2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)
Munich, Germany, August 20-24, 2018

A Mixed Integer Programming Based Recursive Variance Reduction Method for Reliability Evaluation of Linear Sensor Systems

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

Abstract— **Linear models have been successfully used to establish the connections between sensor measurements and source variables in sensor networks. Sensor failures are a leading concern during the estimation of these source variables that cannot be measured directly. The reliability of a sensor system is a probabilistic evaluation of the ability of a system to withstand sensor failures. Finding the exact reliability of a linear sensor system is proven to be a #P problem. Consequently, for most practical systems, it is highly unlikely to obtain exact solutions to this problem within a reasonable timeframe. A viable alternative is to estimate the reliability using the crude Monte Carlo method. However, this method is known to be inefficient for highly reliable systems. An improved Monte Carlo approach called the Recursive Variance Reduction (RVR) method is commonly used in the literature to obtain better reliable estimates. However, the accuracy of this method banks heavily on the approach used in finding minimal cut sets of the linear sensor system. In this paper, we introduce two enhanced RVR methods in which mixed integer programming algorithms are deployed to find minimal cut sets that significantly improve the accuracy of the overall RVR technique. A case study over a wide range of test instances is conducted to establish the efficiency of the proposed methods.**

I. Introduction

In many applications in science and technology, one encounters the problem of predicting various physical phenomena based on the measured information. In a linear sensor system, the source variables $\boldsymbol{x}$ representing these physical phenomena and the sensor measurements $\boldsymbol{u}$ that assist in estimating these variables are linked by a set of linear equations,

$$\boldsymbol{u} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e}, \tag{1}$$

where $\boldsymbol{u}, \boldsymbol{e} \in \mathbb{R}^n$, and $\boldsymbol{x} \in \mathbb{R}^m$. $\boldsymbol{A}$ is an $n \times m$ design matrix $(n \geq m)$ that models the linear measurement process. The matrix $\boldsymbol{A}$ is assumed to be of full column rank i.e., $r(\boldsymbol{A}) = m$, where $r(.)$ denotes the rank function. The last term $\boldsymbol{e}$ is a random noise vector, which is assumed to be normally distributed with mean 0 and covariance matrix $\sigma^2\boldsymbol{I}$. The vector of unknown parameters $\boldsymbol{x}$ is commonly estimated using the Least Square estimator, $\hat{\boldsymbol{x}}_{LS} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{u}$.

Linear sensor systems are widely deployed in complex multistage manufacturing systems to measure dimensional deviations during the assembly process, enabling automatic diagnosis of manufacturing process faults [1]. Figure 1 shows a typical multistation assembly process. At Station 1, two parts are assembled. The resulting sub-assembly is transferred to Station 2 where it is further assembled with two more parts. The final assembly is inspected for quality at Station 3. Coordinate sensors are placed at all the stations to measure the dimensional deviation of parts in either the x or the z direction. The response $\boldsymbol{u}$ includes the measurements obtained by these sensors. The design matrix $\boldsymbol{A}$ is determined by process design information, product geometry and tooling layout. Each part or finished sub-assembly is positioned by a pair of locators, consisting of a four-way locator and a two-way locator. The four-way locator controls the motion of a part in both the x and z directions, while the two-way locator controls the motion in only the z direction. The source variables $\boldsymbol{x}$ are potential process faults due to dimensional locating errors caused by locators.

An LS estimator is highly sensitive to system noise and sensor failures. Though some sensor failures result only in measurement precision degradation, others may be catastrophic providing no useful information. In any case, it is highly important to estimate the reliability of a sensor system against sensor failures. The reliability of a linear sensor system is a probabilistic evaluation of the ability of the system to tolerate sensor failures given the failure probabilities of individual sensors. In the literature, the reliability problem is studied for both modes of sensor failures. In [2] and [3] reliability under the risk of sensor precision degradation is investigated. In contrast, reliability when considering catastrophic sensor failures is explored in [4] and [5]. The research efforts in this paper deal with catastrophic sensor failures. Each sensor is assumed to have two states, either working or failed.

Let us define the reliability of a sensor system under the case of catastrophic sensor failures. *Given the working probability of each sensor in a linear sensor system, and assuming the failure of sensors are s-independent, the* ***reliability of a linear sensor system*** *is defined as the probability that $r(\boldsymbol{A}_{(-\delta)}) = m$, and the number of sensors in $\boldsymbol{A}_{(-\delta)}$ is greater than or equal to k. $\boldsymbol{A}_{(-\delta)}$ is the reduced matrix of $\boldsymbol{A}$ after deleting the rows representing the failed sensors indexed by δ, and k is the minimum required number of*

