Efficient Optimal Sensor Placement for Structural Model Based Diagnosis

Albert Rosich [1], Abed Alrahim Yassine [2], Stéphane Ploix [2]

[1] *SAC group, Universitat Politècnica de Catalunya, Rambla de Sant Nebridi, 10, 08222 Terrassa, Spain*
albert.rosich@upc.edu

[2] *G-SCOP lab, INPG, CNRS UMR 5272, BP 46, F-38402 Saint Martin d'Hères Cedex, France*
abed.alrahim.yassine@g-scop.inpg.fr
stephane.ploix@g-scop.inpg.fr

Abstract

This work aims to study which sensors are required to be installed in a process in order to improve certain fault diagnosis specifications. Especially, the present method is based on structural models, thus system models that involve a wide variety of equations (e.g. linear, non-linear algebraic, dynamics) can be easy handled. The use of structural models permits to define the diagnosis properties from the Dulmage-Mendelsohn decomposition, avoiding in this way the computation of any minimal redundant subsystem. Furthermore, in the present paper, the cost of the sensor configuration is considered, therefore the proposed method attempts to find not all the possible solution but the optimal one. The optimal search is efficiently performed by developing an algorithm based on heuristic rules which, in general, allow to significantly reduce the search. Finally, the presented method is applied to an electronic circuit as an illustrative example and some special conclusions are highlighted.

1 INTRODUCTION

Designing an efficient diagnosis system may not be done after the system has been designed, but it can be done from the system design. Indeed, it is known that the performance of a diagnostic system strongly depends on the number and on the location of actuators and sensors. Therefore, designing a system that has to be diagnosed not only require relevant fault diagnosis procedures but also efficient sensor placement algorithms.

First works on sensor placement for fault diagnosis can be found, for instance, in Madron and Veverka [1992], where a sensor placement method which deals with linear systems is proposed. This approach makes use of the Gauss-Jordan elimination method to find a minimum set of variables to be measured. Another method for sensor placement was proposed in Maquin *et al.* [1997]. This method aims at guaranteeing the detectability and isolability of sensor failures. The proposed method is based on the concept of redundancy degree in variables and on the structural analysis of the system model.

Recently, some new works on this topic have been published. This is the case of, for example, Travé-Massuyès *et al.* [2006] and Rosich *et al.* [2007] where methods that requires the design of Analytical Redundancy Relations (ARR) [Blanke *et al.*, 2006] are developed. In Commault *et al.* [2006], a method based on directed graph and a class of separators is presented. In Krysander and Frisk [2008] an efficient method based on a partial order on the well-determined subsets from a structural model has been proposed. Another sensor placement method without designing ARRs has be presented in Yassine *et al.* [2008a] and Yassine *et al.* [2008b]. All these works have in common that the model is described by a graph, which means that only the structure of the model is regarded.

The cited works present different approaches where diagnosis properties are characterised to be easily handled when sensor placement problem is formulated. However, none of them aims to efficiently find the solution when the number of sensors to be considered grows.

On the other hand, there are some works that formulate the sensor placement problem as a mixed integer linear programing since the nature of the problem is combinatorial. In Rosich *et al.* [2009] and Fijany and Vatan [2006], two similar methods are presented where fault detectability and isolability are defined by means of linear inequality constraints. Although the sensor search of these works is quite efficient, the main drawback is that all the ARRs need to be previously generated with all the possible sensors installed. This yields serious computing restrictions when the set of possible sensors is large.

The present paper introduces a sensor placement method which first defines diagnosis spec-

ifications based on structural model properties. Specifically, fault detectability, discriminability and diagnosability are characterised by means of the Dulmage-Mendelsohn decomposition [Dulmage and Mendelsohn, 1959], [Murota, 2000]. The second aim is to develop an efficient algorithm in order to find the optimal sensor set such that diagnosis specifications are fulfilled.

The structure of this paper is the following. Section 2 motivates the problem, as well as some guidelines on sensor modelling are given and the fault diagnosis capabilities of the structural models are introduced. Section 3 addresses how to compute such diagnosis capabilities. Next, Section 4 is devoted to testing whether diagnosis specifications are fulfilled. In Section 5, the optimal sensor search is motivated and introduced. Finally, in Section 6 an application example is presented and in Section 7 conclusions are drawn.

