A UNIFIED AND EFFICIENT ALGORITHMIC APPROACH TO MODEL-BASED DIAGNOSIS AND OPTIMAL SENSOR PLACEMENT

Amir Fijany(1) and Farrokh Vatan(2)

(1) Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA 91109, USA,
Email: Amir.Fijany@jpl.nasa.gov

(2) Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA 91109, USA,
Email: Farrokh.Vatan@jpl.nasa.gov

ABSTRACT

This paper presents a new unified approach for solving diagnosis and sensor placement problems. In previous works [3], we have shown that the main computational challenge in model-based diagnosis, that is, the calculation of minimal diagnosis set, can be represented as the solution of the Hitting Set Problem. We have also shown that this calculation can be mapped onto the Integer Programming problem and have further developed an efficient branch-and-bound technique for its solution. In this paper, we first show that the optimal sensor placement can be also mapped onto the integer Programming problem. We then extend our previously developed branch-and-bound technique for efficient solution of the optimal sensor placement problem. To our knowledge, this is the first systematic (i.e., non-exhaustive search) approach for the solution of the optimal sensor placement problem. Our new method can be used both at the design level, where the number and position of sensors are not known, and the case where we want to add more sensors to the system to enhance its diagnosability.

1. INTRODUCTION

In the current state of practice, the most disciplined approach to diagnosis is the “model-based” approach [10], employing knowledge of device operation and connectivity in the form of models. This approach, which reasons from first principles, provides far better diagnostic coverage than traditional approaches based upon collection of symptom-to-suspect rules. However, there is a major drawback in current model-based diagnosis systems that severely limits their practicality. In fact, in order to find the minimal diagnosis set, which corresponds to the solution of the Hitting Set Problem, they rely on algorithms which are not efficient for many systems of interest.

On the other hand, the quality and efficiency of a diagnosis system depends on the availability and relevance of the information it can retrieve from the diagnosed plant. To this end, given the impossibility of deploying the maximum desired number of sensors (i.e., one sensor per system’s component), optimal sensors selection and placement is not only of prime importance for an efficient system design but it also directly determines the diagnosability degree of the system. The main computational challenge in optimal sensor placement is due to the fact that it requires solution of a “covering problem” which can be formulated as an optimization problem with a non-linear objective function.

In this paper, we present a unified and efficient algorithmic approach for the solution of the diagnosis (the Hitting Set) and the optimal sensor placement (the Set Covering) problems. To our knowledge there has not been a unified algorithmic approach for their solution. Since in previous papers (see, e.g., [3] and references there) we have discussed the Diagnosis Problem in details, in this paper we focus on the Sensor Placement Problem.

We first show that the Hitting Set and Sensor Covering problems can be both mapped onto special cases of 0/1 Integer Programming Problem; in the case of Hitting Set, the objective function is linear while in the case of Sensor Covering, is general, it is not. This mapping enables us, for the first time, a priori determination of the lower and upper bounds on the size of the solution. Based on these bounds, we introduce the concept of solution window for the problem. We also propose a new branch-and-bound technique that not only is faster than the current techniques in terms of number of operations (by exploiting the structure of the problem) but also, using the concept of solution window, allows a massive reduction (pruning) in the number of branches. Furthermore, as the branch-and-bound proceeds, the solution window is dynamically updated and narrowed to enable further pruning.

The results of the performance of the new algorithm on a set of test cases for the Hitting Set problem [3] show the advantage of our new algorithm over the traditional branch-and-bound algorithm; in fact the new algorithm has achieved several orders of magnitude speedup over the standard algorithms. However, we believe that the main advantage of this new algorithmic approach is not...
only its efficiency in solving both problems but also the fact that, for the first time, it establishes an explicit relation between the solutions of these problems. Hence it can provide better insight into an improved (in terms of diagnosability) system design as well as a priori assessment of limits and capabilities of the diagnosis engine for system diagnosis.

2. DIAGNOSIS PROBLEM

The diagnosis process starts with identifying symptoms that represent inconsistencies (discrepancies) between the system's model (description) and the system's actual behavior [10]. Each symptom identifies a set of conflicting components as initial candidates. A minimal diagnosis is the smallest set of components that intersects all conflict sets (or a Hitting Set of the conflicts). The underlying general approach in different model-based diagnosis systems can be described as a two-step “divide-and-conquer” technique wherein finding the minimal diagnosis set is accomplished in two steps: a) Generating conflict sets from symptoms, and b) Calculating minimal diagnosis set from the conflict sets. In summary, the conflict generation corresponds to forming a collection of sets, and calculating minimal diagnosis corresponds to solving the hitting set problem for this collection (see Fig. 1).