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

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

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

978-1-5386-3593-3/18/$31.00 ©2018 IEEE

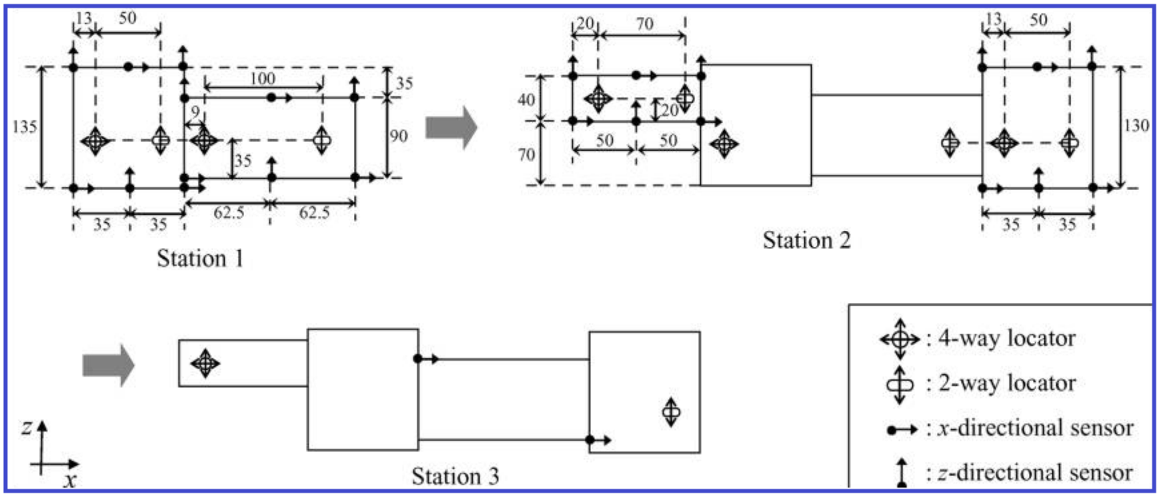

Fig. 1: A Three Station Assembly Process with Coordinate Sensors [1]

functioning sensors [5]. When either of these cases occurs, we say the sensor system failed. By system failure, we mean that the sensor system can no longer estimate the source variables using classic LS estimators. We often fix k before performing any estimation process to ensure the desired level of statistical efficiency.

In [4], Yang and Chen proved that the exact reliability evaluation under catastrophic sensor failure is a #P complete problem. They also developed exact algorithms for finding reliability for two special cases of linear sensor systems. Most practical purposes require the deployment of large-scale sensor systems; and for such systems, exact evaluation of reliability is a highly elusive undertaking. For example, the coordinate sensor system used in the multistation assembly process usually consists of hundreds of sensors. For these complex systems, a straightforward alternative is to estimate the reliability by applying crude Monte Carlo method. For systems that are highly reliable, the estimates from the crude Monte Carlo method are often inefficient. An improved Monte Carlo method called the Recursive Variance Reduction (RVR) method is applied in [5] to give more acceptable estimates of the reliability. The main idea of the RVR method is to reduce the sample space by obtaining as many combinations of sensor states that ensure system failure or success before applying the crude Monte Carlo method.

The reduction in sample space is achieved in [5] by finding the minimal cut sets (defined in Section II) of the linear sensor system. The accuracy of the estimated reliability by the RVR method depends on the efficacy of the approach employed to find these minimal cut sets. Ideally, minimal cut sets that can contribute to the highest reduction in the sample space is preferred. We, based on our analysis, have observed that the current approach used in [5] to finding the minimal cut sets can be improved by utilizing the mathematical tool called the matroid theory. *In matroid theory, the minimal subset of rows of* $\boldsymbol{A}$ *whose removal causes the reduced matrix to be rank-deficient is called a cocircuit of the vector matroid defined over the rows of* $\boldsymbol{A}$. These cocircuits can indeed provide the minimal cut sets. Over the years, various approaches have been studied to find the smallest cocircuit, the value of which has vital importance in robust regression analysis [1], [6], [7]. In [6], the smallest cocircuits are obtained using mixed integer programming.

In this paper, we propose two algorithms to estimate the reliability of a linear sensor system, namely RVR_{MIP} and RVR_{DMIP}. In RVR_{MIP}, we modify the mixed integer programming approach in [6] and integrate this within the RVR method to find minimal cut sets that are much more efficient in reducing the sample space. In the second algorithm RVR_{DMIP}, we take advantage of the the theory of matroids to formulate an equivalent mixed integer program over the dual of the vector matroid $\boldsymbol{A}$ to obtain minimal cut sets and incorporate this within the RVR approach. We implement and test these algorithms over a wide range of test instances inspired by the multistation assembly process application. Our test results discussed in Section IV illustrate the superior performance of our algorithms when compared to the previous RVR algorithm. Using our algorithms, we were able to substantially reduce the variance of the estimated system reliability for all the test instances.

The rest of the paper is organized as follows: Section II gives a brief review of the algorithms previously proposed in the literature. In Section III, we introduce our approach to finding the minimal cut sets and then present our algorithms for reliability evaluation. We report our computational studies in Section IV and conclude with some remarks in Section V.

II. Related Work

Consider the linear sensor system $\mathcal{S} = (s_1, s_2, ..., s_n)$ having n sensors defined over $\boldsymbol{A}$. Let v_i be the random variable that defines the state of the i^{th} sensor. $v_i = 1$ (sensor s_i works) with probability p_i and $v_i = 0$ (sensor s_i fails) with probability $1 - p_i$. Define $\mathcal{V} = (v_1, v_2, ..., v_n)$ as the component state vector. Assume ϕ to be the random variable that defines the state of the system $\mathcal{S}$, where $\phi = 1$ if the system works, else 0. Then the reliability of $\mathcal{S}$, $r_s = E(\phi)$, where $E(.)$ denotes the expectation of a random variable. To apply the crude Monte Carlo method, we generate N s-independent samples of the component state vector $\mathcal{V}$. In each sample, the component v_i of $\mathcal{V}$ takes a value 0 or 1 based on the corresponding failure or working probability of sensor i. Let ϕ_j represent the system state for the sample j. In any sample j, if the number of working sensors is less than k, or $r(\boldsymbol{A}_{(-\delta)}) < m$, ϕ_j is set to 0. Otherwise, it is set to 1. Then, the system reliability can be estimated by an unbiased estimator $\hat{r}_c$ with a sample variance $\hat{V}_c$, where

$$\hat{r}_c = \sum_{j=1}^{N} \phi_j / N, \quad \hat{V}_c = \frac{1}{N-1} \sum_{j=1}^{N} (\phi_j - \hat{r}_c)^2. \qquad (2)$$

