Optimal test / sensor selection problems formalized as integer programs
Christian Artigues, Olivier Celestin Bassène, Elodie Chanthery, Asma Gasmi, Louise Travé-Massuyès

To cite this version:
Christian Artigues, Olivier Celestin Bassène, Elodie Chanthery, Asma Gasmi, Louise Travé-Massuyès. Optimal test / sensor selection problems formalized as integer programs. 29th International Workshop on Principles of Diagnosis - DX’18, Aug 2018, Varsovie, Poland. hal-01882322

HAL Id: hal-01882322
https://hal.archives-ouvertes.fr/hal-01882322
Submitted on 26 Sep 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Optimal test / sensor selection problems formalized as integer programs

C. Artigues, O. C. Bassène, E. Chanthery, A. Gasmi, and L. Travé-Massuyès

LAAS-CNRS, Université de Toulouse, CNRS, INSA, Toulouse, France

e-mail: artigues, ocbassen, elodie.chanthery, asma.gasmi, louise@laas.fr

Authors are given in alphabetical order.

Abstract

Diagnosis is the reasoning leading to the identification of the cause of a problem. Given a system instrumented with a set of sensors, diagnosis can be performed thanks to diagnosis tests that are designed from the system model. The tests only involve measured variables and can be checked with the measured values. The configuration of tests that pass and tests that do not pass provides a way to isolate the faults. However the fact that all faults are discriminable, i.e. the system is fully diagnosable, depends on the set of tests and hence of the sensors that are placed on the system. The number of tests that can be designed is generally huge and more than sufficient to achieve full or maximal diagnosability. This paper addresses several variants of the problem of selecting the set of tests/sensors so that diagnosability is maximal and a cost criterion is minimized. The variant problems are formalized in the integer programming framework.

1 Introduction

Since the end of the last century, the industrial world has continued to evolve and with this evolution, the systems and industrial processes are more complex and efficient. The growing complexity of industrial systems that have become increasingly demanding in terms of reliability, backup contact performance and availability has given rise to increasing interest of diagnosis. As a result, companies are trying to develop and implement diagnosis and maintenance systems to ensure the reliability and availability of installations. The effectiveness of the diagnosis system depends on the relevance of the information received on the system to be diagnosed through sensors. The efficiency of sensors can be evaluated with the level of diagnosability that can achieved, i.e. which faults are discriminable (or isolable) and which are not. In this paper, we consider that the system is composed by a set of interconnected components and that faults are associated to components. Diagnosis is generally achieved thanks to a set of diagnosis tests that are checked against the measures. Diagnosability refers to two properties: detectability, that means that a fault can be detected by at least one test; and isolability, that means that each fault has its own way to answer the tests.

Which diagnosis tests and how many of them are achievable depends on the sensors and their locations. In practice, diagnosis systems are forced by the cost of sensors and by the computational cost related to the number of diagnosis tests to be checked that can be prohibitive for on-line applications. Hence, optimizing the cost of sensors and/or tests while still guarantying a good level of diagnosability is essential in many application domains.

To achieve this, optimization methods can be put in place. In this paper, we consider five increasingly complex optimization problems. Problem I simply addresses the selection of a minimal number of tests guarantying full diagnosability. By full diagnosability, we mean that any fault in any component can be distinguished from any fault in another component. Problem II addresses the selection of a set of minimal cost sensors, selected among a set of possible sensors, that guaranty full diagnosability. This latter problem is obviously more difficult because sensors do not influence diagnosability directly but through the tests that can be actually implemented. Problem III considers that full diagnosability may not be achievable with the possible sensors but minimal cost selection must achieve maximal diagnosability. Problem IV maximizes components diagnosability when sensors themselves can undergo faults. Finally, Problem V requires that sensor faults are also diagnosable in addition to component faults. It maximizes diagnosability in this context. The two later problems hence provide solutions to robust optimal sensor selection problems. The five problems are formalized in an integer programming framework.

The paper is structured as follows. Section 2 provides an overview of the state of the art and positions the contribution. In Section 3, the notations and definitions used in the paper are presented. Section 4 formalizes the five optimization problems in an integer programming framework. Section 5 proposes an evaluation on an academic example, then on two real examples, a LEO satellite case study and a water desalinator. Finally, section 6 discusses the results and concludes the paper.