To overcome the complexity of calculating minimal diagnosis set, we will utilize and expand our new discovery relating this calculation and the solution of the Hitting Set Problem to the solution of Integer Programming Problem (see [3] and references there). Let $x = (x_1, x_2, \ldots, x_n)$ be a binary vector, wherein $x_j = 1$ if the member $m_j$ belongs to the minimal hitting set and hence the minimal diagnosis set, otherwise $x_j = 0$. It can be then shown [3] that we have the following formulation of the Hitting Set Problem as a 0/1 Integer Programming Problem:

$$
\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{n} x_i \\
\text{subject to} & \quad Ax \geq b, \quad x_j = 0 \text{ or } 1,
\end{align*}
$$

where $b = (1,\ldots,1)$ is a vector whose components are all equal to one. This new mapping allows us to utilize existing efficient integer programming algorithms [8], permitting solution of problems with a much larger size. Moreover, we have developed a new branch-and-bound technique [3] for solving this problem with a much better performance than the traditional algorithms.

3. DIGANOSABILITY

The quality and efficiency of a diagnosis system depends on the availability and relevance of the information it can retrieve from the diagnosed plant. The quality of the measurements is expressed by the diagnosability degree, i.e., given a set of sensors, which faults can be discriminated? There is no straightforward relation between the number of sensors and diagnosability of the systems; increasing the number of sensors alone does not guarantee a higher level of diagnosability. The relevance of information provided by an additional sensor and its correlation with information provided by other sensors must also be taken into account. Besides the issue of diagnosability, we should also consider the economical issues. We must provide a sensor system that achieves a desired degree of diagnosability at the lowest possible cost. The different issues regarding sensor placement problem can be summarized as follows:

- **Diagnosability Degree.** Determining the diagnosability degree of a system, i.e., characterizing the set of the faults that can be discriminated.
- **Minimal Sensor Set.** Finding a minimal additional sensor set that guarantees a specific degree of diagnosability.
- **Minimal Cost Sensors.** In the case that different sensors are assigned with different costs, finding the minimal cost additional sensors which achieve a specific degree of diagnosability.

Our new approach for solving these problems is motivated by our successful method for solving the diagnosis problem [3]. The structural analysis of the
system and the potential information carried by each sensor provide a set of relations usually called the Analytical Redundant Relations (ARRs) (see, e.g., [2]). We can also consider the additional sensors (the potential sensors that will provide the desired degree of diagnosability) and their corresponding ARRs. The information of all these ARRs can be summarized in a fault signature matrix [1, 4–7]. Then the above sensor optimization problems can be formulated as a combinatorial problem regarding the signature matrix. The existing methods for solving these combinatorial problems usually boil down to exhaustive search methods [4, 5]. Here, we first present a new and rigorous mathematical representation of the problem which enables us to map it onto the Integer Programming problem. For solving the diagnosis problem, we have developed a new branch-and-bound technique which has achieved an order of magnitude speedup over the standard algorithms. We extend this technique for the solution of the sensor optimization problem through its mapping to the Integer Programming problem. This would provide a powerful efficient technique for solving the difficult problem of sensor placement optimization.

4. FAULT SIGNATURE MATRIX

In the model-based approach, the behavior of a system is described by a set of relations among variables (parameters) of the system. These relations, not only describe the nominal behavior of each component, but they also determine the structure and interconnections of the components. A simple example explains this notion. Consider the circuit represented in Fig. 2 (this is an example of a polybox system; see, e.g., [2]). This system consists of 7 components, which are 4 multiplier gates, \( M_1, M_2, M_3, \) and \( M_4 \) and 3 adder gates, \( A_1, A_2, \) and \( A_3 \). Let \( x, y, z, \) and \( t \) denote the outputs of the multiplier gates and \( f, g, \) and \( h \) denote the outputs of the adder gates. The model of this system can be described by the following relations:

\[
\begin{align*}
M_1: & \quad x = a \times b, \\
M_2: & \quad y = b \times c, \\
M_3: & \quad z = c \times d, \\
M_4: & \quad t = d \times e, \\
A_1: & \quad r_1 = f = x + y, \\
A_2: & \quad r_2 = g = y + z, \\
A_3: & \quad r_3 = h = z + t.
\end{align*}
\]