2 SENSOR PLACEMENT PROBLEM FOR DIAGNOSIS

2.1 Sensor placement motivation

The solution of a diagnostic problem is generally decomposed into two consecutive steps. The first step is known as fault detection also called conflict or symptom generation, and the second step is known as fault isolation or diagnostic analysis. In structural model based diagnosis, the fault detectability and isolability capabilities can be determined, from a theoretical point of view, by means of the minimal testable subsets of constraints.

The minimal testable subsets can be obtained from constraint combinations using possible conflict generation [Pulido and Alonso, 2002], bipartite graph [Blanke *et al.*, 2006], Dulmage-Mendelsohn decomposition [Krysander *et al.*, 2008] or elimination rules [Ploix *et al.*, 2008]. An inconsistency in a minimal testable subset means that, at least, one of the behavior modes associated to this constraint subset is not actual. Therefore, tracing which constraints belongs to a minimal testable subset makes possible to identify the detectable faults and the isolable faults. In Travé-Massuyès *et al.* [2006], it is shown that the family of minimal testable subsets depends on the observations from the available sensors installed in the system, thus it turns out that diagnostic performance also depends on these sensors.

Additional sensors lead to additional minimal testable subsets. Given some diagnosis specifications, one possible approach is to test on the set of testable subsets whether the performance of the diagnostic system satisfies the requested specifications. When they are not satisfied, the minimal testable subsets are modified by adding more sensors and the process is repeated once again until specifications are reached. This strategy is carried out, for example, in Travé-Massuyès *et al.* [2006] and Rosich *et al.* [2007]. However, this approach requires lots of computations due to the generation of testable subsets is time demanding.

A different approach for sensor placement is proposed in this paper which does not require the computation of minimal testable subsets. The present method solves the sensor placement problem by computing the diagnosis capabilities directly from the model structure. Hence, the minimal testable subset computation burden is avoided.

Given a structural model, a set of candidate sensors to be installed in the system and a cost associated to each sensor, the problem to be solved in the present paper can be summarized as finding the optimal sensor configuration such that the requested diagnosis specifications are guaranteed by the structural model properties.

2.2 Behavioral sensor modelling

The behavior model of a system can be defined as a set of relations which constrains the domain of a set of system variables, V. This relations are then modelled by a constraint set, K, where each constraint, $k \in K$, in the model involves a subset of variables in V. In a component-oriented model, these constraints are associated to the components of a system, thus knowing the behaviour model of a component is straightforward by just selecting those constrains associated with the component.

In the present work we will consider sensor as a components of the system, which means that each sensor has to be associated with some model constraint. Let $obs(v)$ be the observed value acquired by a sensor which is measuring the variable $v \in V$, then the corresponding model of the sensor is

$$v - obs(v) = 0 \tag{1}$$

Equation (1) is called *sensor constraint* and represents the behavior model of the sensor component. Therefore, a sensor will be regarded installed in the system as long as its corresponding sensor constraint is included in the system model. A consequence of using sensor constraints to model sensors, is that all system variables V needs to be considered as unknown. It is worth noting that the major works devoted to sensor placement for diagnosis use the same strategy to handle sensors, e.g. Krysander and Frisk [2008], Rosich *et al.* [2007] and Yassine *et al.* [2008a].

2.3 Diagnosis specifications

In this section diagnosis specifications used throughout the paper are introduced. First, however, it is assumed without loss of generality, that a fault can only affect one model constraint. This implies that when a fault occurs in the system, only one constraint may become inconsistent with the observation. Hence, a one-to-one relationship between faults and model constraints is established. This allows to characterise the diagnosis capabilities by just regarding the constraints of the model.

Furthermore, the structure of the model (or the structural model) will be used to derive the diagnosis properties. A structural model is a simplified description of analytical model where only counts which variables depend on which equations. The structural model can therefore be represented by

a bipartite graph (Blanke *et al.* [2006]) where K is one set of constraint nodes, V is the other set of variable nodes and A is the set of edges that connect constraints with variables according to

$$A = \{(k, v) \mid k \in K \text{ depends on } v \in V\} \quad (2)$$

The structural model permits to extend some diagnosis properties to properties of the Dulmage-Mendelsohn decomposition [Dulmage and Mendelsohn, 1959], [Murota, 2000]. This decomposition decomposes the structural model in the under-constrained part K^-, the just-constrained part K^0 and the over-constrained part K^+. In Blanke *et al.* [2006], it was first noticed that fault detectability is strongly related with the over-constrained part, and in Krysander and Frisk [2008] this concept was further developed by also regarding fault isolability. Following these works, detectability, discriminability and diagnosability are defined by means of the properties of the Dulmage-Mendelsohn decomposition on the structural model K.

