 1$*, then* D *is a minimal cut set,*

ii. *else each subset of* D *with a cardinality of* $n - |F| - k + 1$ *is a minimal cut set.*

As we discussed in Section 2.3, fixing the values of an α_i to 1 has the equivalent effect of deleting the corresponding column from $\boldsymbol{A}^T$. Similarly, fixing an α_i to 0 forces MIP_{CO} to find a D that does not involve sensor s_i. Hence, the minimal cut set obtained from D will also be the minimal cut set over $M[(\boldsymbol{A}^T/W) \setminus F]$. However, our experimental results indicate that it is better to delete the rows representing sensors in F while solving MIP_{CO} since the deletion operation takes very small computational time, and the resulting matrix will have a smaller size. Still, these modifications can only result in a minor reduction in the running time of MIP_{CO}.

In Section 4.2, we utilized Theorem 2 to decompose MIP_{CO} into smaller subproblems while finding the smallest cocircuit and solved these subproblems providing a significant reduction in the running time. But Theorem 2 decomposes the cogirth (minimal cocircuit) problem. By integrating Theorem 2 within the RVR framework, we can get the smallest cocircuit over $(\boldsymbol{A}^T/W) \setminus F$ at each iteration. To achieve this we needs to solve DMIP_{CO} over $(\boldsymbol{A}^T/W) \setminus F$. Even though this approach reduces the size of the MIPs solved, at each iteration we need to solve a large number of them. Interestingly, we might not even need to find the smallest minimal cut set at each iteration to achieve a highly superior reliability estimate.

We propose an alternative approach in which a single MIP_{CO} instance over a submatrix of $\boldsymbol{A}^T$ is solved at each iteration to find a minimal cut set. However, to facilitate such a decomposition, we need to establish theorems that allow for disintegrating the problem of finding the cocircuits over a structured matroid, not just its smallest cocircuit. The minimal cut sets obtained by this approach might not be the smallest. However, since we find a minimal cut set at each iteration of RVR, the quality of the entire set of these minimal cut sets will more or less match the quality of the ones obtained from solving DMIP_{CO} over $(\boldsymbol{A}^T/W) \setminus F$. Hence the accuracy of the reliability estimate can be maintained, but the running time can be greatly reduced.

We now study the properties of cocircuits of a BBD structured matroid $M[\boldsymbol{A}^T]$. The following lemma is a direct results of the independent augmentation axiom.

Lemma 8. *If B_1 and B_2 are two distinct bases of a matroid M, then for every $e_1 \in B_1 \setminus B_2$ there exist some $e_2 \in B_2 \setminus B_1$ such that $B_1 \setminus \{e_1\} \cup \{e_2\}$ is a base of M.*

Now we present another lemma with will help to prove our decomposition theorem to disintegrate the reliability analysis problem.

Lemma 9. *Let $M \setminus S = M_1 \oplus M_2 \oplus ... \oplus M_r$, and let D be a cocircuit. Suppose $D_j = D \cap E(M_j)$, for $j = 1, 2, ..., r$ and $D_s = D \cap S$. Then D_j is a codependent set in M_j.*

Proof. We prove this by contradiction. Assume some $D_l \neq \emptyset$ is such that it is a coindependent set in M_l. Hence, $E(M_l) - D_l$ is a spanning set in M_l and there exists a base $B_l \subseteq E(M_l) - D_l$ in M_l. Let $P = \sum_{j \in J - \{l\}} D_j \cup D_s$. B_l, being a base, is an independent set in M_l. Then B_l is also an independent set in $M \setminus P$. So, by Lemma 4 there is a base $B' \in M \setminus P$, such that $B_l \subseteq B'$.

As D is a cocircuit and D_l is a non-empty set, P is a coindependent set in M. Therefore, $E(M) - P$ is a spanning set in M and $r(E(M) - P) = r(M)$. Thus, B' is also a base in M.

Let us assume $D_l \cap B' \neq \emptyset$. Then, there exists an $e \in D_l \cap B'$ such that $e \notin B_l$. But, since $e \in E(M_l)$ and B_l being a base in M_l, $\{e\} \cup B_l$ is a dependent set in M_l. This contradicts our assumption that B' is a base in M.

Hence $D_l \cap B' = \emptyset$ and we get

$$B' \subseteq E(M) - (D_l \cup P) = E(M) - D.$$

But, this contradicts our assumption that D is a cocircuit. This proves that D_j is a codependent set in M_j. □

For $j = 1, ..., r$, let

$$\mathcal{C}^*_{j,D_s}(M) = \left\{ D : D \in \mathcal{C}^*_j(M) \text{ and } D \cap S = D_s \right\}, \text{and}$$
$$\tilde{\mathcal{C}}^*_{j,D_s}(M) = \left\{ D : D \in \mathcal{C}^*_j(M) \text{ and } D \cap S \subseteq D_s \right\}$$

Using lemmas 8 and 9, we can now prove the following decomposition theorem that serves as the basis for our first algorithm to estimate the reliability.

Theorem 4. *Let M be a matroid such that $M \setminus S = M_1 \oplus M_2 \oplus ... \oplus M_r$ and $|S| = n_s$. Also, for any $\tilde{D}_j \in \tilde{\mathcal{C}}^*_{j,D_s}(M)$, let $\tilde{P}_j = \tilde{D}_j \setminus D_s$. Then:*

(a) For any $J \subseteq \{1, 2, ..., r\}$ and $J \in \mathcal{J}_{|D_s|+1}$, $(\bigcup\limits_{j \in J} \tilde{P}_j)$ is a codependent set in M.

(b) $\mathcal{C}^*(M) = \bigcup\limits_{J \in \mathcal{J}_{n_s+1}} \mathcal{C}^*_J(M)$.