For a system which is highly reliable, the component state vectors that correspond to system failure form just a minuscule fraction of the overall sample space of component state vectors. Hence, for large-scale systems, we need to use an inordinately large sample size to attain any reasonable estimate of reliability. In [5], the recursive variance reduction method - an improved Monte Carlo method used commonly in network reliability literature, is applied to overcome this limitation. As discussed in the previous section, the underlying idea of the RVR method is to obtain as many sets of component state vectors that guarantee a system failure or success. Once these component states are obtained, the sample space can be reduced, and then the crude Monte Carlo method be applied. Consider the sample space Ω consisting of all 2^n possible component state vectors. Let the sets Ω_W (with probability p_W) and Ω_F (with probability p_F) represent the sets of component state vectors that ensure system functioning and system failure respectively. The sampling space can now be reduced to $\Omega_U = \Omega - (\Omega_W \cup \Omega_F)$. Then, we can calculate the system reliability as

$$\begin{aligned} r_s &= p_W \; E[\phi|\Omega_W] + p_F \; E[\phi|\Omega_F] + p_U \; E[\phi|\Omega_U] && (3) \\ &= p_W + p_U \; E[\phi|\Omega_U] && (4) \end{aligned}$$

The variance of the reliability estimate is decreased based on

$$\begin{aligned} Var(p_W + p_U \; E[\phi|\Omega_U]) &= (r - p_W)(1 - r - p_F) && (5) \\ &\leq r(1-r) = Var(\phi). && (6) \end{aligned}$$

Accordingly, from (6) we can conclude that the RVR method permits a more accurate estimate of the reliability. The larger the values of p_W and p_F, the greater the reduction in variance and higher the efficiency of the RVR method. This is not surprising since larger values of p_W and p_F results in greater reduction in sample space.

The reduction in the sample space is achieved in [5] by finding the minimal cut sets of the linear sensor system. *Given the states of some sensors in a linear sensor system fixed as either functional 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.* Consider a minimal cut set $\mathcal{G} = \{s_{i_1}, s_{i_2}, ..., s_{i_h}\}$, where s_{i_j} is the j^{th} sensor in $\mathcal{G}$. Then, by definition of minimal cut sets, any component state vector $\mathcal{V}$ with v_{i_j} set to 0 for $j = 1, ..., h$ results in system failure. The failure probability of $\mathcal{G}$ can be calculated as $\Pi_{j=1}^{h}(1 - p_{i_j})$. Undoubtedly, we wish to find a minimal cut set with the largest failure probability.

We now review the approach by which the minimal cut sets are obtained in [5]. Consider the case where none of the sensors are fixed. First, the cocircuits of the matroid over the rows of $\boldsymbol{A}$ are obtained using the following theorem (proved in [5]). *The set of sensors which correspond to the nonzero elements in a nonzero column of the reduced column echelon form of* $\boldsymbol{A}$ *is a cocircuit of* $\boldsymbol{A}$. In future, we refer to this technique of finding the cocircuits as RCEF_{CO}. As defined earlier; the cocircuits are the minimal set of rows that when removed from $\boldsymbol{A}$ reduces its rank. Therefore, a component state vector with 0 entries for those sensors indexed by a cocircuit will result in system failure. Let $\mathcal{C}$ be a cocircuit. If $|\mathcal{C}| \leq n - k + 1$, then $\mathcal{C}$ is also a minimal cut set. This is because the simultaneous failure of any proper subset of sensors in $\mathcal{C}$ can neither produce a rank deficient matrix $\boldsymbol{A}_{(-\delta)}$ nor can reduce the number of sensors that are not fixed to less than k. Consequently, any proper subset of $\mathcal{C}$ does not guarantee a system failure; this makes $\mathcal{C}$ a minimal cut set. However, if $|\mathcal{C}| > n - k + 1$, then each subset of $\mathcal{C}$ with a cardinality of $n - k + 1$ is a minimal cut set.

Next, consider a system where some sensors are fixed as working and some others as failed. In this case, we first transform $\boldsymbol{A}$ to $\boldsymbol{A}'$ as follows. For each sensor fixed as failed, delete the corresponding rows from $\boldsymbol{A}$. For each sensor known to be working, the corresponding row (denoted by e) in $\boldsymbol{A}$ is transformed to a unit vector by performing elementary column operations. Then delete the column and row containing the unique nonzero entry in e. We call this the pivot operation. Following this, the cocircuits of the matroid defined over the rows of $\boldsymbol{A}'$ are obtained using RCEF_{CO}. Then as before we can obtain the minimal cut sets. However, the value of k has to be reduced to $k - q$, where q denotes the number of sensors fixed as functional; and the value n has to be replaced with the current number of rows in $\boldsymbol{A}'$. Among all the minimal cut sets available, we choose the one with the largest failure probability while executing RVR.