2 Related work

The objective of this paper is to propose a formalization of five increasingly complex optimization problems related to diagnosability. The optimization targets the cost of sensors and/or the number of diagnosis tests.
The optimal test selection problem, as known in the literature, aims at minimizing the cost of the tests while satisfying some isolability constraints. This problem is obviously related to the sensor placement problem, which searches the minimum cost sensor configuration that satisfies a given set of fault diagnosis specifications. However, sensor selection is a more difficult problem because sensors do not influence diagnosability directly but through the tests that can be actually implemented.

In the literature, optimal test selection is often associated to the problem of test prioritization that corresponds to choosing the next best test or measurement to disambiguate a faulty situation. In practice, this is an integral part of the troubleshooting task. This domain has received a lot of attention [13; 14; 4].

Solutions have been proposed through heuristic optimization techniques [12; 15], focusing on diagnosis ambiguity reduction as the main optimization criterion, which leads to adopt heuristics based on information theory [8]. The methods known as diagnosis Test Prioritization Techniques are based on the Information Gain heuristic. They maximize the diagnosis information gain per test and increase the rate at which diagnosis quality improves [7] but they are often limited by their complexity. Nevertheless, we can mention the gReedy diagnosis Prioritization by ambiguTiY Reduction (RAPTOR) method [6] as a instance that achieves to restrain this issue.

Some authors address the sensor selection problem [18]. Several papers adopt the structural analysis framework [2; 5; 10], which is in line with this paper. In particular, [9] uses a structural model of the system to compute all minimal sensor sets that make faults isolable from each others. [19] uses an alternative structural model decomposition, based on gathering equations that can not be isolable. [11] presents an approach for diagnosability analysis and sensor placement based on genetic algorithms.

The problem of optimal test selection can also be formulated as a combinatorial optimization problem. Such formulation has already been proposed using binary integer programming [11; 17; 16], which is in line with the approach adopted in this paper. However, this approach has not been applied to maximize the robust optimal selection problem, when sensors themselves can be faulty and sensor faults must be diagnosable. In other words, Problems I and II find solutions in the literature. For these problems we contribute with a new compact binary integer programming formulation. Problems III, IV, and V solve variants of the diagnosability maximization problem.

### 3 Notations and definitions

Let us consider a system $\Sigma$ composed of a set $C$ of $n_c$ components and instrumented with a set of sensors $S$. Assume that a set $T$ of $N$ binary diagnosis tests is available for this system. The output of a test is either 0 when the test passes or 1 when it does not pass.

Every test $T_i$, $i = 1, \ldots, N$, covers a subset of components of $2^S$ that form the component test support denoted by $\text{Supp}_C(T_i)$ and it is supported by a subset of sensors of $2^S$ that form the sensor test support denoted by $\text{Supps}(T_i)$. This means that:

- all the sensors of $\text{Supps}(T_i)$ must be installed on the system for the test $T_i$ to be available,
- when the test $T_i$ does not pass, one of the components of $\text{Supp}_C(T_i)$ or $\text{Supps}(T_i)$ if sensors are assumed to undergo faults, is faulty.

The matrix that crosses tests on lines and components and sensors on columns is called the covering matrix and is denoted by $M$. $M$ is the concatenation of two matrices:
- $M_C$ is the submatrix that covers the set of components $C$,
- $M_S$ is the submatrix that covers the set of sensors.

Let us denote by $M_{(i,j)}$, $i = 1, \ldots, N$, $j = 1, \ldots, n_C + n_S$, the entry of $M$ at the $i$th line and $j$th column. The entries of $M_C$ and $M_S$ are defined similarly as $M_{(i,j)}$:
- $i = 1, \ldots, N$, $j = 1, \ldots, n_C$, and $M_{(i,j)}$, $i = 1, \ldots, N$, $j = 1, \ldots, n_S$, respectively. $c_i$ denotes the column corresponding to component $C_i$ and $s_k$ denotes the column corresponding to sensor $S_k$.

In the matrix $M$ an entry at "1" on line $i$ means that the corresponding component / sensor belongs to $\text{Supp}_C(T_i) / \text{Supps}(T_i)$. This is illustrated in Table 1.