Proof. First, let us consider a matroid with a separating set S consisting of a single element indexed by e. Consider two cocircuits $D_i \in C^*_{j,\{e\}}(M)$ and $D_j \in C^*_{j,\{e\}}(M)$ such that $i \neq j$. Let $D_i = P_i \cup \{e\}$ such that $D_i \cap E(M_i) = P_i$. Similarly, let $D_j = P_j \cup \{e\}$ such that $D_j \cap E(M_i) = P_j$. We now prove that $P_i \cup P_j$ is a codependent set in M.

From Lemma 9, P_i is a codependent set in M_i. But P_i is also a cocircuit in M_i. If not, then there exist some $P \in P_i$ such that $P_i \setminus P$ is a cocircuit and hence $P_i \setminus P \cup \{e\}$ is a cocircuit of M, which is a contradiction. Similarly, P_i is a codependent set in M_i.

Now assume $P_i \cup P_j, i \neq j$ is a coindependent set in M. Then there is a basis B in M such that

$$B \subseteq E(M) - (P_i \cup P_j).$$

Also there exist bases B^i and B^j such that

$$B^i \subseteq E(M) - (P_i \setminus \{x_i\} \cup e)$$

and

$$B^j \subseteq E(M) - (P_j \setminus \{x_j\} \cup e)$$

This is because $P_i \setminus \{x_i\} \cup e$ and $P_j \setminus \{x_j\} \cup e$ are coindependent sets in M.

Since $B^i \subset E(M) - \{e\}$ and $B^j \subset E(M) - \{e\}$, there exist bases

$$B_i^i \subseteq E(M_i) - P_i \setminus \{x_i\}$$

and

$$B_j^j \subseteq E(M_j) - P_j \setminus \{x_j\}$$

in M_i and M_j respectively.

Now consider the bases B and B^i. Since $P_i \cup P_j \cup e$ is a codependent set, $e \in B$. Otherwise $B \subseteq E(M) - (P_i \cup P_j \cup e)$ which is a contradiction. Also, $e \notin B^i$.

Then by Lemma 4, for any $e \in B \setminus B^i$, there is some $y \in B^i \setminus B$ such that $B' = B \setminus \{e\} \cup \{y\}$ is a base in M. However,

$$B' \subseteq E(M) - (P_i \cup P_j \cup e) \setminus \{y\}$$

cannot be a subset for the following reason. If $y \in E(M) - (E(M_i) \cup E(M_j))$, then

$$B' \subseteq E(M) - (P_i \cup P_j \cup e)$$

which is not a dependent set.

Even if $y \in E(M_i)$ or $y \in E(M_j)$, either

$$B' \subseteq E(M) \setminus (P_i \cup e)$$

or

$$B' \subseteq E(M) \setminus (P_j \cup e).$$

Hence, B' cannot be a base and $P_i \cup P_j$ is a codependent set in M.

Now, let us extend this approach to the matroid M with the separating set S consisting of n_s elements. Consider two cocircuits D_i and D_j such that $D_i \in \mathcal{C}^*_{i,D_s}(M)$ and $D_j \in \mathcal{C}^*_{j,D_s}(M)$ and

$i \neq j$.

Given any element $e \in D_s$, let $P_i = D_i \setminus \{e\}$ and $P_j = D_j \setminus \{e\}$. Then, $P_i \cup P_j$ is a codependent set in M. Now, consider another cocircuit $D_k \in \mathcal{C}^*_{i,D_s}(M)$ such that $k \neq i \neq j$.

For two separate indices e_1 and e_2 of D_s, let

$$P'_i = D_i \setminus \{e_1, e_2\},\ P'_j = D_j \setminus \{e_1, e_2\},\ \text{and } P'_k = D_k \setminus \{e_1, e_2\}.$$

Then, it can easily be proved that $P'_i \cup P'_j \cup P'_k$ is a codependent set in M.

Continuing this approach of deleting each element in D_s, we can prove that,

$$\text{for any } J \subseteq \{1, 2, ..., r\},\ \text{and } J \in \mathcal{J}_{|D_s|+1}, \bigcup_{j \in J} P_j \text{ is a codependent set in } M,$$

where $P_j = D_j \setminus D_s$ for all $j \in J$.

Any $\tilde{D}_j \in \tilde{\mathcal{C}}^*_{j,D_s}(M)$, with $\tilde{P}_j = \tilde{D}_j \setminus D_s$, is such that $\tilde{P}_j \cup D_s$ is a codependent set in M. Therefore, $(\bigcup_{j \in J} \tilde{P}_j)$ is a codependent set in M. Part (b) directly follows from Part(a). □

Theorem 4 provides a framework for finding all the cocircuits of a matroid. Since our design matrix $\boldsymbol{A}$ has a separating set S where $M[\boldsymbol{A}^T \setminus S] = M[\boldsymbol{A}_1^T] \oplus M[\boldsymbol{A}_2^T] \oplus ... \oplus M[\boldsymbol{A}_r^T]$, we can easily apply Theorem 4 to $M[\boldsymbol{A}^T]$. Part (a) of the theorem gives a procedure to find codependent sets and thereby cocircuits involving multiple blocks from cocircuits over single blocks. Part (b) says that any cocircuit of a matroid M with a separating set S cannot involve more that $|S| + 1$ single block matroids. The RVR framework iteratively finds cocircuits over $M[\boldsymbol{A}^T]$ which are then used to obtain the minimal cut sets. If we use MIP over $\boldsymbol{A}$ to solve for the minimal cut sets, the running time of the algorithm is going to be extremely large. However, Theorem 4 can be employed to disintegrate this problem so that the cocircuits over $M[\boldsymbol{A}^T]$ can be obtained by solving MIP over submatrices of $\boldsymbol{A}$.

We integrate MIP$_{\text{CO}}$ within the RVR framework and then utilize the concepts from Theorem 4 to present the RVR$_{\text{CO}}$ algorithm.