Definition 1 (Detectable constraint) *A constraint $k \in K$ is detectable if*

$$k \in K^+ \quad (3)$$

Equation (3) implies that at least it exists one minimal testable subset $K' \subseteq K$ such that $k \in K'$. Therefore, it can be guaranteed, theoretically speaking, that the inconsistency of a detectable constraint will be detected by at least one minimal testable subset.

Definition 2 (Discriminable constraints) *Two different constraints $k_1, k_2 \in K^+$ are discriminable if*

$$k_1 \in (K \setminus \{k_2\})^+ \quad (4)$$

This implies that for k_1 and k_2 to be discriminable between them, it must exist at least two minimal testable subsets, $K', K'' \subset K$, such that $k_1 \in K'$ and $k_2 \notin K'$ as well as $k_2 \in K''$ and $k_1 \notin K''$. Also note that the discriminable constraint definition is the same as the fault isolability definition used by Krysander and Frisk [2008].

Definition 3 (Diagnosable constraint) *A detectable constraint $k \in K^+$ is diagnosable if*

$$(K \setminus \{k\})^+ = K^+ \setminus \{k\} \quad (5)$$

This definition ensures that no other model constraint belongs to the same minimal testable subsets as the diagnosable constraint k belongs to. In some sense, diagnosability can be viewed as a special case of discriminability where all the remaining model constraints are involved, i.e. all the equations in $K^+ \setminus \{k\}$ are discriminable from k.

Diagnosis specifications dealing with detectability, discriminability and diagnosability are represented by:

- the set of constraints, K^{spec}_{det}, that must be at least detectable.
- the collection, $\mathbb{K}^{spec}_{disc}$, of constraints sets such that two constraint from different sets must be discriminable.
- the set of constraints, K^{spec}_{diag}, that must be diagnosable.

Note that, according to these definitions, the diagnosis specifications K^{spec}_{diag}, $\mathbb{K}^{spec}_{disc}$ and K^{spec}_{det} are meaningful if the two following expressions hold,

$$K_i \cap K_j = \emptyset \quad \text{for } K_i, K_j \in \left(\mathbb{K}^{spec}_{disc} \cup \{K^{spec}_{diag}\}\right); i \neq j \quad (6)$$

$$\left(\bigcup_{K_i \in \mathbb{K}^{spec}_{disc}} (K_i) \cup K^{spec}_{diag} \right) \subseteq K^{spec}_{det} \subseteq K \quad (7)$$

Expression (6) avoids discriminating or diagnosing between the same equation, which is impossible for obvious reasons, therefore it is imposed that all the constraint sets in $\mathbb{K}^{spec}_{disc}$ together with K^{spec}_{diag} must be disjoint. Expression (7) ensures that both diagnosable and discriminable components are also detectable, according to Definition 2 and Definition 3. If these properties are satisfied, the diagnosis specifications are qualified as consistent in K.

3 COMPUTING DETECTABILITY, DISCRIMINABILITY AND DIAGNOSABILITY

The computation of the detectable, discriminable and diagnosable sets is done by means of the Dulmage-Mendelsohn decomposition. According to Definition 1, the under-constrained and the just-constrained parts, K^- and K^0, of the model are the parts that only contain non-detectable constraints. Therefore, it is only needed to obtain the over-constrained part in order to determine the set of detectable constraints, i.e. $K_{det} = K^+$.

Given an over-constrained (detectable) set of constraints K^+, the discriminable and diagnosable constraints can be computed by regarding the equivalent class defined in Krysander *et al.* [2008]. Constraints $k_1, k_2 \in K^+$ are in the same equivalent class if the equivalent relation between k_1 and k_2 defined by

$$k_2 \notin (K \setminus \{k_1\})^+ \quad (8)$$

holds. This means that the over-constrained part, K^+ of the model can be partitioned in $(K_1, \ldots, K_n)$ equivalent classes. Then, according to (8), if K_i only involves one constraint ($|K_i| = 1$) then this constraint is diagnosable, whereas if K_i involves more than one constraint ($|K_i| > 1$) then these constrains are pairwise non discriminable, i.e. all the constraints in K_i come together in a minimal testable subsets. Such sets with cardinality greater than one are named *linked blocks* in Yassine *et al.* [2008a].