<table>
<thead>
<tr>
<th>$C_1$</th>
<th>$C_2$</th>
<th>$\ldots$</th>
<th>$C_{n_C}$</th>
<th>$S_1$</th>
<th>$\ldots$</th>
<th>$S_{n_S}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_1$</td>
<td>0</td>
<td>1</td>
<td>$\ldots$</td>
<td>1</td>
<td>$\ldots$</td>
<td>1</td>
</tr>
<tr>
<td>$T_2$</td>
<td>1</td>
<td>0</td>
<td>$\ldots$</td>
<td>0</td>
<td>$\ldots$</td>
<td>0</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\ddots$</td>
<td>$\vdots$</td>
<td>$\ddots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>$T_N$</td>
<td>1</td>
<td>0</td>
<td>$\ldots$</td>
<td>0</td>
<td>$\ldots$</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 1: Covering Matrix.

Selecting a set of tests among the $N$ available tests and a set of sensors among the $n_S$ available sensors comes back to choosing a submatrix $M^*$ composed of a subset of lines and columns of $M$. The corresponding submatrices of $M_C$ and $M_S$ are denoted by $M^*_C$ and $M^*_S$, respectively.

**Definition 3.1** (Instrumented test). A test is said to be instrumented for a submatrix $M^*$ if all the sensors required for the test are available in $M^*$ and free of faults. $n^*_C$ is the number of instrumented tests for the submatrix $M^*$.

**Definition 3.2** (Detectability). The fault $f_j$ of component $C_j$ is said to be detectable if the column $j$ of $M_C$ has at least one "1".

**Definition 3.3** (Isolability). A component $C_j$ is said to be isolable from a component $C_i$ if the columns $i$ and $j$ of $M_C$ are different.

Let us focus on component faults and leave aside sensor faults for now.

**Definition 3.4** (Admissible matrix $M^*$). A matrix $M^*$ is said to be admissible if the set of instrumented tests for $M^*$ provides the isolability for all pairs of components.

**Property 3.5.** $M^*$ is admissible if and only if $M^*_C$ has $n^*_C$ different columns and all tests required for this are instrumented.
Since the entries of the covering matrix $M$ are binary, this property can be formalized by:

$$\sum_{k=1}^{N} |M^c(k, i) - M^c(k, j)| \geq 1, \quad \forall (i, j) \in \{1, ..., n_C\}, i \neq j$$ (1)

### Example
Let us consider the covering matrix given in Table 2 and assume that the three sensors are available, hence all the tests $T_1$ to $T_6$ are instrumented.

<table>
<thead>
<tr>
<th>$T_1$</th>
<th>$C_1$</th>
<th>$C_2$</th>
<th>$C_3$</th>
<th>$C_4$</th>
<th>$C_5$</th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</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>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 2: Full diagnosability covering matrix

We can notice that all the columns referring to components are different, which means that all components are isolable. In other words, the available tests guarantee full diagnosability to the system. Now assume that sensor $S_3$ is not selected, then columns $C_1$ and $C_4$ are identical, hence components $C_1$ and $C_4$ are not isolable anymore.

Our goal is to find a submatrix $M^*$ whose structure guarantees full or maximal diagnosability to the system while minimizing some cost criterion.

## 4 Problem Formalization

There are several variants to formulate an optimal test/sensor selection problem. We are interested in investigating integer programming formulations. In this section, we hence provide the mathematical formulation of five variant problems that can be solved in the integer programming framework. We consider the following hypotheses:

- Problems I and II assume that full diagnosability is achievable and that sensors are not faulty.
- Problems III, IV, and V assume that full diagnosability may not be achievable. In addition, Problems IV and V relax the assumption that sensors cannot be faulty, targeting a robust solution with respect to components only and with respect to components and sensors, respectively.

In all the problems, we do assume that all the tests are instrumented in $M$.

### 4.1 Integer Programming

In this section, we briefly introduce Integer Programming (IP) that we use to solve all different formulations of our problem.