RVR$_{\text{CO}}$: *Estimate the System Reliability*

Given a sensor system defined by the matrix $\boldsymbol{A}$ having n sensors $S = (s_1, s_2, ..., s_n)$ with working probabilities p_i, $i = 1, ..., n$. A sample size of N is chosen. Set $W = F = \emptyset$. Initialize $t = 1$

```
I. Execute the following procedure to calculate $\alpha$ and $\beta$

   Procedure $P(W, F, N, t)$:

      1. if $N = 0$
            $\alpha = \beta = 0$; return
      2. else if $r(\boldsymbol{A}_{(W)}) = m$ and $|W| \geq k$
            $\alpha = \beta = N$; return
      3. else if $r(\boldsymbol{A}_{(-F)}) < m$ or $n - |F| < k$
            $\alpha = \beta = 0$; return

      4. get $F_s$ as the set of failed sensors that belong to the border $S$

      5. if $t > n_s - |F_s| + 1$, set $t = n_s - |F_s| + 1$

      6. find a minimal cut set of $S$ as follows:
         6.a remove all the rows in $\boldsymbol{A}$ corresponding to sensors in $\mathcal{F}$
         6.b set $\alpha_i$'s representing those sensors in $\mathcal{W}$ to $0$
         6.c select a random a $J \in \mathcal{J}_t$
         6.d find a minimal rank reduction set by executing MIP$_{\text{CO}}$ over $\boldsymbol{A}[J]$
                  if MIP$_{\text{CO}}$ is infeasible, update $t = t + 1$ and go to Step 7
                  else get the minimal cut set $\mathcal{G} = \{s_{<1>}, s_{<2>}, ..., s_{<h>}\}$ and go to Step 7

      7. divide $\Omega$ into $\Omega_0, \Omega_1, ..., \Omega_h$ based on (2.17) and (2.18). Define $Q_0 = Pr\{\Omega_0\}$
```

```
    8. generate a trial $\big(N(S_{G_t}), ..., N(S_{G_h})\big)$ of the r.v. with multinomial distribution
          $\mathcal{M}\Big(N, p(S_{G_1}), ..., p(S_{G_h})\Big)$, where $p(S_{G_t}) = Pr\{\Omega_t | \Omega_0\}$

    9. for each $t \in \{1, 2, ..., h\}$:
       9.1 set $W_t = W \cup s_{<t>}$ and $F_t = F \bigcup_{l=1}^{t-1} s_{<l>}$
       9.2 call procedure$\Big(P(W_t, F_t, N(S_{G_t}, t)\Big)$ to calculate $\alpha_t$ and $\beta_t$

  II. get $\alpha = (1 - Q_0) \sum_{t=1}^{h} \alpha_t$ and $\beta = (1 - Q_0)^2 \sum_{t=1}^{h} \beta_t$

 III. calculate $\hat{r}_{rvr} = \frac{\alpha}{N}$ and $\hat{V}_{rvr} = \frac{1}{N-1}\left[\beta - \frac{\alpha^2}{N}\right]$
```

The RVR$_{\text{CO}}$ algorithm replaces the inefficient RCEF$_{\text{CO}}$ method with MIP$_{\text{CO}}$ to find the minimal cut sets. However, instead of applying MIP$_{\text{CO}}$ directly over the design matrix $\boldsymbol{A}$, we start by selecting a random block of size 1. We consider matrices involving more blocks only if we fail to obtain a cocircuit using the current set of blocks. Now we utilize the property (a) in Theorem 4 to limit the size of the input matrix to MIP$_{\text{CO}}$ at each iteration. Notice that, we find a cocircuit of the reduced matrix of $\boldsymbol{A}$, i.e., the matrix obtained by removing rows corresponding to failed sensors. However, the cocircuit cannot contain any columns that represent a functional sensor. Hence, we can say that any cocircuit will contain a maximum of $n_s - |F_s| + 1$ blocks. So the upper bound on t is set to this limit thereby capping the size of the input matrix to co. The significance of starting with a single block is evident from the fact that the value $n_s - |F_s| + 1$ at the later iterations is more likely to reduce as more and more border rows will get fixed as failed or working.

6.2 A Circuit Based Approach

Our second algorithm to enhance the RVR procedure is to find the minimal cut sets over $(\boldsymbol{A}^T/W) \setminus F$ by solving a dual problem over $\boldsymbol{H}$ after accounting for fixed sensors. The basic idea of this approach stems from the duality of matroids. In MIP$_{\text{CIR}}$, as discussed in Section 5.1, we find a nonzero $\boldsymbol{y}$ in the null space of $\boldsymbol{H}$ that has the smallest l_0-norm. Then, the smallest circuit $\mathcal{C}$ over $\boldsymbol{H}$ is also the smallest cocircuit over $\boldsymbol{A}$. Hence, we can find the minimal cut sets over $\boldsymbol{A}$ by following a dual approach that mirrors RVR$_{\text{CO}}$.

To introduce this dual RVR method, we need to first deal with the fixed sensors, i.e., sensors in W and F. However, from the duality of matroids, we can easily see that to account for fixed sensors in $\boldsymbol{H}$, we need to apply the dual of the operations performed over $\boldsymbol{A}$. Specifically, for sensors set as failed, the corresponding columns in $\boldsymbol{H}$ needs to be contracted, and for sensors set as working, the corresponding columns in $\boldsymbol{H}$ needs to be deleted. As with MIP$_{\text{CO}}$, the contraction can be replaced by setting the respective β_i's to 1. Then, we can easily dualize the approach in RVR$_{\text{CO}}$. This dual algorithm, denoted as RVR$_{\text{CIR}}$, is similar to RVR$_{\text{CO}}$ except for the fact that we solve MIP$_{\text{CIR}}$ over submatrices of $\boldsymbol{H}$ to obtain the minimal cut sets.