In the RVR algorithm, we start by finding a minimal cut set of $\boldsymbol{A}$. Let $\mathcal{G} = \{s_{i_1}, s_{i_2}, ..., s_{i_h}\}$ be the minimal cut set, where $|\mathcal{G}| = h$. Let Ω_0 represents the set of component state vectors in which all the sensors indexed by $\mathcal{G}$ are set to 0, giving a set of states that result in system failure. Now, we partition the remaining sample space $\Omega - \Omega_0$ into h disjoint sets denoted by Ω_j, $j = 1, ..., h$. In Ω_j, the first $j - 1$ elements indexed by $\mathcal{G}$ are set to 0, the j^{th} element is set to 1, and the remaining elements are not fixed. It can be easily verified that $\cup_{j=0}^{h} \Omega_j = \Omega$. Now, for each of the sets Ω_j, we again find minimal cut sets and continue. We now have all the pieces to put together the RVR algorithm proposed in [5].

RVR Algorithm [5]

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

The system reliability can be estimated by an unbiased estimator $\hat{r}_{rvr}$ with a sample variance $\hat{V}_{rvr}$ calculated as

$$\hat{r}_{rvr} = \frac{\alpha}{N},\ \hat{V}_{rvr} = \frac{1}{N-1}\left[\beta - \frac{\alpha^2}{N}\right], \tag{7}$$

where α and β are obtained by recursively calling the Procedure $P(\mathcal{W}, \mathcal{F}, N)$ given below. $\mathcal{W}$ and $\mathcal{F}$ denote the sets of sensors fixed as working and failed respectively and are set to $\emptyset$ initially.

Procedure $P(\mathcal{W}, \mathcal{F}, N)$: Calculate α and β

i. Set $\mathcal{U} = \mathcal{S} - (\mathcal{W} \cup \mathcal{F})$. Let $\boldsymbol{A}_1$ be the submatrix of $\boldsymbol{A}$ consisting of rows indexed by $\mathcal{W} \cup \mathcal{U}$ and $\boldsymbol{A}_2$ those rows indexed by $\mathcal{W}$.

i.a) If $N = 0$, then $\alpha = \beta = 0$; return.

i.b) If $r(\boldsymbol{A}_2) = m$ and $|\mathcal{W}| \geq k$, then $\alpha = \beta = N$; return.

i.c) If $r(\boldsymbol{A}_1) < m$ or $|\mathcal{W} \cup \mathcal{U}| < k$, then $\alpha = \beta = 0$; return.

ii. Find a minimal cut set $\mathcal{G} = \{s_{i_1}, s_{i_2}, ..., s_{i_h}\}$ of the linear sensor system using the RCEF$_{\text{CO}}$ technique.

iii. Divide the current sample space into $h+1$ sets denoted by $\Omega_j, j = 0, 1, ..., h$ such that
$\Omega_0 = \Big\{(v_1, v_2, ..., v_n) \mid v_{i_j} = 0,\ \forall j = 1, ..., h\Big\}$ and
$\Omega_j = \Big\{(v_1, v_2, ..., v_n) \mid v_{i_l} = 0,\ \forall l = 1, ..., j-1, v_{i_j} = 1\Big\}$,
$j = 1, ..., h$.

iv. Divide the total number of samples N into N_j samples each, across Ω_j, $j = 1, ..., h$, based on the multinomial distribution defined by $M\left(N, p(\Omega_1), ..., p(\Omega_h)\right)$. The probability of Ω_j, $p(\Omega_j)$ is calculated as

$$p(\Omega_j) = \frac{p_{i_j}.\Pi_{l=1}^{j-1}(1 - p_{i_l})}{1 - Q},\ j = 1, ..., h, \tag{8}$$

where p_{i_j} is the working probability of s_{i_j} and

$$Q = \Pi_{j=1}^{h}(1 - p_{i_j}). \tag{9}$$

v. For $j = 1, ..., h$, set $\mathcal{W}_j = \mathcal{W} \cup s_{i_j}$, and $\mathcal{F}_j = \mathcal{F} \cup_{l=1}^{j-1} s_{i_l}$. Then, recursively call the procedure $P(\mathcal{W}_j, \mathcal{F}_j, N_j)$ to calculate α_j and β_j.

vi. Compute α and β as

$$\alpha = (1 - Q)\sum_{j=1}^{h}\alpha_j,\ \beta = (1 - Q)^2\sum_{j=1}^{h}\beta_j. \tag{10}$$

In RVR algorithm, we recursively calculate the values of α_j and β_j by calling the procedure $P(\mathcal{W}_j, \mathcal{F}_j, N_j)$. If a set of component states guarantee system functioning, then in step (1.b), α and β are set to the number of samples N_j. If a set of component states is certain to cause system failure, then in step (i.c), α and β are set to zero.

The approach RCEF$_{\text{CO}}$ is not the ideal one to find the best possible cut sets. Consider a case when all the sensors have the same working probability. Then, a minimal cut set with the smallest cardinality will have the maximum failure probability. However, the RVR algorithm quite often fails to find this minimal cut set. We, in this paper, focus on systems with same sensor probabilities; nonetheless, our approach can be applied to general cases. To find minimal cut sets with the smallest cardinality, we first need to get the smallest cocircuits. Finding the smallest cocircuits of a vector matroid is a well-studied problem in robust regression analysis.