An integer programming problem is a mathematical optimization or feasibility program in which some or all of the variables are restricted to be integers. In many settings the term refers to Integer Linear Programming (ILP), in which the objective function and the constraints (other than the integer constraints) are linear. However, we have the special case of 0-1 integer linear programming, which is an integer linear program in which unknowns are binary.

### 4.2 Problem I: Optimal test selection with fault free sensors

In this section, the goal is to minimize the number of tests that guarantee full component diagnosability under the assumption that sensors do not fail. In other words, the solution must provide an admissible matrix $M^*$.

Let us define the following variables:

- $z_i$ is a boolean variable associated to each test $i \in T$ such that:
  $$z_i = \begin{cases} 1 & \text{if the test } i \text{ is chosen} \\ 0 & \text{otherwise} \end{cases}$$

- $f_{ikl}$ is a parameter defined as follows:
  $$f_{ikl} = \begin{cases} 1 & \text{if the fault of component } k \\ 0 & \text{is isolable from the fault of component } l \text{ with the test } i \text{ otherwise} \end{cases}$$

**Problem I** can be formalized as follows:

$$\min \sum_{i \in T} z_i$$ (2)

Subject to:

$$\sum_{i \in T} f_{ikl} z_i \geq 1, \quad \text{for } (k, l) \in C^2 \quad \text{and} \quad k < l$$ (3)

$$z_i \in \{0, 1\} \quad \forall i \in T$$ (4)

Equation (3) verifies that components are 2 per 2 isolable.

Problem I amounts to the set covering problem (hitting sets), a well-known NP-hard problem.

### 4.3 Problem II: Optimal sensor selection with fault free sensors

In this section, we also consider that sensors do not fail. The goal is now to minimize the cost of the used sensors, i.e., the sensors that are required to obtain an admissible matrix $M^*$.

Besides the already defined parameters and variables, we introduce new items as follows:

- $\alpha_{ij}$ is a boolean parameter such that:
  $$\alpha_{ij} = \begin{cases} 1 & \text{if the test } i \text{ uses the sensor } j \in S \\ 0 & \text{otherwise} \end{cases}$$

- $p_j$ is the cost of sensor $j$.

**Problem II** can be formalized as follows:

$$\min \sum_{j \in S} p_j y_j$$ (5)

Subject to:

$$y_j \geq z_i \quad \text{for } i \in T \quad \text{and} \quad j \in S \quad \text{s.t} \quad \alpha_{ij} = 1$$ (6)
\[ \sum_{i \in T} f_{kl} z_i \geq 1, \text{ for } (k, l) \in C^2 \text{ and } k < l \]  
(7)

\[ y_j \geq 0, \quad \forall j \in S \]  
(8)

\[ z_i \in \{0, 1\}, \quad \forall i \in T \]  
(9)

Objective (5) represents the sum of the costs of selected sensors. Constraints (6) impose to select all the sensors that instrument a selected test. Constraints (7) ensure isolability (covering constraint).

Let us notice that, for efficiency purposes\(^1\), the \(y_j\)'s are coded as continuous variables in the program. In practice, the r.h.s of inequality (6) is binary and, when equal to 1, it forces the \(y_j\)'s to be equal to 1 and when equal to 0, the \(y_j\)'s are forced to 0 by the minimization objective (5).

Note that Problem II is similar to the maximal covering location problem [3], which is also NP-hard.

4.4 Problem III: Maximization of isolability with fault free sensors

In this section, we assume the general case in which there may not be a solution to ensure full isolability. In this case, the solution provided by the solver must ensure maximal isolability. We still have the concern of minimizing the sensor cost. Thus, the problem becomes a bi-objective optimization problem.

Problem III can be formalized as follows:

\[ \max(\sum_{k,l} e_{kl} - \alpha \sum_{j \in S} p_j y_j), \forall (k,l) \in C^2 \]  
(10)

Subject to:

\[ y_j \geq z_i \quad \text{for } i \in T \text{ and } j \in S \text{ s.t } \alpha_{ij} = 1 \]  
(11)

\[ \sum_{i \in T} f_{kl} z_i \geq e_{kl}, \quad (k,l) \in C^2, \quad k < l \]  
(12)

\[ y_j \geq 0, \quad \forall j \in S \]  
(13)

\[ 0 \leq e_{kl} \leq 1, \quad l \in S, \quad \forall k \in C \]  
(14)

\[ z_i \in \{0, 1\}, \quad \forall i \in T \]  
(15)