The decomposition of $\boldsymbol{H}$ is facilitated by a dual decomposition theorem that can be derived from Theorem 4. The following notations are used in presenting the theorem.
For $j = 1, ..., r$, let

$$\mathcal{C}_{j,D_s}(M^*) = \{D : D \in \mathcal{C}_j(M^*) \text{ and } D \cap S = D_s\}\text{ , and}$$
$$\tilde{\mathcal{C}}_{j,D_s}(M^*) = \{D : D \in \mathcal{C}_j(M^*) \text{ and } D \cap S \subseteq D_s\}$$

Theorem 5. *Let M be a matroid such that $M \setminus S = M_1 \oplus M_2 \oplus ... \oplus M_r$ and $|S| = n_s$. Also, for any $\tilde{D}_j \in \tilde{\mathcal{C}}_{j,D_s}(M^*)$, let $\tilde{P}_j = \tilde{D}_j / D_s$. Then:*

(a) For any $J \subseteq \{1, 2, ..., r\}$ and $J \in \mathcal{J}_{|D_s|+1}$, $(\bigcup\limits_{j \in J} \tilde{P}_j)$ is a codependent set in M.

(b) $\mathcal{C}(M^*) = \bigcup\limits_{J \in \mathcal{J}_{n_s+1}} \mathcal{C}_J(M^*)$.

The proof of this theorem is omitted, as it is straightforward from Theorem 4. Theorem 5 can be employed to disintegrate the problem of finding the circuits over the dual matroid $M[\boldsymbol{H}]$. Hence, the reliability evaluation problem can be solved using this dual decomposition approach, wherein the circuits over submatrices of $M[\boldsymbol{H}]$ are computed to find the reliability estimate of design matrix

$\boldsymbol{A}$.

The algorithm RVR$_{\text{CIR}}$ is presented below.

RVR$_{\text{CO}}$: *Estimate the System Reliability*

Given a sensor system defined by the matrix $\boldsymbol{A}$ having n sensors $S = (s_1, s_2, ..., s_n)$ with working probabilities p_i, $i = 1, ..., n$. A sample size of N is chosen. Set $W = F = \emptyset$. Initialize $t = 1$

```
I. Execute the following procedure to calculate $\alpha$ and $\beta$

  Procedure $P(W, F, N, t)$:

    1. if $N = 0$
         $\alpha = \beta = 0$; return
    2. else if $r(\boldsymbol{H}_{(W)}) = m$ and $|W| \geq k$
         $\alpha = \beta = N$; return
    3. else if $r(\boldsymbol{H}_{(-F)}) < m$ or $n - |F| < k$
         $\alpha = \beta = 0$; return

    4. get $\mathcal{F}_s$ as the set of failed sensors that belong to the border $S$

    5. if $t > n_s - |F_s| + 1$, set $t = n_s - |F_s| + 1$

    6. find a minimal cut set of $S$ as follows:
       6.a remove all the columns in $\boldsymbol{H}$ corresponding to sensors in $\mathcal{W}$
       6.b set $\beta_i$'s representing those sensors in $\mathcal{W}$ to $0$
       6.c select a random a $J \in \mathcal{J}_t$
       6.d find a minimal rank reduction set by executing MIP$_{\text{CIR}}$ over $\boldsymbol{H}[J]$
             if MIP$_{\text{CIR}}$ is infeasible, update $t = t + 1$ and go to Step 7
             else get the minimal cut set $\mathcal{G} = \{s_{<1>}, s_{<2>}, ..., s_{<h>}\}$ and go to Step 7
```

```
        7. divide $\Omega$ into $\Omega_0, \Omega_1, ..., \Omega_h$ based on (2.17) and (2.18). Define $Q_0 = Pr\{\Omega_0\}$

        6. generate a trial $\big(N(S_{G_t}), ..., N(S_{G_h})\big)$ of the r.v. with multinomial distribution
              $\mathcal{M}\Big(N, p(S_{G_1}), ..., p(S_{G_h})\Big)$, where $p(S_{G_t}) = Pr\{\Omega_t|\Omega_0\}$

        7. for each $t \in \{1, 2, ..., h\}$:
            7.1 set $W_t = W \cup s_{<t>}$ and $F_t = F \bigcup_{l=1}^{t-1} s_{<l>}$
            7.2 call procedure$\Big(P(W_t, F_t, N(S_{G_t}, t)\Big)$ to calculate $\alpha_t$ and $\beta_t$

    II. get $\alpha = (1 - Q_0) \sum_{t=1}^{h} \alpha_t$ and $\beta = (1 - Q_0)^2 \sum_{t=1}^{h} \beta_t$

    III. calculate $\hat{r}_{rvr} = \frac{\alpha}{N}$ and $\hat{V}_{rvr} = \frac{1}{N-1}\left[\beta - \frac{\alpha^2}{N}\right]$
```

7. COMPUTATIONAL RESULTS

We present a detailed analysis of our computational experiments in this chapter. For the degree of redundancy problem, we compare the performance of our algorithms DMIP_{CO} and DMIP_{CIR} against MIP_{CO} [14] and BDMIF [30]. We also report the running time and the degree of redundancy estimates obtained by 2-STAGEL1. All the algorithms are executed for 11 different test categories, each category consisting of five BBD design matrices of the same dimensions. The instances within a category also have the same structure, i.e., the dimensions of the blocks and the borders are the same for each of the instances within a category.

Our algorithms for reliability evaluation, RVR_{CO} and RVR_{CIR}, are compared against the RVR_{RREF} algorithm [3] by implementing and executing these algorithms over 8 of the 11 instances used in the redundancy degree analysis. The rank calculations in the reliability evaluation algorithms are performed using QR factorization, a substantially more efficient method to the alternatives like the SVD decomposition [28].