Kianfar et al. [6] proposed a 0-1 mixed integer programming algorithm (denoted as MIP) to find the smallest cocircuit. Their approach is based on the fact that the smallest cocircuit problem can be solved by finding the minimum number of rows that if deleted from A, the remaining matrix has a nonzero null space (for more details, please refer to [6]). The MIP formulation presented in [6] is as follows:

$$\min \sum_{i=1}^{n} q_i \tag{11}$$

$$\text{s.t.} \qquad -q_i \leq \sum_{j=1}^{m} a_{ij}x_j \leq q_i,\ i = 1, ..., n \tag{12}$$

$$-1 + 2z_j \leq x_j \leq 1,\ j = 1, ..., m;\ \sum_{j=1}^{m} z_j = 1 \tag{13}$$

$$x_j \in \mathbb{R},\ q_i, z_j \in \{0, 1\},\ i = 1, ..., n;\ j = 1, ..., m. \tag{14}$$

In this formulation, q_i is a binary variable that takes the value 1 if row i is chosen to be removed from the matrix $\boldsymbol{A}$. The vector $\boldsymbol{x} = (x_1, x_2..., x_n)$ represents the nonzero member of the null space of the reduced matrix obtained by removing all rows with $q_i = 1$ from $\boldsymbol{A}$. If $q_i = 0$, then we must have $\boldsymbol{a}_i\boldsymbol{x} = 0$; but if $q_i = 1$, we can have $\boldsymbol{a}_i\boldsymbol{x} \neq 0$. This is captured in constraint (12). Constraints (13) guarantee a nonzero $\boldsymbol{x}$. Then the smallest cocircuit can be obtained as the set of sensors for which $q_i = 1$.

We can improve the RVR algorithm by replacing the RCEF$_{\text{CO}}$ technique with this MIP formulation. One way to find cocircuits for systems with fixed sensor states using MIP is to replace $\boldsymbol{A}$ with $\boldsymbol{A}'$ as in RCEF$_{\text{CO}}$. But, we present a more practical approach later. Our second scheme to improve the RVR algorithm is to find the smallest cocircuits of $\boldsymbol{A}$ by solving a circuit problem over the dual matroid of $\boldsymbol{A}$. In Section III, we explain these approaches in greater detail.

III. Reliability Evaluation Using Mixed Integer Programming

The MIP algorithm [6], as discussed in Section II, can be used to find the minimal cut sets within the RVR approach. When no sensors are fixed, we can solve the MIP over $\boldsymbol{A}$ to find the smallest cocircuit, obtained as $\mathcal{C} = \{s_i : q_i = 1,\ i = 1, ..., n\}$. These cocircuits can indeed provide the minimal cut sets of the linear sensor system. Now, consider a case with some sensors fixed as functional or failed. We can remove the rows in $\boldsymbol{A}$ corresponding to the failed sensors before MIP can be applied. As for the sensors whose states are fixed as working, instead of pivoting as in RCEF$_{\text{CO}}$, we can set q_i's indexed by the respective sensors to 0. By doing so, we avoid finding cocircuits involving s_i's; this is equivalent to fixing the corresponding sensors as working. For the sensors fixed as failed, the row removal can also be replaced by setting q_i's to 1. But, it is more desirable to remove the rows altogether as the size of the overall MIP gets reduced. Once the cocircuits are obtained, we can find the minimal cut sets as in RCEF$_{\mathrm{CO}}$. Now we present our first algorithm RVR$_{\mathrm{MIP}}$.

RVR$_{\mathrm{MIP}}$ Algorithm

Input: Given a sensor system defined by the matrix $\boldsymbol{A}$ having n sensors with working probabilities p_i, $i = 1, ..., n$. A sample size of N is chosen.

The system reliability can be estimated by an unbiased estimator $\hat{r}_{mip}$ with a sample variance $\hat{V}_{mip}$ calculated as

$$\hat{r}_{mip} = \frac{\alpha}{N}, \ \hat{V}_{mip} = \frac{1}{N-1}\left[\beta - \frac{\alpha^2}{N}\right], \tag{15}$$

Procedure $P(\mathcal{W}, \mathcal{F}, N)$: Calculate α and β

i. Set $\mathcal{U} = \mathcal{S} - (\mathcal{W} \cup \mathcal{F})$. Get $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ as in RVR algorithm.
   i.a) If $N = 0$, then $\alpha = \beta = 0$; return.
   i.b) If $r(\boldsymbol{A}_2) = m$ and $|\mathcal{W}| \geq k$, then $\alpha = \beta = N$; return.
   i.c) If $r(\boldsymbol{A}_1) < m$ or $|\mathcal{W} \cup \mathcal{U}| < k$, then $\alpha = \beta = 0$; return.

ii. Remove all the rows in $\boldsymbol{A}$ corresponding to sensors in $\mathcal{F}$. Set the binary variable q_i representing those sensors in $\mathcal{W}$ to 0. Then find a minimal cut set $\mathcal{G} = \{s_{i_1}, s_{i_2}, ..., s_{i_h}\}$ of the linear sensor system using the MIP formulation.

iii. Divide the current sample space into sets $\Omega_j, j = 0, 1, ..., h$ using the same strategy as in RVR algorithm.

iv. Divide N across each Ω_j based on the multinomial distribution defined by $M\left(N, p(\Omega_1), ..., p(\Omega_h)\right)$. The probability of Ω_j, $p(\Omega_j)$ is calculated following the same steps as in RVR algorithm. Get $Q = \Pi_{j=1}^{h}(1 - p_{i_j})$.