Note that these relations also fully describe the interconnections of the components, and hence they reveal the underlying structure of the system. We call these relations the primary relations (PRs) of the system and we define the set of PRs and the variables of the system as

\[
E = \{r_1, r_2, r_3, r_4, r_5, r_6, r_7\}, \quad V = \{x, y, z, t, f, g, h\}. \quad (2)
\]

Once the sensors are introduced, the set of variables \( V \) can be partitioned as \( V = O \cup U \), where \( O \) is the set of observed variables (i.e., variables measured by the sensors) and \( U \) the set of unknown (non-measured) variables. If the number of relations in \( E \) is more than the number of unknown variables \( U \), then it is possible to find relations among observed variables \( O \). These relations are called Analytical Redundant Relations (ARRs). The basic property of an ARR is that it can be evaluated from the observed values. Without loss of generality, we can assume that an ARR is of the form \( \Phi(a_1, a_2, \ldots, a_k) = 0 \), where \( a_j \in O \), and for a given set of values for \( a_j \) either \( \Phi(a_1, a_2, \ldots, a_k) = 0 \) is satisfied or not.

We consider a set \( F \) of faults. In most cases, each fault \( F \in F \) is identified by malfunction of a set of components \( C_1, \ldots, C_k \) and is denoted as \( F_{c_1, \ldots, c_k} \). The Fault Signature Matrix, or simply the Signature Matrix, is a 0/1 matrix where the rows are labeled by ARRs and columns are labeled by the faults, such that for the entry \( s_{ij} \) at row labeled by the ARR \( R_i \) and the column labeled by the fault \( F_j \), we have \( s_{ij} = 1 \) if a component involved in the fault \( F_j \) appears in \( R_i \), otherwise \( s_{ij} = 0 \). Therefore, each column of signature matrix is a binary Fault Signature Vector (FSV) containing 1 for each ARR sensitive to that fault and 0 for insensitive relations. For example, suppose that in the system of Fig. 2 there are 3 sensors which measure the output variables \( f, g, \) and \( h \). Then the following relations are the ARRs:

\[
\begin{align*}
f - ab - bc = 0, & \quad g - bc - de = 0, & \quad h - cd - de = 0, \\
f - g - ab + cd = 0, & \quad g - h - bc + de = 0.
\end{align*}
\]

If we consider only single component failures and represent each fault with its corresponding component, then the signature matrix is as follows:
One obvious way to achieve maximum discriminability is to measure all the variables of the system. However, it may be possible to achieve the same level of discriminability by measuring a subset of all variables. Therefore, the problem of finding the optimal sensors is equivalent to finding minimal number of additional sensors which provide the same level of discriminability as the case where every variable is measured. This is done by assuming that all unknown variables of the system have a hypothetical sensor and the corresponding (hypothetical) ARRs are considered. The new signature matrix which is obtained is called the Hypothetical Signature Matrix (HSM).

For example, in the case of the system of Fig. 2, to generate the hypothetical signature matrix (HSM), we assume that all variables $x, y, z, t, f, g,$ and $h$ are observed and there are corresponding sensors (denoted by $S(x), S(y),$ and so on). Then all relations in (3) are ARRs; moreover, we get the following additional ARRs:

\[
\begin{align*}
    f - g - x + z &= 0, \\
    f - ab - bc &= 0, \\
    g - y + t &= 0, \\
    g - bc - de &= 0, \\
    h - cd - de &= 0, \\
    f - g + ab + cd &= 0, \\
    g - h - bc + de &= 0.
\end{align*}
\]

Therefore, we obtain the HSM of Table 1.

The set of sensors corresponded with each ARR (rows of Table 1) are as follows, respectively:

\[
\{x, y, \{t\}, \{x, y, f\}, \{y, z, g\}, \{z, t, h\}, \{x, z, f, g\}, \{y, t, g, h\}, \{f\}, \{g\}, \{f, g\}, \{g, h\}\}.
\]

5. SENSOR COVERING AND DISCRIMINABILITY PROBLEMS

Once the hypothetical signature matrix is introduced, it is then possible to formulate the essential problems regarding sensor placement. Here we address two basic problems: covering (detecting) all faults in a given set of faults $F$, and discriminating between faults in $F$. The first problem requires that for any fault $F \in F$ the fault signature vector of $F$ should contain at least one non-zero component; i.e., the measurement of at least one sensor should detect this fault. Discrimination between faults $F_1, F_2 \in F$ is possible if and only if all the fault signature vectors of $F_1$ and $F_2$ are different. The following theorem provides an algebraic-combinatorial formulation for these problems in term of signature matrix.