We implemented all the algorithms in C++ and used ILOG CPLEX 12.9 callable library [35] to solve all the optimization problems. The algorithms were executed on a PC which has an Intel Core$^{\text{TM}}$ i7-4910MQ 2.90GHz processor and 32 GB of RAM. We set a limit of 10 hours on the running time of each instance. The instances in categories 1 and 2 are based on the multi-stage assembly examples reported in [22], and the characteristics of the remaining ones are inspired by the instances in [14]. All the instances used in this paper are available at http://ise.tamu.edu/people/faculty/kianfar/modal/index_files/VKDPinstances.zip.

7.1 Computational Results for the Degree of Redundancy Problem

In Table 7.1, we present our computational experiments for the heuristic algorithm 2-STAGEL1. Each row of Table 7.1 represents an instance category. The table shows the average running time of 2-STAGEL1 over all the instance categories along with the Stage 1 and Stage 2 estimates for the degree of redundancy. The table also reports the running time of $\text{MIP}_{\text{CO-W}}$, the warm start version of MIP$_{\text{CO}}$. The value of $d(\hat{\boldsymbol{A}})$ from Stage 1 of 2-STAGEL1 is not optimal for most of the test instances. However, the Stage 2 improves upon this estimate, and provides the optimal solution for all the instance categories except for those in 650×340. The algorithm MIP$_{\text{CO-W}}$ improves upon MIP$_{\text{CO}}$ by about 10% on average.

In Table 7.2, we summarize the results for the redundancy degree problem. As before, each row of this table represents an instance category. For each test category, the table reports the dimension ($n \times m$), the number of border rows (n_s), and the number of blocks (r) for the matrices in that category along with the average running times (average over all the five instances in each category) for MIP$_{\text{CO}}$, BDMIF, DMIP$_{\text{CO}}$, and DMIP$_{\text{CIR}}$. The instances with the best run times are reported in bold. Table 1 also shows the average value of the degree of redundancy, $d(\boldsymbol{A})$, for each of the instance categories.

To examine the effectiveness of our decomposition theorems with that of the bound and decompose theorem, we include the maximum number of blocks, t, in the decomposed submatrices using our theorems along with the decomposition bound b_t that will allow computations over t-block submatrices using the bound and decompose Theorem. The minimum decomposition bound allowable for this theorem, $b_{min} = n_s\, r/(r-1) - 1$, is also listed in this table.

Except for instance categories 1 and 3, which are anyway much smaller, the best running times are reported for either DMIP$_{\text{CIR}}$ or DMIP$_{\text{CO}}$. Fully about 60% of the tested instances saw a reduction in running time of more than 75% with both DMIP$_{\text{CIR}}$ or DMIP$_{\text{CO}}$ compared to the best performing algorithm among MIP$_{\text{CO}}$ and BDMIF. For the two largest instance categories, 1009×252 and 2018×504, the average reduction in the running time was more than 80%.

Matrices in categories 450×250 , 650×340, and 501×384 have large 1-blocks with significant borders. These instances are also on average 10% denser than the remaining ones. While the 501×384 instances are not solved by any of the algorithms, the other two category instances are only solved by DMIP$_{\text{CIR}}$ and DMIP$_{\text{CO}}$, that too with an average running time of around 2 minutes. The running times reported for DMIP$_{\text{CIR}}$ are comparable to that for DMIP$_{\text{CO}}$. However, the slight variances are largely due to the differences in the size of inputs and the MIP formulations for the

Table 7.1: Running Times and Estimates from 2-StageL1

No	$n \times m$	Avg. $d(\hat{\boldsymbol{A}})$		2-StageL1	MIP$_{\text{CO-W}}$
		Stage 1	Stage 2		
1	26×12	4.8	4	1.85 sec.	1.89 sec.
2	66×27	7.4	7	5.6 sec.	15.6 sec.
3	154×72	6	5	7.42 sec.	9.12 sec.
4	316×144	3.8	3.8	13.91 sec.	23.8 sec.
5	485×360	1.6	1.6	15.29 sec.	22.63 sec.
6	222×55	14	14	12.38 sec.	31.21 sec.
7	1009×252	17	17	1.99 min.	20.46 min.
8	2018×504	18	18	6.32 min.	$>$ 10 hrs.
9	450×250	20.2	19	2.28 min.	$>$ 10 hrs.
10	650×340	36	35	3.7 min.	$>$ 10 hrs.
11	501×384	35	34	2.89 min.	$>$ 10 hrs.

two.

The categories 450×250 and 650×340, in particular, highlight the effectiveness of our decomposition theorems. For example, for all the instances in these categories, we can decompose the problems into 1 blocks ($t = 1$) using our theorems, while the bound and decompose theorem allows such a decomposition only when the value of the d is more than 7 for 450×250 instances, and 19 for 650×340 instances.

The minimum decomposition bounds, b_{min}, are also too large for these instances for BDMIF to execute MIF over submatrices of reasonable size. In fact, except for instances in 316×144 and 485×360, all the others are reduced to 1 block subproblems when employing our decomposition theorems.

Not only was our algorithms successful in reducing the size of the subproblems, but the total number of subproblems to be solved are also considerably smaller than BDMIF. For example, for instances in 2018×504, DMIP$_{\text{CO}}$ solved 84 subproblems of size 24×6, executing MIP$_{\text{CO}}$ over these subproblems, to get the value of t, and then solved around 45 subproblems (average over all