v. For $j = 1, ..., h$, set $\mathcal{W}_j = \mathcal{W} \cup s_{i_j}$, and $\mathcal{F}_j = \mathcal{F} \cup_{l=1}^{j-1} s_{i_l}$. Then, recursively call the procedure $P(\mathcal{W}_j, \mathcal{F}_j, N_j)$ to calculate α_j and β_j.

vi. Compute α and β as

$$\alpha = (1 - Q)\sum_{j=1}^{h}\alpha_j, \ \beta = (1 - Q)^2\sum_{j=1}^{h}\beta_j. \tag{16}$$

The RVR$_{\mathrm{MIP}}$ algorithm replaces the inefficient RCEF$_{\mathrm{CO}}$ method with the MIP approach. When all the probabilities p_i are the same, this algorithm finds the minimal cut set with the largest failure probability. Even if this is not the case, the smallest cocircuits obtained will provide minimal cut sets with very large failure probabilities.

RVR$_{\mathrm{DMIP}}$ uses matroid theory duality to find minimal cut sets by solving a circuit problem over the dual of $\boldsymbol{A}$. A matroid $M = (E, \mathcal{F})$ is an ordered pair consisting of a ground set E and a collection $\mathcal{F}$ of independent subsets of E. In order for M to be a matroid it should satisfy the independent augmentation axiom; i.e., if I_1 and I_2 are in $\mathcal{F}$ and $|I_1| < |I_2|$, then there exists an element $e \in I_2 - I_1$ such that $I_1 \cup e \in \mathcal{F}$. A minimal dependent set in a matroid is called its circuit. *For every matroid, there is an associated dual matroid such that the circuits of the matroid are also the cocircuits of its dual.* Vector matroids are matroids defined over a matrix, and by convention the columns form the ground set E. For more details, please refer to [8].

Since we are interested in the dependence relationship among the rows of $\boldsymbol{A}$, we consider the matroid $M[\boldsymbol{A}^T]$ defined over the columns of $\boldsymbol{A}^T$. Now, we can find the dual matroid of $M[\boldsymbol{A}^T]$ as: Get the reduced row echelon form of $\boldsymbol{A}^T$ denoted by $[\boldsymbol{I}_m|\boldsymbol{D}]$, where $\boldsymbol{I}_m$ is the $m \times m$ identity matrix. Then, the dual of $M[\boldsymbol{A}^T]$ is the matroid defined over the columns of $[-\boldsymbol{D}^T|\boldsymbol{I}_{n-m}]$, denoted by $M^*[-\boldsymbol{D}^T|\boldsymbol{I}_{n-m}]$. By duality, the cocircuits of $M[\boldsymbol{A}^T]$ are also the circuits of $M^*[-\boldsymbol{D}^T|\boldsymbol{I}_{n-m}]$. Due to this association, one can find the circuits over the $(n-m) \times n$ matrix $\boldsymbol{H} = [-\boldsymbol{D}^T|\boldsymbol{I}_{n-m}]$ to obtain the minimal cut sets of the linear system defined over $\boldsymbol{A}$. This is the basis of the RVR$_{\mathrm{DMIP}}$ algorithm.

First, let us formulate the smallest circuit problem as a mixed integer problem. We call this DMIP, where "D" stands for the dual. The DMIP formulation is as follows:

$$\min \sum_{i=1}^{n} q_i \tag{17}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} h_{ji} y_i = 0, \ j = 1, ..., n - m \tag{18}$$

$$-q_i + 2z_i \leq y_i \leq q_i, \ i = 1, ..., n; \ \sum_{i=1}^{n} z_i = 1 \tag{19}$$

$$y_i \in \mathbb{R}, \ q_i, z_i \in \{0, 1\}, \ i = 1, ..., n. \tag{20}$$

In DMIP, we find a nonzero $\boldsymbol{y} = (y_1, ..., y_n)$ in the null space of $\boldsymbol{H}$ that has the smallest l_0-norm. Then, the smallest circuit $\mathcal{C}$ over $\boldsymbol{H}$ can be obtained as $\mathcal{C} = \{s_i : q_i = 1, \ i = 1, ..., n\}$. Since $\mathcal{C}$ is also the smallest cocircuit over $\boldsymbol{A}$, we can now find the minimal cut sets of $\boldsymbol{A}$ as before. If some of the sensor states are fixed, then the DMIP approach has to be modified to address these cases. As we know, the rows in $\boldsymbol{A}$ corresponding to failed sensors is removed, and rows representing working sensors is pivoted before applying RCEF$_{\mathrm{CO}}$. We represented this reduced matrix by $\boldsymbol{A}'$. The pivoting operation is called a contraction in matroid theory. The contraction is the dual of deletion. So the equivalent dual matroid for $\boldsymbol{A}'$ can be obtained by performing the dual operations. Specifically, for sensors set as failed, the corresponding rows in $\boldsymbol{H}^T$ are pivoted (or equivalently pivot over the columns of $\boldsymbol{H}$); and for sensors set as working, the corresponding rows in $\boldsymbol{H}^T$ are deleted, to get the dual matroid of $\boldsymbol{A}'$. These fixed sensors cases can also be equivalently addressed by setting $q_i = 0$ for those sensors set as working, and $q_i = 1$ for those that are failed. However, in the RVR$_{\mathrm{DMIP}}$ Algorithm, for sensors fixed as working, we remove the corresponding column from $\boldsymbol{H}$, and for the sensors set as failed, we set $q_i = 1$.