To compute an equivalent classes from a set of detectable constraints K_{det} it suffices to see which constraints are not in the over-constrained part

at removing one constraint, $k \in K_{det}$. Due to K_{det} is an over-constrained set (it has no under and just-constrained parts), the equivalent equations appear to the just-constrained part when k is removed. Therefore, an equivalent class can be computed as

$$K_i = (K_{det} \setminus \{k\})^0 \cup \{k\} \tag{9}$$

Algorithm 1 computes the detectable constraints set, K_{det}, the discriminable constraint sets, $\mathbb{K}_{disc}$, and the diagnosable constraint set, K_{diag}, within a structural system model K. The algorithm first determines the set of detectable constraints K_{det}. Once the detectable set is found, then all the equivalent classes are computed from K_{det}. The equivalent classes $(K_1, \ldots, K_n)$ are computed by applying (9). Then, the computed equivalent class K_i is inserted into the $\mathbb{K}_{disc}$ or the K_{diag} according to the number of constraints involved in it.

Algorithm 1
$(K_{det}, \mathbb{K}_{disc}, K_{diag}) = \text{decompose}(K)$
Find the different subsets of constraints in K according to definitions 1, 2 and 3

Require: A structural model K
  $K_{det} \leftarrow K^+$
  $\mathbb{K}_{disc} \leftarrow \emptyset$
  $K_{diag} \leftarrow \emptyset$
  $K' \leftarrow K_{det}$
  $i \leftarrow 1$
  while $K' \neq \emptyset$ **do**
    Select $k \in K'$
    $K_i \leftarrow (K_{det} \setminus \{k\})^0 \cup \{k\}$
    if $|K_i| > 1$ **then**
      $\mathbb{K}_{disc} \leftarrow \mathbb{K}_{disc} \cup \{K_i\}$
    end if
    if $|K_i| = 1$ **then**
      $K_{diag} \leftarrow K_{diag} \cup \{K_i\}$
    end if
    $K' \leftarrow K' \setminus K_i$
    $i \leftarrow i + 1$
  end while

Note that when an equivalent class is found by removing a constraint it is not necessary to repeat the procedure with the remaining constraints in the equivalent class since the same set would be found more than once. This is avoided with the variable in K' which is a stack of possible constraints that can be removed. When an equivalent class K_i is found, its constraints are removed from K', thus no repeated equivalent classes are found. Algorithm 1 is motivated from Krysander *et al.* [2008] where equivalent classes are computed using similar proceeding.

4 TESTING DIAGNOSIS SPECIFICATIONS

Consider a system modeled by a constraint set K, where K_{det}, $\mathbb{K}_{disc}$ and K_{diag} represents the diagnosis performance of the system, while K_{det}^{spec}, $\mathbb{K}_{disc}^{spec}$ and K_{diag}^{spec} are the required specifications. It is obvious that if $K_{det}^{spec} = K_{det}$, $\mathbb{K}_{disc}^{spec} = \mathbb{K}_{disc}$ and $K_{diag}^{spec} = K_{diag}$ then it can be stated that diagnosis specifications are fulfilled. However, there are other cases where diagnosis specifications are also fulfilled.

For instance, when the system has more detectable constraints than the ones specified, i.e.

$$K_{det}^{spec} \subseteq K_{det} \tag{10}$$

the detectability specifications are also fulfilled since there is no inconvenience for having extra detectable constraints.

Similar reasoning can be applied for the diagnosable constraints, therefore if

$$K_{diag}^{spec} \subseteq K_{diag} \tag{11}$$

the diagnosability specifications are fulfilled.

The case concerning discriminable constraints is a bit more complex since sets of non-discriminable constraints are handled. Given a set of non-discriminable constraints $K \in \mathbb{K}_{disc}$, this set fulfils specifications if there exists at most one set $K' \in \mathbb{K}_{disc}^{spec}$ such that it shares a constraint with K, otherwise it would mean that specified discriminable constraint are non-discriminable. This can be extended to all the set in $\mathbb{K}_{disc}^{spec}$, thus discriminability specifications are fulfilled if

$$|\{K' \in \mathbb{K}_{disc}^{spec} \mid K \cap K' \neq \emptyset\}| \leq 1 \tag{12}$$

for each $K \in \mathbb{K}_{disc}$.