In this formulation, variables \(e_{kl}\) are used to obtain partial isolability: \(e_{kl} = 1\) means that the fault of component \(k\) is isolable from the fault of component \(l\). Their sum is maximized by the objective function and variables \(e_{kl}\) are introduced as the r.h.s of constraints (12), which assures that the required tests can be selected. Let us notice that, for efficiency purposes\(^1\), the \(e_{kl}\)'s are coded as continuous variables in the program. The explanation is the same as in Problem II for the \(y_j\)'s. For the \(e_{kl}\)'s, it is the same but for a maximization criterion. The l.h.s of inequality (12) is binary: when equal to 0, it forces \(e_{kl}\) to be equal to 0, and when equal to 1, \(e_{kl}\) is forced to 1 by the maximization criterion (10). Note also that the isolability problem is symmetric, thus \(e_{kl} = e_{lk}\).

In the objective function (10), \(\alpha\) is a parameter that weights the two considered criteria and allows us to set the relative importance of the isolability criterion and the cost criterion. Note that in the objective theory this method does not allow one to obtain the full Pareto front as the so-called non supported solutions cannot be found by the weighted sum objective. However the weighted sum method is largely used in practice.

4.5 Problem IV: Maximization of robust isolability w.r.t. component faults

In Problem IV, we consider another variant of the problem where sensors may fail. We want however the output solution to achieve maximal isolability for component faults with minimum cost even when a single sensor fails.

We add a new element to our integer linear program:

- \(T_s\) is a parameter that denotes the set of tests instrumented by sensor \(s\).

The optimization Problem IV becomes:

\[ \max(\sum_{k,l} e_{kl} - \alpha \sum_{j \in S} p_j y_j), \quad k \in C, \quad l \in C \]  
(16)

Subject to:

\[ y_j \geq z_i \quad \text{for } i \in T, \quad j \in S \text{ s.t } \alpha_{ij} = 1 \]  
(17)

\[ \sum_{i \in T \setminus T_s} f_{kl} z_i \geq e_{kl}, \quad (k,l) \in C^2, \quad s \in S, \quad k < l \]  
(18)

\[ y_j \geq 0, \quad \forall j \in S \]  
(19)

\[ 0 \leq e_{kl} \leq 1, \quad \forall l \in S, \quad \forall k \in C \]  
(20)

\[ z_i \in \{0, 1\}, \quad i \in T \]  
(21)

In Problem IV, a covering constraint (18) has been introduced for each sensor, guaranteeing that the isolability set by \(e_{kl}\) is reached eventhough the sensor fails.

4.6 Problem V: Maximization of robust isolability w.r.t. component and sensor fault

In Problem V, we yet consider another variant of the problem. Like in Problem IV, one sensor may fail. However, we want the solution to achieve maximal isolability for component and sensor faults with minimum cost. Consequently, the index \(l\) now refers to components \(\text{and sensors} (l \in (C \cup S))\) in constraints (24) and not only to components like in Problem IV.

Problem V can be formalized as follows:

\[ \max(\sum_{k,l} e_{kl} - \alpha \sum_{j \in S} p_j y_j), \quad k \in C, \quad l \in (C \cup S) \]  
(22)

Subject to:

\[ y_j \geq z_i \quad \text{for } i \in T, \quad j \in S \text{ s.t } \alpha_{ij} = 1 \]  
(23)

\[ \sum_{i \in T \setminus T_s} f_{kl} z_i \geq e_{kl}, \quad k \in C, \quad l \in (C \cup S), \quad s \in S, \quad k < l \]  
(24)

\[ y_j \geq 0, \quad \forall j \in S \]  
(25)

\[ 0 \leq e_{kl} \leq 1, \quad \forall l \in S, \quad \forall k \in C \]  
(26)

\[ z_i \in \{0, 1\}, \quad i \in T \]  
(27)

\(^1\)Note that in recent solvers, it may not be more efficient.
5 Implementation and validation

In the implementation of our mathematically formalized problems, we used the Gurobi Optimizer with the Python API. In what follows, we provide illustrative solutions to an academic example and the solutions to two industrial problems.