RVR$_{\mathrm{DMIP}}$ Algorithm

Input: Given a sensor system defined by the matrix $\boldsymbol{A}$. Get the dual matrix $\boldsymbol{H} = [-\boldsymbol{D}^T|\boldsymbol{I}_{n-m}]$. A sample size is N.

Calculate $\hat{r}_{dmip}$ and $\hat{V}_{dmip}$ as

$$\hat{r}_{dmip} = \frac{\alpha}{N}, \ \hat{V}_{dmip} = \frac{1}{N-1}\left[\beta - \frac{\alpha^2}{N}\right], \tag{21}$$

Procedure $P(\mathcal{W}, \mathcal{F}, N)$: Calculate α and β

Follow the same steps as in RVR_{MIP}, but replace MIP in step (ii) with DMIP over $\boldsymbol{H}$.

Before applying DMIP, remove all the columns in $\boldsymbol{H}$ corresponding to sensors in $\mathcal{W}$. Set the binary variable q_i representing those sensors in $\mathcal{F}$ to 1.

The size of the matrices $\boldsymbol{H}$ and $\boldsymbol{A}$, as well as the respective formulations, will have an impact on the runtime of the algorithms RVR_{MIP} and RVR_{DMIP}. Though both these approaches find the smallest cocircuits (smallest circuits over $\boldsymbol{H}$ in case of RVR_{DMIP}), the estimated values of reliability and variance may slightly differ. One reason for this is the algorithms choosing different cocircuits (or circuits) when there are multiple smallest ones available. However, our analysis indicates that the runtimes, the reliability estimates, and the sample variance for these two algorithms are highly comparable. More importantly, these values are significantly superior when compared to RVR. In the next section, we detail our computational analysis.

IV. Computational Experiments

In this section, we present the results of our computational experiments conducted to compare the performance of our algorithms (RVR_{MIP} and RVR_{DMIP}) with each other and with the previously proposed RVR algorithm discussed in Section II. We performed our computations on a testbed consisting of 5 categories of randomly generated instances. Each category consists of five individual instances of identical size. All the instances are based on the multi-station assembly examples reported in [9]. We implemented all the algorithms in C++ and used CPLEX 12.6.1 as integer programming solver wherever needed. All the algorithms were run on a PC which has an Intel Core$^{\text{TM}}$ i7-3667U 2.50GHz processor and 4 GD of RAM. We choose two different sample sizes 100,000 and 1,000,000. The reliability of each sensor, i.e., the working probability of each sensor is set as 0.95. The value of k is fixed at $m+1$.

Tables I and II summarize the results of our computational experiments. Each row in these tables corresponds to an instance category. For each algorithm, we report the average runtime (t), the average percentage reliability estimate ($\hat{r}$), and the average variance of estimated system reliability ($\hat{V}$) against the corresponding algorithm and for each instance category. All the runtimes are reported in minutes. In Table 1, the values reported are for the sample size of 100,000 and, in Table 2, those reported are for the sample size of 1,000,000. The subscripts rvr, mip, and $dmip$ under t, $\hat{r}$, and $\hat{V}$ identifies the corresponding algorithms RVR, RVR_{MIP}, and RVR_{DMIP}. To compare the overall performance of our algorithms relative to RVR, we introduced the factor Γ. The values Γ_{mip} and Γ_{dmip} in the tables are calculated as:

$$\Gamma_{mip} = (t_{mip} \times \hat{V}_{mip})/(t_{rvr} \times \hat{V}_{rvr}) \text{ and} \tag{22}$$

$$\Gamma_{dmip} = (t_{dmip} \times \hat{V}_{dmip})/(t_{rvr} \times \hat{V}_{rvr}). \tag{23}$$

A value of $\Gamma < 1$ signifies the superior performance of the respective algorithm compared to RVR; the smaller the value, the better the performance. We also compute another parameter N_{req}, reported in both the tables under the RVR algorithm, which gives an approximate number of samples required by this algorithm to attain an estimated variance comparable to the average value of the variance estimate reported for RVR_{MIP} and RVR_{DMIP}.

An analysis of the values for the runtime and the variance estimate indicates that both of our algorithms, RVR_{MIP} and RVR_{DMIP}, are superior to the RVR algorithm. The performance of all the algorithms improves a large sample size is used, as is expected of any Monte Carlo based algorithms. Evidently, a comparison of the values in Tables I and II points to this fact. The runtimes for our algorithms are slightly larger than the values reported for RVR for the smaller instances 26×12 and 66×27. For instances in the category 157×72, these values are almost comparable. However, for the other two larger instances, our algorithms outperform RVR in terms of the runtime for both the sample sizes. But the critical performance parameter $\hat{V}$ is significantly smaller for our algorithms, remarkably so for the case with a sample size of 1,000,000. This smaller value points to a more definitive estimate of reliability. Moreover, the value of reliability obtained using our algorithms is slightly smaller than the ones obtained by RVR. This decrease is to be expected, as our algorithms search through a broader set of component states as compared to RVR for a given sample size due to the availability of superior minimal cut sets. Also, notice that the values or reliability obtained for the sample size of 1,000,000 is smaller than the ones with a sample size of 100,000. This is because we explore more minimal cut sets when a larger sample size is used.