Table 7.2: A Performance Comparison of Redundancy Degree Algorithms

No.	$n \times p$	n_s	r	b_{min}	t	b_t	MIP$_{\text{CO}}$	BDMIF	DMIP$_{\text{CO}}$	DMIP$_{\text{CIR}}$	Avg. $d(\boldsymbol{A})$
1	26×12	2	4	1.67	1	3	**0.13 sec.**	3.81 sec.	0.26 sec.	0.34 sec.	4
2	66×27	3	9	2.37	1	5	17.18 sec.	43.07 sec.	**2.01 sec.**	2.2 sec.	7
3	154×72	2	2	3	1	3	**2.34 sec.**	20.33 sec.	3.41 sec.	3.18 sec.	5
4	316×144	8	4	9.67	2	11	32.35 sec.	3.13 min.	13.51 sec.	**13.28 sec.**	3.8
5	485×360	5	8	4.71	3	6.5	38.09 sec.	5.78 min.	20.13 sec.	**17.46 sec.**	1.6
6	222×55	2	11	1.2	1	3	43.15 sec.	1.64 min.	**9.12 sec.**	9.83 sec.	14
7	1009×252	1	42	0.024	1	1	> 10 hrs.	19.9 min.	3.12 min.	**3.23 min.**	17
8	2018×504	2	84	1.024	1	3	> 10 hrs.	1.47 hrs.	11.25 min.	**11.04 min.**	18
9	450×250	4	2	7	1	7	> 10 hrs.	> 10 hrs.	1.36 min.	**1.26 min.**	19
10	650×340	10	4	12.33	1	19	> 10 hrs.	> 10 hrs.	2.32 min.	**1.89 min.**	34
11	501×384	9	4	11	1	17	> 10 hrs.	> 10 hrs.	> 10 hrs.	> 10 hrs.	$\geq 1, \leq 34$

five instances) of size 26×6 to find $d(\boldsymbol{A})$. However, BDMIF solved an MIF with $d = 1$ over the entire matrix, then solved $\binom{84}{2}$ subproblems each of size 50×12, and another 1348 subproblems of size 26×6 until a reduction in rank was observed at $d = 19$.

7.2 Computational Results for the Reliability Evaluation Problem

In Table 7.3 and Table 7.4, we present the computational analysis for the reliability evaluation algorithms RVR$_{\text{RREF}}$, RVR$_{\text{CO}}$ and RVR$_{\text{CIR}}$. As with Table 7.2, each row in these tables corresponds to an instance category. For each algorithm and each category, we list the average running time (t) and the average variance of estimated system reliability ($\hat{V}$) over all the instances in that category. While Table 7.3 reports the values based on a sample size $N_1 = 100,000$, Table 7.4 reports for $N_2 = 1,000,000$. The parameter Γ for RVR$_{\text{CO}}$ and RVR$_{\text{CIR}}$ calculated as

$$\Gamma_{\text{RVR}_{\text{CO}}(\text{RVR}_{\text{CIR}})} = \frac{t_{\text{RVR}_{\text{CO}}(\text{RVR}_{\text{CIR}})} \times \hat{V}_{\text{RVR}_{\text{CO}}(\text{RVR}_{\text{CIR}})}}{t_{\text{RVR}_{\text{RREF}}} \times \hat{V}_{\text{RVR}_{\text{RREF}}}}$$

is used as a criterion to compare the performance of our algorithms against RVR$_{\text{RREF}}$.

Another performance indicator listed in Table 7.3 and Table 7.4 is the value N_{req}, the approx-

Table 7.3: A Performance Comparison of Reliability Analysis Algorithms for N = 100,000

No	$n \times m$	RVR_{RREF}			RVR_{CO}			RVR_{CIR}		
		t	$\hat{V}$	N_{req}	t	$\hat{V}$	Γ	t	$\hat{V}$	Γ
1	26×12	43.15 sec.	3.51E-03	133000	39.71 sec.	6.74E-05	1.77E-02	40.17 sec.	6.65E-05	1.76E-02
2	66×27	2.15 min.	2.42E-03	139000	1.42 min.	4.39E-05	1.20E-02	1.45 min.	4.37E-05	1.22E-02
3	154×72	2.31 min.	1.13E-03	163000	1.94 min.	1.89E-05	1.41E-02	1.91 min.	1.87E-05	1.37E-02
4	316×144	5.32 min.	3.02E-02	124000	3.77 min.	7.38E-04	1.73E-02	3.81 min.	7.40E-04	1.75E-02
5	485×360	6.25 min.	2.94E-02	127000	3.92 min.	8.25E-04	1.76E-02	3.98 min.	8.29E-04	1.80E-02
6	222×55	5.13 min.	2.01E-03	182000	3.4 min.	2.55E-05	8.41E-03	3.62 min.	2.58E-05	9.06E-03
7	1009×252	28.37 min.	6.34E-03	218000	17.22 min.	7.06E-05	6.76E-03	17.15 min.	7.04E-05	6.71E-03
8	2018×504	56.22 min.	5.29E-03	223000	33.87 min.	6.23E-05	7.10E-03	32.54 min.	6.31E-05	6.90E-03

imate number of samples required by RVR_{RREF} to obtain a reliability estimate with a variance comparable to the average of the variances reported for RVR_{CO} and RVR_{CIR}.

Fully, on average, both RVR_{CO} and RVR_{CIR} report a decrease in the running time (compared to RVR_{RREF}) of about 30% for a sample size of N_1 and more than 36% for a sample size of N_2. The running time reduction is more than 36% with 100,000 samples and more than 46% with 1,0000,000 samples for all the instances in 1009×252 and 2018×504 for both RVR_{CO} and RVR_{CIR}.

The sample variance of the reliability estimate also presents a significant reduction, with an average reduction of more than 60 times for N_1 and more than 120 times for N_2. For instances in 1009×252 and 2018×504, the variance reduction is over 200 times for both RVR_{CO} and RVR_{CIR} when N_2 samples are used. The variance estimates and the running times for RVR_{CO} are comparable to that of RVR_{CIR}, although RVR_{CIR} executes slightly faster than RVR_{CO} for large instances.

A value of $\Gamma < 1$ signifies a superior performance of RVR_{CO} or RVR_{CIR} against RVR_{RREF}; the smaller the value, the better the performance. We observe that the Γ values across all the instances are significantly smaller (of the order of 10^{-2} or 10^{-3}) for both sample sizes and both algorithms. Even for the instances in 316×144 and 485×360, which report the highest values of Γ, these values are smaller than 0.001 for a sample size of 1,000,000. These are also the least reliable of all the tested instances.