**Theorem 1.** Let $F$ be a set of faults of a system and $S$ be a set of sensors of the system. Suppose that $M$ is the fault signature matrix associated with $F$ and $S$. Then we have the following equivalent formulations for the covering and discriminability problems:

(i) The sensors $S$ can cover (detect) all faults in $F$ if and only if there is no all-zero column in $M$.

(ii) The sensors $S$ can discriminate between faults in $F$ if and only if all columns of $M$ are distinct.

The practical importance of this theorem becomes clear when we try to apply it to a hypothetical signature matrix to obtain an optimal set of new sensors. As we mentioned earlier, by considering all possible (hypothetical) sensors, we find the maximum level of diagnosability of the system. Then the task of sensor optimization is to eliminate maximum number of sensors such that, at the end, the new set of the sensors has the same level of diagnosability as all possible sensors. The (hypothetical) situation that we have all possible sensors is encapsulated in the hypothetical signature matrix. Therefore, we can formulate the sensor placement optimization problem as follow.

**Sensor Placement Optimization Problem.** Given the hypothetical signature matrix $H$ of a system, find a submatrix of $H$ by deleting rows such that the resulting matrix has all distinct columns, none of the columns is an all-zero column, and the number of corresponding sensors is minimal.

We have to add the following caveats to the above statement in order to avoid possible misunderstanding.

<table>
<thead>
<tr>
<th>ARR</th>
<th>$M_1$</th>
<th>$M_2$</th>
<th>$M_3$</th>
<th>$M_4$</th>
<th>$A_1$</th>
<th>$A_2$</th>
<th>$A_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f - ab - bc = 0$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$g - bc - de = 0$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$h - cd - de = 0$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$f - g + ab + cd = 0$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 1. The Hypothetical Signature Matrix (HSM) of the system of Fig. 2.
1. If the hypothetical signature matrix $H$ itself has repeated columns, then it means that the structure of the system does not allow to discriminate between faults corresponding with those repeated columns, no matter how we choose the sensors. In situations like this, we keep one of the repeated columns and we delete the rest of them. The result is a matrix $H'$ whose all columns are distinct.

2. Removing the rows of $H$ represents dropping the corresponding sensors from the final set of sensors. In fact, the consequence of removing one sensor can be deleting more than one row of $H$; i.e., all rows which depend on that sensor. For example, for the case of the HSM represented in Table 1, if we remove the sensor $S_i(x)$ then we have to delete rows $1$, $5$, and $8$.

For the signature matrix $H$ defined in Table 1, the optimal solution is defined by the rows $\{1, 4, 10, 11, 12\}$ and the corresponding set of sensors is $\{S_1(x), S_2(t), S_3(g), S_4(h)\}$. The submatrix of $H$ defined by these rows is

$$
\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 0
\end{pmatrix}
$$

which has the required properties. In this case, the solution set of rows is not unique and there are 13 other sets of rows which also determine the same optimal set of sensors.

6. A NEW BRANCH-AND-BOUND METHOD

Here, we present a new method for Sensor Placement Optimization Problem. The method starts with mapping of the problem onto a special case of Integer Programming. We then present a variant of the well-known branch-and-bound technique for its solution. We have already developed a similar branch-and-bound technique for diagnosis problem [3].

A. Integer Programming Formulation

To describe the mapping of the Sensor Placement Optimization Problem as an integer programming problem, let us consider a signature matrix $H$ of a system. Let $M = H'$, i.e., the $n \times m$ matrix $M$ is the transposed of $H$. For every row $R$ of $H$, or equivalently every column $C$ of $M$, a corresponding set of sensors $S(R)$ or $S(C)$ is defined. Then an equivalent formulation of the problem is as follows: choose a subset of the columns of $M$ such that the submatrix defined by these columns has no zero rows, all its columns are distinct, and the total number of corresponding sensors is minimal. Let us define a binary vector $x=(x_1, x_2, \ldots, x_n)$ whose dimension is the same as the number of columns of the matrix $M$. Then we can interpret $x$ as a selection of a subset of columns of $M$: $x_j = 1$ if and only if the $j$th column of $M$ is chosen. Then the condition $Mx \geq 1$ (where $1=(1,1,\ldots,1)^T$ is an all-one vector of appropriate dimension) implies that the solution defined by $x$ has this property that the corresponding submatrix has no all-zero row.