According to the expressions in (10), (11) and (12), the diagnosis specifications are not fulfilled as long as a specified detectable constraint becomes non-detectable or a specified diagnosable constraint becomes non-diagnosable (i.e. is non-detectable or belongs to any set $K \in \mathbb{K}_{disc}$), or two specified discriminable constraints become non-discriminable (i.e. are non-detectable or both constraints belong to the same set $K \in \mathbb{K}_{disc}$). Otherwise the diagnosis specifications are fulfilled and expressions in (10), (11) and (12) hold.

Example 1 *Assume a system model with 10 constraints, $K = \{k_1, \ldots, k_{10}\}$ with the following diagnosis specifications*

$$K_{det}^{spec} = \{k_1, k_2, \ldots, k_6\}$$
$$\mathbb{K}_{disc}^{spec} = \{\{k_1, k_2, k_3\}, \{k_4, k_5\}\}$$
$$K_{diag}^{spec} = \{k_6\}$$

After computing diagnosis performance by means of Algorithm 1, the following sets are obtained:

$$K_{det} = \{k_1, k_2, \ldots, k_{10}\}$$
$$\mathbb{K}_{disc} = \{\{k_2, k_3\}, \{k_4, k_8\}, \{k_5, k_9\}, \{k_7, k_{10}\}\}$$
$$K_{diag} = \{k_1, k_6\}$$

which, according to (10), (11) and (12) the specifications are fulfilled.

Algorithm 2 tests whether the diagnosis specifications are fulfilled for a possible set of sensors to be installed in the system. The set S of candidate sensors contains the system variables in V that

will be measured. Therefore, given the set S, it is straightforward to construct the model of the system with the sensors installed in it by just adding the corresponding sensors constraint (see (1)) to the original model. Once the model with the sensor is obtained, the diagnosis properties are computed by means of Algorithm 1. Then detectability, diagnosability and discriminability specifications are verified according to (10), (11) and (12), respectively.

Algorithm 2
isFeasible($K, S, K^{spec}_{det}, \mathbb{K}^{spec}_{disc}, K^{spec}_{diag}$)
Check whether the sensor placement satisfies the specifications

Require:
1. A structural model K.
2. A set of candidate sensors $S \subseteq V$.
3. The diagnosis specifications, $K^{spec}_{det}, \mathbb{K}^{spec}_{disc}, K_{diag}$ in K.

Construct the sensor constraints K_S from S
$(K_{det}, \mathbb{K}_{disc}, K_{diag}) = \text{decompose}(K \cup K_S)$
if $(K^{spec}_{det} \not\subseteq K_{det})$ **then**
  return false
end if
if $(K^{spec}_{diag} \not\subseteq K_{diag})$ **then**
  return false
end if
for $K \in \mathbb{K}_{disc}$ **do**
  if $|\{K' \in \mathbb{K}^{spec}_{disc} \mid K \cap K' \neq \emptyset\}| > 1$ **then**
    return false
  end if
end for
return true

If the diagnosis specifications are verified then the set of candidate sensors S is a feasible solution for the sensor placement problem. Algorithm 2 returns a boolean value indicating whether S is a feasible configuration.

5 OPTIMAL SENSOR SEARCH

5.1 Search requirements

The sensor placement problem is addressed by searching the optimal sensor configuration such that diagnosis specifications are fulfilled when these sensors are installed in the system. The cost of sensor can be motivated by several reasons, e.g. the purchase price, the installation difficulties or the measurement performances. However, the nature of the problem is combinatorial which means that exhaustive search are not feasible when the number of sensors to be considered grows. Here, to carry out efficiently the optimal search, two requisites are imposed:

- **requisite 1**: Given two sensor candidates S_1 and S_2 such that $S_1 \subset S_2$ then the cost of S_1 must be lower than the cost of S_2. This is summarised in the next expression

$$S_1 \subset S_2 \leftrightarrow C(S_1) < C(S_2) \tag{13}$$

  where $C(S)$ denotes the cost of sensor set S.

- **requisite 2**: If the sensor set S does not fulfil diagnosis specifications then no subset of S can fulfil diagnosis specifications, i.e.

$$S \text{ is not a solution} \rightarrow S' \text{ is not a solution} \tag{14}$$

  for any $S' \subseteq S$.

It is assumed, for the sake of simplicity, that the cost $C(s)$ of installing a sensor $s \in S$ is a positive real number. The cost of a sensor configuration S is defined as

$$C(S) = \sum_{s \in S} C(s) \tag{15}$$

Therefore, using the cost function in (15), requisite 1 is satisfied.