Table 7.4: A Performance Comparison of Reliability Analysis Algorithms for N = 1,000,000

No	$n \times m$	RVR$_{\text{RREF}}$			RVR$_{\text{CO}}$			RVR$_{\text{CIR}}$		
		t	$\hat{V}$	N_{req}	t	$\hat{V}$	Γ	t	$\hat{V}$	Γ
1	26×12	1.31 min.	1.62E-05	1210000	1.14 min.	1.94E-07	1.04E-02	1.15 min.	1.95E-07	1.06E-02
2	66×27	4.76 min.	8.54E-06	1313000	3.23 min.	9.68E-08	7.69E-03	3.35 min.	9.91E-08	8.17E-03
3	154×72	6.27 min.	3.41E-05	1383000	4.07 min.	3.51E-07	6.68E-03	4.01 min.	3.45E-07	6.47E-03
4	316×144	10.48 min.	2.98E-04	1108000	6.55 min.	4.34E-06	9.10E-03	6.31 min.	4.28E-06	8.65E-03
5	485×360	11.97 min.	3.11E-04	1104000	7.52 min.	4.69E-06	9.47E-03	7.45 min.	4.72E-06	9.45E-03
6	222×55	10.16 min.	5.49E-05	1431000	5.33 min.	4.23E-07	4.04E-03	5.36 min.	4.20E-07	4.03E-03
7	1009×252	56.92 min.	2.18E-05	1464000	28.05 min.	1.02E-07	2.31E-03	27.13 min.	1.03E-07	2.25E-03
8	2018×504	2.06 hrs.	2.53E-05	1488000	1.05 hrs.	1.14E-07	2.30E-03	1.02 hrs.	1.09E-07	2.13E-03

As expected, the reduction in the variance is more pronounced for instances which are highly reliable, namely 154×72, 222×55, 1009×252, and 2018×504. On average, RVR$_{\text{RREF}}$ required more than 160,000 samples and 1,300,000 samples to get a comparable reliability estimate as that of RVR$_{\text{CO}}$ and RVR$_{\text{CIR}}$ with 100,000 samples and 1,000,000 samples respectively as indicated by the values of N_{req}.

8. CONCLUSION AND FUTURE RESEARCH

8.1 Conclusion

In this dissertation, we developed algorithms to compute the degree of redundancy and reliability of large-scale linear sensor systems. A polynomial-time algorithm to find the degree of redundancy and reliability is highly unlikely due to the complexity of these problems. The sensitive task of design, analysis, and operation of linear sensor systems calls for algorithms that can determine the redundancy degree and reliability of very large linear models. However, such algorithms are virtually nonexistent in the current literature. Our research efforts into finding the degree of redundancy and estimating the reliability resulted in two main approaches: One a heuristic method, and the second a matroid decomposition procedure.

In our heuristic approach, we developed an algorithm, 2-STAGEL1, that solves a series of linear programs to find good solutions to the degree of redundancy problem. We then used the solution from 2-STAGEL1 to find a good starting solution for the existing MIP formulation [14] for the degree of redundancy problem. Finally, we solved this MIP using CPLEX 12.9 solver after adding these starting solutions to warm start the MIP. We also establish the connection between the redundancy degree problem and the well-defined circuit and cocircuit problems in matroid theory.

Most linear sensor systems in practice exhibit structural properties. We took leverage of the structure of the linear system's design matrix and introduced decomposition theorems characterizing the properties of the cocircuits and circuits of structured matroids. We then developed algorithms based on these decomposition theorems to compute the degree of redundancy and estimate the reliability of large-scale structured linear sensor systems commonly found in practice. Specifically, we proposed two cocircuit based matroid decomposition algorithms: DMIP_{CIR} which finds the degree of redundancy and RVR_{CIR} which evaluates the reliability. Similarly, we proposed two circuit based matroid decomposition algorithms: DMIP_{CO} for the redundancy degree problem and RVR_{CO} for the reliability evaluation problem. Our decomposition theorems are based on the matroid theory properties of connectivity and duality.

Following a divide-and-conquer approach, we decomposed the given problem into smaller subproblems and then solved the resulting subproblems using MIP formulations. By performing computations over these subproblems, our algorithms were able to get the solutions much faster than the previously existing algorithms in the literature. The variance of the reliability estimates from RVR_{CO} and RVR_{CIR} was much smaller than the ones obtained from the existing RVR_{RREF} algorithm [3].

8.2 Future Research

Several new research topics arise from the methodological developments in this dissertation. Some of the directions from the results in this dissertation are as follows:

1. All the theorems in this dissertation are proved for general matroids, thereby making the presented algorithmic frameworks suitable for any general matroids. Hence, these approaches can be extended to graphic matroids to solve for the circuits and cocircuits over graphs. This adaptability of our decomposition algorithms can provide significant computational gains when applied to network reliability problems.

2. With the degree of redundancy and reliability as modeling parameters, we can develop a full-fledged robust regression model to analyze fixture faults in multi-stage manufacturing processes. We plan to use the values of degree of redundancy and reliability from our algorithms to improve the efficiency of statistical regression estimator models [2, 22] for fault diagnosis and other applications of linear sensor systems.

3. Our algorithms solve MIPs to find the circuits and cocircuits over the design matrix that defines the linear sensor system to solve for degree of redundancy and reliability. It is interesting to investigate the possibility of strengthening the MIP formulations via polyhedral study and generation of new strong valid inequalities. If successful, we can integrate our decomposition techniques with these stronger MIP formulations thereby enhancing the existing approaches in solving these problems.

REFERENCES

[1] Q. Yang and Y. Chen, “Reliability of coordinate sensor systems under the risk of sensor precision degradations,” *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 2, pp. 291–302, 2010.

[2] J. J. Cho, Y. Chen, and Y. Ding, “Calculating the breakdown point of sparse linear models,” *Technometrics*, vol. 51, no. 1, pp. 34–46, 2009.