Before we describe the branch-and-bound method, we show how it is possible to obtain lower and upper bounds for the solution of the problem. As it will be explained later, these bounds are essential for our new branch-and-bound technique.

B. Lower Bound

The integer programming mapping offers additional advantages that can be exploited to develop yet more efficient algorithm. By using this mapping, we can determine the minimum size of the solution of the Hitting Set Problem or the minimum number of rows of the solution of Sensor Placement Optimization Problem without solving the problem explicitly. For this purpose, we consider the 1–norm and 2–norm of vectors defined as

$$
M = \begin{pmatrix}
M_1 \\
M_2
\end{pmatrix}
$$

then the Sensor Placement Optimization Problem can be formulated as the following optimization problem:

$$
\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{n} S(x_i) \\
\text{subject to} & \quad Mx \geq 1, \quad x_i = 0 \text{ or } 1.
\end{align*}
$$

Here $S(x_i)$ is the set of sensors associated with the $i$th column of $M$ if $x_i=1$, and it is empty set if $x_i=0$. Before we describe the branch-and-bound method, we show how it is possible to obtain lower and upper bounds for the solution of the problem. As it will be explained later, these bounds are essential for our new branch-and-bound technique.
\[ |S| = \sum_{j=1}^{n} |x_j|, \quad |S'| = \sqrt{\sum_{j=1}^{n} x_j^2}. \]

Let us consider the case of the Sensor Placement Problem defined by (5). For the vector \( \mathbf{1} \) in (5), we then have \( \|\mathbf{1}\|_t = t \) and \( \|\mathbf{1}\|_\infty = \sqrt{t} \), where \( t \) is the number of rows of the matrix \( \bar{M} \). Since the elements of both vectors \( \bar{M} \mathbf{1} \) and \( \mathbf{1} \) are positive, we can then drive the following two inequalities:

\[
\|\bar{M}\|_1 \times \|\mathbf{1}\|_t \geq t \quad \Rightarrow \quad \|\bar{M}\|_t \geq \frac{t}{\|\mathbf{1}\|_1} \|\mathbf{1}\|_t \\
\|\bar{M}\|_\infty \times \|\mathbf{1}\|_{\sqrt{t}} \geq \sqrt{t} \quad \Rightarrow \quad \|\bar{M}\|_{\sqrt{t}} \geq \frac{\sqrt{t}}{\|\mathbf{1}\|_{\sqrt{t}}} \|\mathbf{1}\|_{\sqrt{t}}
\]

(6)

Since \( \mathbf{x} \) is a binary vector, then both norms in (6) give the bound on the size of the solution, that is, the number of nonzero elements of vector \( \mathbf{x} \) which, indeed, corresponds to the number of rows of the solution.

Note that, depending on the structure of the problem, i.e., the 1- and 2-norm of the matrix \( \bar{M} \) and \( t \), a sharper bound can be derived from either of (6).

### C. Upper Bound

Given a signature matrix \( H \), by upper bound we mean a fast and not necessarily an optimal solution for the problem. This means we want to find a submatrix \( G \) of \( H \) by deleting some rows, such that all columns of \( G \) are non-zero and distinct. The problem always has a trivial solution, which is \( G=H \), but we try to find a better solution while we keep this stage computationally as simple as possible. We suggest a greedy approach. First we make sure that the solution has no all-zero column. One easy solution is a hitting set of the rows; i.e., a 0/1 solution for \( H^T \mathbf{x} \geq \mathbf{1} \), where \( H^T \) is the transposed of the matrix \( H \) and \( \mathbf{1} \) is the all-one vector (see [3] for details). This way we get a set of rows of \( H \) that defines a submatrix \( G_0 \) with no all-zero column. We find the set \( S_1 \) of all sensors involved in rows of \( G_0 \) and add all other rows of \( H \) which only depend on sensors in \( S_1 \). The result is a submatrix \( G_1 \).