5.1 Academic example

Let us consider a simple example where the number of components $n_C$, the number of sensors $n_S$, and the number of tests $n_T$ are all fixed at 9.

The covering matrix we consider is given in Table 3 (w.r.t. components) and Table 4 (w.r.t. sensors).

<table>
<thead>
<tr>
<th>$T_1$</th>
<th>$T_2$</th>
<th>$T_3$</th>
<th>$T_4$</th>
<th>$T_5$</th>
<th>$T_6$</th>
<th>$T_7$</th>
<th>$T_8$</th>
<th>$T_9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 3: Covering matrix w.r.t. components $M_C$ for the academic example

<table>
<thead>
<tr>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
<th>$S_4$</th>
<th>$S_5$</th>
<th>$S_6$</th>
<th>$S_7$</th>
<th>$S_8$</th>
<th>$S_9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>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>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 4: Covering matrix w.r.t. sensors $M_S$ for the academic example

Results for Problem I and Problem II

Problem I (Optimal test selection with fault free sensors) and Problem 2 (Optimal sensor selection with fault free sensors) were implemented in the Gurobi Optimizer. Their execution did provide no solution. This means that full diagnosability is not possible given the covering matrices $M_C$ and $M_S$. This could be anticipated because in $M_C$, we can notice that $C_2 = C_6$ and $C_5 = C_7$. These components are hence not isolable.

Results for the Problem III

After implementing Problem 3 (Maximization of isolability with fault free sensors) with the Gurobi Optimizer, the solution provided by its execution consisted in a subset of 7 sensors $\{S_1, S_2, S_3, S_4, S_5, S_6, S_7\}$. These sensors can indeed be used to instrument 5 tests $T_2, T_5, T_6, T_7, T_8$ that maximize diagnosability. The corresponding solution matrix $M^*$ is given in Table 5 where we can check which component columns are different leading to 34 pairs of isolable components.

<table>
<thead>
<tr>
<th>$T_1$</th>
<th>$T_2$</th>
<th>$T_3$</th>
<th>$T_4$</th>
<th>$T_5$</th>
<th>$T_6$</th>
<th>$T_7$</th>
<th>$T_8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5: Solution matrix $M^*$ for Problem III applied to the academic example

Results for the Problem IV

After implementing Problem IV (Maximization of robust isolability w.r.t. component faults) with the Gurobi Optimizer, the solution provided by its execution consisted in a subset of 7 sensors $\{S_1, S_2, S_3, S_4, S_5, S_6, S_7\}$. These sensors can indeed be used to instrument 6 tests $T_2, T_4, T_5, T_6, T_8, T_9$, achieving maximal robust diagnosability w.r.t. component faults. The corresponding matrix $M^*$ is given in Table 6 where we can check that all component columns are different leading to 25 pairs of isolable components but $c_6$ and $c_7$ even when one of the sensors is faulty and cannot be used to instrument any test.

<table>
<thead>
<tr>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
<th>$S_4$</th>
<th>$S_5$</th>
<th>$S_6$</th>
<th>$S_7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 6: Solution matrix $M^*$ for Problem IV applied to the academic example

Interestingly, it can be noticed that the number of required sensors is the same as for Problem III. Nevertheless, the sensors are different and they can be used to instrument 6 tests instead of 5, achieving this way robust diagnosability.

Results for the Problem V

After implementing Problem V (Maximization of robust isolability w.r.t. component faults and sensor faults) with the Gurobi Optimizer, the solution provided by its execution consisted in all the sensors $\{S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8\}$. However not all the tests are necessary. 8 instrumented tests $T_1, T_2, T_3, T_5, T_6, T_7, T_8, T_9$ are enough to achieve maximal robust diagnosability w.r.t. component faults and sensor faults. The corresponding solution matrices $M^*_C$ and $M^*_S$ are given in Table 7 and Table 8 where we can check which component and sensor columns are different even when one of the sensors is faulty and cannot be used to instrument any test. We obtain 96 pairs of isolable components/sensors.