[3] Q. Yang and Y. Chen, “Monte Carlo methods for reliability evaluation of linear sensor systems,” *IEEE Transactions on Reliability*, vol. 60, no. 1, pp. 305–314, 2011.

[4] D. Apley and J. Shi, “Diagnosis of multiple fixture faults in panel assembly,” *Journal of Manufacturing Science and Engineering*, vol. 120, no. 4, pp. 793–801, 1998.

[5] S. Zhou, Y. Ding, Y. Chen, and J. Shi, “Diagnosability study of multistage manufacturing processes based on linear mixed-effects models,” *Technometrics*, vol. 45, no. 4, pp. 312–325, 2003.

[6] S. Zhou, Q. Huang, and J. Shi, “State space modeling of dimensional variation propagation in multistage machining process using differential motion vectors,” *IEEE Transactions on robotics and automation*, vol. 19, no. 2, pp. 296–309, 2003.

[7] D. H. Johnson and D. E. Dudgeon, *Array signal processing: concepts and techniques*. Simon & Schuster, 1992.

[8] 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, no. 1, pp. 81–95, 2010.

[9] D. Ceglarek, J. Shi, and S. Wu, “A knowledge-based diagnostic approach for the launch of the auto-body assembly process,” 1994.

[10] Y. Chen and Q. Yang, "Sensor system reliability modeling and analysis for fault diagnosis in multistage manufacturing processes," *IIE Transactions*, vol. 41, no. 9, pp. 819–830, 2009.

[11] Y. Ding, D. Ceglarek, and J. Shi, "Fault diagnosis of multistage manufacturing processes by using state space approach," *Journal of Manufacturing Science and Engineering*, vol. 124, no. 2, pp. 313–322, 2002.

[12] J. A. Camelio and S. J. Hu, "Multiple fault diagnosis for sheet metal fixtures using designated component analysis," *Journal of Manufacturing Science and Engineering*, vol. 126, no. 1, pp. 91–97, 2004.

[13] S. Zhou, Y. Chen, and J. Shi, "Statistical estimation and testing for variation root-cause identification of multistage manufacturing processes," *IEEE Transactions on Automation Science and Engineering*, vol. 1, no. 1, pp. 73–83, 2004.

[14] 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, pp. 447–450, Apr. 2011.

[15] R. R. Wilcox, *Introduction to robust estimation and hypothesis testing*. Academic press, 2011.

[16] C. W. Coakley and L. Mili, "Exact fit points under simple regression with replication," *Statistics & probability letters*, vol. 17, no. 4, pp. 265–271, 1993.

[17] L. Mili, C. W. Coakley, *et al.*, "Robust estimation in structured linear regression," *The Annals of Statistics*, vol. 24, no. 6, pp. 2593–2607, 1996.

[18] P. L. Davies, "Aspects of robust linear regression," *The Annals of statistics*, pp. 1843–1899, 1993.

[19] P. J. Rousseeuw, "Least median of squares regression," *Journal of the American Statistical Association*, vol. 79, pp. 871–880, 1984.

[20] D. Simpson, "Introduction to rousseeuw (1984) least median of squares regression," in *Breakthroughs in Statistics*, pp. 433–461, Springer, 1997.

[21] A. Vardy, "The intractability of computing the minimum distance of a code," *IEEE Transactions on Information Theory*, vol. 43, no. 6, pp. 1757–1766, 1997.

[22] J. J. Cho, Y. Chen, and Y. Ding, "On the (co) girth of connected matroids," *Discrete Applied Mathematics*, vol. 155, no. 18, pp. 2456–2470, 2007.

[23] W. Kuo and M. J. Zuo, *Optimal reliability modeling: principles and applications*. John Wiley & Sons, 2003.

[24] E. Manzi, M. Labbé, G. Latouche, and F. Maffioli, "Fishman's sampling plan for computing network reliability," *IEEE Transactions on Reliability*, vol. 50, no. 1, pp. 41–46, 2001.

[25] H. Cancela and M. El Khadiri, "The recursive variance-reduction simulation algorithm for network reliability evaluation," *IEEE Transactions on Reliability*, vol. 52, no. 2, pp. 207–212, 2003.

[26] J. G. Oxley, *Matroid theory*. Oxford University Press, USA, 2006.

[27] K. Murota, *Matrices and matroids for systems analysis*, vol. 20. Springer Science & Business Media, 2009.

[28] G. H. Golub and C. F. Van Loan, *Matrix computations*, vol. 3. JHU Press, 1996.

[29] J. Lee and J. Ryan, "Matroid applications and algorithms," *ORSA Journal on Computing*, vol. 4, no. 1, pp. 70–98, 1992.

[30] M. Bansal, K. Kianfar, Y. Ding, and 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, no. 4, pp. 1151–1157, 2013.

[31] G. L. Nemhauser and L. A. Wolsey, "Integer programming and combinatorial optimization," *Wiley, Chichester. GL Nemhauser, MWP Savelsbergh, GS Sigismondi (1992). Constraint*

Classification for Mixed Integer Programming Formulations. COAL Bulletin, vol. 20, pp. 8–12, 1988.

[32] S. Foucart and H. Rauhut, “A mathematical introduction to compressive sensing,” *Bull. Am. Math*, vol. 54, pp. 151–165, 2017.

[33] D. L. Donoho and M. Elad, “Optimally sparse representation in general (nonorthogonal) dictionaries via l_1 minimization,” *Proceedings of the National Academy of Sciences*, vol. 100, no. 5, pp. 2197–2202, 2003.

[34] 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: A Journal Issued by the Courant Institute of Mathematical Sciences*, vol. 59, no. 6, pp. 797–829, 2006.

[35] CPLEX, “IBM-ILOG.” `https://www.ibm.com/support/knowledgecenter/SSSA5P_12.9.0/ilog.odms.studio.help/Optimization_Studio/topics/COS_home.html`, 2019.