If all columns of \( G_1 \) are distinct then \( G=G_1 \). Otherwise we choose two identical columns, say columns \( i \) and \( j \). Now consider the set \( R_1 \) of all rows not involved in submatrix \( G_1 \) which have different values at columns \( i \) and \( j \), i.e., they have pattern \((0,1)\) or \((1,0)\) at this columns. Then from the set of rows \( R_1 \) we choose one row which depends on least numbers of sensors not in \( S_2 \). We add the set of sensors of this row to \( S_2 \) to obtain the set \( S_3 \), and let \( G_2 \) be the submatrix of \( H \) consists of all rows of \( H \) only depending on the sensors in \( S_3 \). Then \( G_2 \) is an extension of \( G_1 \) and its \( i \) and \( j \) columns are distinct. If all columns of \( G_2 \) are distinct then this matrix is the solution, otherwise we repeat the above procedure.

For example, consider the signature matrix \( H \) defined in the Table 1. Using the hitting set upper bound algorithm for the rows we find the last two rows as the solution. Thus

\[
G_5 = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}
\]

These two rows depend on the sensors \( S(f) \), \( S(g) \), and \( S(h) \); therefore, \( S_5 = \{ S(f), S(g), S(h) \} \) and by adding all rows of \( H \) depending on these sensors (which are the rows 10–12), we obtain

\[
G_6 = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\
\end{bmatrix}
\]

Now columns 1 and 5 of \( G_6 \) are identical, and a candidate row that has different values at these columns and depends on minimum number of sensors is row 1, which only depends on sensors \( S(x) \). Therefore, \( S_7 = \{ S(x), S(f), S(g), S(h) \} \). Then the submatrix of \( H \) defined by the sensors in \( S_7 \) is

\[
G_7 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\
\end{bmatrix}
\]

The matrix \( G_7 \) still has two identical columns: columns 4 and 7. Then row 4 has different values at these columns and depends only to sensor \( S(t) \). Thus \( S_8 = \{ S(x), S(f), S(t), S(g), S(h) \} \), and the corresponding submatrix is

\[
G_8 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\
\end{bmatrix}
\]

Now all columns of \( G_8 \) are distinct and the solution is the set \( S_8 \).
The above method can easily be extended to the case that a set \( R \) of rows is chosen and the solution should contain this set.

**D. Branch-and-Bound Algorithm**

Before describing our method, we introduce a set of useful notations and functions. We start with an \( n \times m \) signature \( H \). We label the rows of \( H \) by the numbers 1, 2, …, \( n \), and we denote any subset of these rows simply by a subset of \{1, 2, …, \( n \)\}. For example, we denote the set of 1\(^{st} \) and 3\(^{rd} \) rows by \{1,3\}. The branch-and-bound method is based on search on the nodes of a tree. Each node of the search tree has a label of the form

\[(R_{in}, R_{out})\]

where \( R_{in} \) and \( R_{out} \) are disjoint subsets of the rows of the matrix \( H \). The meaning of this partition is that \( R_{in} \) is the set of the rows considered as part of the solution, and \( R_{out} \) is the set of the rows considered not as part of the solution.

Here is the list of the auxiliary functions.