5.2 The LEO satellite benchmark

A LEO satellite (Low Earth Orbit) [20], also known as the OTB orbit, is a satellite in the zone up to 2,000 kilome-
Satellite is composed of rate gyros for each of the three axes, and vector sensors which are used to periodically clear the accumulated attitude drift error from the rate gyroscopes. As a result, we consider faults on rate and vector sensors as well as faults on the reaction wheels. This sums up to 9 faults $f_1$ to $f_9$ for the ADCS. The covering matrix that we considered has faults as columns instead of components. An entry at “1” on line $i$ means that the test $T_i$ does not pass when the corresponding fault is present.

### Results for Problem I

An analysis of the ADCS and its model has provided 2448 tests for this system [20]. In this case study, we have considered Problem I and implemented the corresponding program in the Gurobi Optimizer. We found that only 4 of the 2448 tests are necessary to achieve full diagnosability and discriminate all the faults. These tests are: $T_{315}$, $T_{346}$, $T_{503}$ and $T_{2100}$.

The solution matrix $M^*$ shows that the tests found ensure the isolability of all faults and their number is minimal. Optimality can be confirmed from the theoretical minimal number of tests. It is obtained by considering that a set of tests exists such that every test iteratively partitions the fault set in two sets of non-discriminable faults. So if we have $n_c$ components, then $log_2(n_c)$ is the greatest lower bound of the theoretical minimal number of tests. In this case study, $n_c = 9$ and the theoretical minimal number of tests is hence 4, which is in accordance with the result provided by Problem I.

### 5.3 The desalinator benchmark

#### System description

The desalination system [20] is a system that removes salt from the water. It uses several chemical, electrical and even thermal procedures (cf. Figure 2). Given the complexity of this system, there is a need for properly instrumenting the system and providing relevant information that helps diagnosing the chemical desalination process.

If we take the desalinator case study presented in [20], it has 7 possible faults:

- $f_1$ is the fault of the conductivity sensor,
- $f_2$ is the fault of the flow sensor,
- $f_3$ is a high pressure pump fault,
- $f_4$ is the fault of the temperature sensor,
- $f_5$ is the fault of scaling of the membrane,
- $f_6$ is the defect of the membrane,
- $f_7$ is the failure of the pump acid pump pH.

The desalination system [20] is a system that removes salt from the water. It uses several chemical, electrical and even thermal procedures (cf. Figure 2). Given the complexity of this system, there is a need for properly instrumenting the system and providing relevant information that helps diagnosing the chemical desalination process.

If we take the desalinator case study presented in [20], it has 7 possible faults:

- $f_1$ is the fault of the conductivity sensor,
- $f_2$ is the fault of the flow sensor,
- $f_3$ is a high pressure pump fault,
- $f_4$ is the fault of the temperature sensor,
- $f_5$ is the fault of scaling of the membrane,
- $f_6$ is the defect of the membrane,
- $f_7$ is the failure of the pump acid pump pH.

The desalination system [20] is a system that removes salt from the water. It uses several chemical, electrical and even thermal procedures (cf. Figure 2). Given the complexity of this system, there is a need for properly instrumenting the system and providing relevant information that helps diagnosing the chemical desalination process.

If we take the desalinator case study presented in [20], it has 7 possible faults:

- $f_1$ is the fault of the conductivity sensor,
- $f_2$ is the fault of the flow sensor,
- $f_3$ is a high pressure pump fault,
- $f_4$ is the fault of the temperature sensor,
- $f_5$ is the fault of scaling of the membrane,
- $f_6$ is the defect of the membrane,
- $f_7$ is the failure of the pump acid pump pH.

The desalination system [20] is a system that removes salt from the water. It uses several chemical, electrical and even thermal procedures (cf. Figure 2). Given the complexity of this system, there is a need for properly instrumenting the system and providing relevant information that helps diagnosing the chemical desalination process.

If we take the desalinator case study presented in [20], it has 7 possible faults:

- $f_1$ is the fault of the conductivity sensor,
- $f_2$ is the fault of the flow sensor,
- $f_3$ is a high pressure pump fault,
- $f_4$ is the fault of the temperature sensor,
- $f_5$ is the fault of scaling of the membrane,
- $f_6$ is the defect of the membrane,
- $f_7$ is the failure of the pump acid pump pH.