The runtimes, the reliability and the variance estimates for our algorithms are highly comparable, as noted in the previous section. For example, for all the instances tested, the values for the estimated reliability from RVR_{MIP} and RVR_{DMIP} differ by less than 0.02% for both sample sizes. The overall test performance captured by Γ is the smallest for the category 154×72; this is also the most reliable one among the test categories. The largest Γ values are recorded for the least reliable instance categories, 316×144 and 485×360. However, even for these instances the value of Γ obtained is smaller than 0.29 for the sample size of 100,000 and is smaller than 0.14 for the sample size 1,000,000. The value of N_{req}, another performance metric, shows that the number of samples required by RVR to reach an estimated variance comparable to that of our algorithms is also significantly higher. For the instances in the category 154×72, on average we needed around 326,000 samples for RVR to attain an estimated variance comparable to the one obtained for our algorithms when a sample size of 1000,000 is used. Meanwhile, in order for RVR to match the value of estimated variances obtained by our algorithms using 1,000,000 samples, we needed more than 1,400,000 samples for all the instances in the first three categories. These numbers are yet another indicator of the efficiency of our algorithms in finding solid estimates for the reliability.

TABLE I: Performance Comparison of Various Algorithms for $N = 100000$

$n \times m$	RVR				RVR$_{\text{MIP}}$				RVR$_{\text{DMIP}}$			
	t_{rvr}	$\hat{r}_{rvr}$	$\hat{V}_{rvr}$	N_{req}	t_{mip}	$\hat{r}_{mip}$	$\hat{V}_{mip}$	Γ_{mip}	t_{dmip}	$\hat{r}_{dmip}$	$\hat{V}_{dmip}$	Γ_{dmip}
26 × 12	0.73	99.122	3.517E-03	264000	1.18	99.108	4.653E-04	0.214	1.21	99.103	4.617E-04	0.217
66 × 27	2.33	99.156	2.411E-03	276000	2.52	99.127	3.542E-04	0.160	2.54	99.122	3.553E-04	0.161
154 × 72	2.90	99.518	1.143E-03	326000	3.01	99.472	1.210E-04	0.110	2.99	99.462	1.183E-04	0.107
316 × 144	5.92	97.321	3.031E-02	186000	5.353	97.136	8.727E-03	0.261	5.64	97.128	8.885E-03	0.279
485 × 360	6.514	97.242	2.963E-02	164000	6.042	97.102	9.041E-03	0.283	5.79	97.066	9.020E-03	0.271

TABLE II: Performance Comparison of Various Algorithms for $N = 1000000$

$n \times m$	RVR				RVR$_{\text{MIP}}$				RVR$_{\text{DMIP}}$			
	t_{rvr}	$\hat{r}_{rvr}$	$\hat{V}_{rvr}$	N_{req}	t_{mip}	$\hat{r}_{mip}$	$\hat{V}_{mip}$	Γ_{mip}	t_{dmip}	$\hat{r}_{dmip}$	$\hat{V}_{dmip}$	Γ_{dmip}
26 × 12	1.45	99.104	1.632E-04	1411000	2.09	99.093	8.285E-06	0.073	2.11	99.091	8.555E-06	0.076
66 × 27	4.91	99.132	8.562E-05	1554000	5.34	99.102	5.338E-06	0.068	5.23	99.098	4.587E-06	0.057
154 × 72	7.09	99.452	3.421E-05	1721000	6.19	99.431	1.543E-06	0.039	6.07	99.430	1.708E-06	0.043
316 × 144	15.88	97.288	2.021E-03	1207000	15.13	97.106	2.531E-04	0.119	14.93	97.094	2.877E-04	0.134
485 × 360	17.74	97.170	2.120E-03	1193000	15.16	96.904	3.004E-04	0.121	14.77	96.923	2.742E-04	0.108

V. Concluding Remarks

The problem of estimating the reliability of a linear sensor system is considered. To obtain a dependable estimate of the system reliability, we proposed mixed integer programming techniques to improve the existing recursive variance reduction algorithm. The key to a successful recursive reduction method is to find minimal cut sets that can yield the greatest reduction in the sample space of the crude Monte Carlo method. We introduced two advanced recursive variance reduction algorithms, RVR$_{\text{MIP}}$ and RVR$_{\text{DMIP}}$, both of which can provide highly accurate estimates for reliability. The RVR$_{\text{MIP}}$ algorithm solves a mixed integer program over the matrix that designs the linear system thereby finding the minimal cut sets of smallest cardinality. The RVR$_{\text{DMIP}}$ algorithm gets these minimal cut sets by solving an equivalent dual problem. The more reliable the linear sensor system, the better the performance of our algorithms.

References

[1] J. J. Cho, Y. Chen, and Y. Ding, "On the breakdown point of structured linear models," *Technometrics*, vol. in - press, 2009.

[2] Y. Chen and Q. Yang, "Reliability of two-stage weighted-k-out-of-n systems with components in common," *IEEE Transactions on Reliability*, vol. 54, no. 3, pp. 431–440, 2005.

[3] 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.

[4] 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.

[5] 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.

[6] K. Kianfar, A. Pourhabib, and Y. Ding, "An integer programming approach for analyzing the measurement redundancy in structured linear systems," *IEEE Transactions on Automation Science and Engineering*, vol. 8, no. 2, pp. 447–450, Apr. 2011.

[7] 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.

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

[9] 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.