- **Sensor**[\( R \)]: For a given set \( R \) of rows of the signature matrix \( H \), Sensor[\( R \)] returns the set of all sensors the rows in \( R \) depending on. For example, in the case of the signature matrix of Table 1, we have Sensor[\{13,14\}] = \{S(1),S(g),S(h)\}.
- **Closure**[\( R \)]: For a given set \( R \) of rows of the signature matrix \( H \), Closure[\( R \)] returns the set of all rows of \( H \) which depend only on sensors in Sensor[\( R \)]. Note that \( R \) is always a subset of Closure[\( R \)]. For example, in the case of the signature matrix of Table 1, we have Closure[\{13,14\}] = {10,11,12,13,14}.
- **Choice**[\( A \)]: For a 0/1 matrix \( A \), Choice[\( A \)] returns the row of \( A \) with the following property: the difference between the number of 0’s and 1’s of that row is minimal. If there are several rows satisfying this condition, then Choice[\( A \)] returns the first one. For example, in the case of the signature matrix \( H \) of Table 1, we have Choice[\( H \)] = \{10\}.
- **Upper_Bound**[\( T, H, S \)]: For a signature matrix \( H \), corresponding set of sensors \( S \) and a set \( T \) of rows of \( H \), Upper_Bound[\( T, H, S \)] returns an upper bound for the number of rows of the solution of sensor placement problem while we assume \( T \) is part of the solution.
- **Upper_Bound_Set**[\( T, H, S \)]: This function is similar to the previous function, but it returns the set of the rows of \( H \) which realizes the bound Upper_Bound[\( T, H, S \)].
- **Test_Leaf**[\( T_{in}, T_{out}, U, H, S \)]: For a signature matrix \( H \), corresponding set of sensors \( S \), a the label \( (T_{in}, T_{out}) \), and a number \( U \), this function returns True if \( T_{in} \cup T_{out} \) is the set of all rows of \( H \) or the lower bound of the sensor placement problem with the condition that \( T_{in} \) is part of the solution and \( T_{out} \) is not part of the solution is greater than the number \( U \), otherwise the function returns the value False.
- **Split**[\( T_{in}, T_{out}, H, S \)]: For a signature matrix \( H \), corresponding set of sensors \( S \), and the label \( (T_{in}, T_{out}) \), this function returns a pair \( (\lambda_0, \lambda_1) \) of new labels if \( T_{in} \cup T_{out} \) is not the set of all rows of \( H \), otherwise it is undefined. In the case that the function is defined, let \( A \) be the submatrix obtained from \( H \) by removing all rows in the set Closure[\( T_{in} \cup T_{out} \)], and let \( R_0 = \text{Choice}[A] \). Then \( \lambda_0 = (T_{in} \cup \{R_0\}, T_{out}) \), \( \lambda_1 = (T_{in}, T_{out} \cup \{R_0\}) \).

Now we are ready to present our new branch-and-bound algorithm. It is presented in Fig. 2.

![Fig. 2. Sensor placement branch-and-bound algorithm.](image)

7. **DIAGNOSIS AND SIGNATURE MATRIX**

It is possible to formulate the solution of the diagnosis problem in terms of signature matrix (see, e.g., [2]). Briefly, with the values of observed parameters at hand, first we determine which ARRs are not satisfied. These ARRs are in one-to-one correspondence with the
conflict sets. Then we consider the submatrix defined by the rows associated with these ARRs. This is the same matrix \( A \) we defined in Section 2.

8. CONCLUSION

We have presented a new method for solving the optimal sensor placement problem and its relation with diagnosis problem. We showed that both problems could be formulated as Integer Programming optimization problem with linear and non-linear objective functions. The main contribution of this paper is a new mapping of the optimal sensor placement onto the Integer Programming and the development of a branch-and-bound algorithm for its solution. To our knowledge, this is the first systematic (i.e., non exhaustive search) approach for the problem. As we mentioned before, our new branch-and-bound algorithm has achieved significant (over an order of magnitude) speedup over the best known algorithm for solution of the Integer Programming problem corresponding to the diagnosis problem. We are currently benchmarking the branch-and-bound algorithm for the optimal sensor placement.

The method that we discussed here mainly concerns with the basic case of finding minimal size set of sensors among all possible (hypothetical) sensors. But this does not restrict capability of our method to deal with other scenarios for sensor placement. For example, in the case that there is a cost associated with each sensor, the optimization problem is to find the set of minimal cost sensors. Our technique can easily be modified to cover this case as well. For this purpose, we have to modify the subroutine that finds the upper bound such that instead of the size of the set of sensors the total cost of the sensors is considered as the objective function. Also in the case that there is already a fixed set of sensors and we want to add more sensors to the system to enhance its diagnosability, we can simply utilize our method but instead of the hypothetical signature matrix we have to consider its submatrix obtained by deleting the rows corresponding with the set of fixed sensors. In the subsequent paper we will discuss all this matters in detail.

Also, we would like to mention that there is a close relation between the problem of finding conflict sets for diagnosis problem and the problem of constructing signature matrix for sensor placement problem [9]. This is another evidence of close the relation of these two problems.

ACKNOWLEDGMENT

The research described in this paper was performed at the Jet Propulsion Laboratory (JPL), California Institute of Technology, under contract with National Aeronautics and Space Administration (NASA). This work is supported by NASA ESMD Intelligent Systems (IS) Transition Program, under Hybrid Diagnosis/Prognosis Task.

REFERENCES