The desalination system [20] is a system that removes salt from the water. It uses several chemical, electrical and even thermal procedures (cf. Figure 2). Given the complexity of this system, there is a need for properly instrumenting the system and providing relevant information that helps diagnosing the chemical desalination process.

If we take the desalinator case study presented in [20], it has 7 possible faults:

- $f_1$ is the fault of the conductivity sensor,
- $f_2$ is the fault of the flow sensor,
- $f_3$ is a high pressure pump fault,
- $f_4$ is the fault of the temperature sensor,
- $f_5$ is the fault of scaling of the membrane,
- $f_6$ is the defect of the membrane,
- $f_7$ is the failure of the pump acid pump pH.

The desalination system [20] is a system that removes salt from the water. It uses several chemical, electrical and even thermal procedures (cf. Figure 2). Given the complexity of this system, there is a need for properly instrumenting the system and providing relevant information that helps diagnosing the chemical desalination process.

If we take the desalinator case study presented in [20], it has 7 possible faults:

- $f_1$ is the fault of the conductivity sensor,
- $f_2$ is the fault of the flow sensor,
- $f_3$ is a high pressure pump fault,
- $f_4$ is the fault of the temperature sensor,
- $f_5$ is the fault of scaling of the membrane,
- $f_6$ is the defect of the membrane,
- $f_7$ is the failure of the pump acid pump pH.

The desalination system [20] is a system that removes salt from the water. It uses several chemical, electrical and even thermal procedures (cf. Figure 2). Given the complexity of this system, there is a need for properly instrumenting the system and providing relevant information that helps diagnosing the chemical desalination process.

If we take the desalinator case study presented in [20], it has 7 possible faults:

- $f_1$ is the fault of the conductivity sensor,
- $f_2$ is the fault of the flow sensor,
- $f_3$ is a high pressure pump fault,
- $f_4$ is the fault of the temperature sensor,
- $f_5$ is the fault of scaling of the membrane,
- $f_6$ is the defect of the membrane,
- $f_7$ is the failure of the pump acid pump pH.

The desalination system [20] is a system that removes salt from the water. It uses several chemical, electrical and even thermal procedures (cf. Figure 2). Given the complexity of this system, there is a need for properly instrumenting the system and providing relevant information that helps diagnosing the chemical desalination process.

If we take the desalinator case study presented in [20], it has 7 possible faults:

- $f_1$ is the fault of the conductivity sensor,
- $f_2$ is the fault of the flow sensor,
- $f_3$ is a high pressure pump fault,
- $f_4$ is the fault of the temperature sensor,
are sensors eventhough one sensor may be faulty. These sensors guaranty maximal diagnosability w.r.t. components and sen-
sors eventhough one sensor may be faulty. These sensors are $S_1, S_2$ and $S_5$. Interestingly, 283 tests can be instrumented with these sensors.

Results for Problem III

The solution of Problem III provided by Gurobi is the same as the solution of Problem II, which is obviously correct. Indeed, when full diagnosability is achievable, maximal diagnosability is just full diagnosability. We obtain 28 pairs of isolable components.

Results for Problem IV

The solution of Problem IV provided by Gurobi indicates that 3 sensors are necessary to guaranty maximal diagnosability w.r.t. components eventhough one sensor may be faulty. These sensors are $S_1, S_2$ and $S_4$. 333 tests can then be instrumented and we obtain 59 pairs of isolable components/sensors independently of the failure of one sensor.

6 Conclusion

This paper shows the potential of integer linear programming for modeling five increasingly complex optimization problems related to diagnosability. The optimization targets the cost/number of sensors and/or the number of diagnosis tests under different assumptions. Although Problems I and II find solutions in the litterature, we have contributed with a new compact binary integer programming formulation. Problems III, IV, and V solve variants of the diagnosability maximization problem. Maximizing diagnosability when full diagnosability is not achievable is an interesting problem addressed in this paper. The five problems have been programmed using the Python language accompanied by the Gurobi solver library in order to find an optimal solution to the problems.

Our programs have been validated with the application to two real case studies, a LEO satellite and a water desalinator benchmark [20].

Future work will consider additionnal constraints refering to the practical implementation of the tests. On the other hand, solving the five problems in a decentralized or distributed framework will also be considered.

References