Concerning the requisite 2, in Travé-Massuyès *et al.* [2006] it is shown that given the family of minimal testable subsets generated with S sensors installed in the system, any sensor configuration S' such that $S' \subset S$ produces a subfamily of such minimal testable subsets, i.e. all the minimal testable subsets generated with S' are also generated with S. Hence, it results that when S does not fulfil diagnosis specification, neither fulfils S' since no new minimal testable subset can be generated. Therefore expression in (14) holds and requisite 2 is satisfied.

It is important to point out that expression in (14) does not hold for fault in sensors because this kind of faults depends on whether the sensor is installed. For instance, consider a sensor such that its corresponding constraint must be detectable. Then, it may happen that when the sensor is installed, its corresponding sensor constraint is non-detectable (diagnosis specifications are not fulfilled), however when the sensor is not installed the diagnosis are fulfilled since no sensor constraint, detectable or non-detectable, is expected. Therefore, the present approach is restricted to not handle fault in sensors in order to preserve requisite 2.

As it will be shown in next section, requisites 1 ans 2 play a fundamental roll to perform the optimal search efficiently.

5.2 Search strategy

First, the search space containing all the possible sensors configurations is represented by a graph-tree. To facilitate the comprehension of how the optimal search is performed, a small example with four sensors measuring four system variables, $S = \{v_1, v_2, v_3, v_4\}$, is used. The cost of each sensor in S is respectively $[1, 5, 7, 2]$. The search space for this example is depicted in figure 1, where the sensors are ordered according to decreasing costs, i.e. (v_3, v_2, v_4, v_1).

The concept of *node* which contains two vectors of sensors is introduced, i.e.

$$node.S$$
$$node.R$$

$node.S$ contains the candidate sensor configuration and $node.R$ contains the sensors of $node.S$

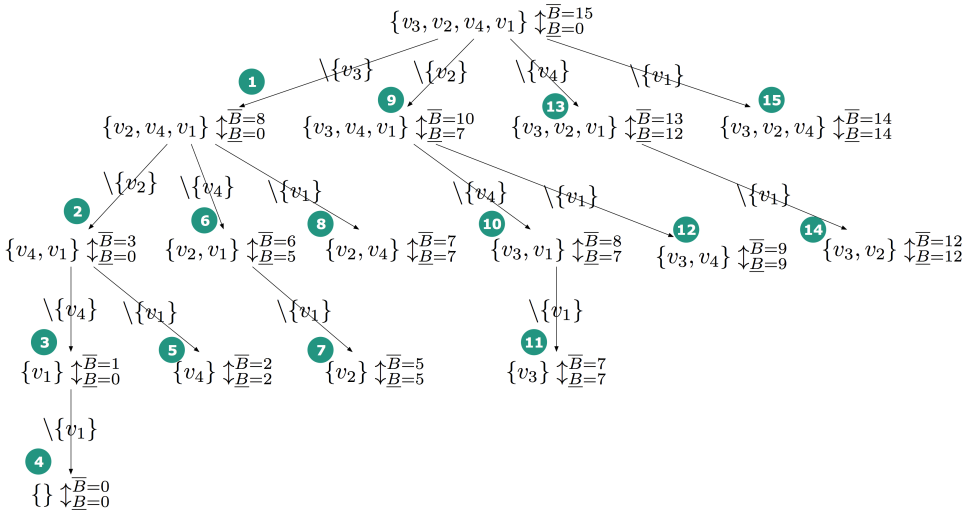

Figure 1: Search tree for 4 sensors

not yet removed in previous nodes of the same level. In other words, $node.R$ contains the sensors that can be removed to create sub-nodes.

Furthermore, an upper bound, $\bar{B}$, and a lower bound, $\underline{B}$, are defined for each node. The $\bar{B}$ is the cost of the sensor configuration in the node, while the $\underline{B}$ is the lowest cost reachable by exploring sub-nodes. The upper bound of a node is computed by

$$\bar{B}(node) = C(node.S)$$

and the lower bound by

$$\underline{B}(node) = C(node.S) - C(node.R)$$

Following, the construction of the tree in Figure 1 is detailed. Given a node, its sub-nodes are obtained by removing the sensors in $node.R$ one by one according to decreasing cost order, i.e. the sensor with the highest cost is firstly removed from the node, then the next sensor with the highest cost and so on until the sensor with the lowest cost is removed from the node. The circled numbers in Figure 1 show the sequence to obtain the whole tree.

The search strategy is based on a deep-first search by choosing first the nodes with lowest costs. Throughout the search, the best solution is updated in S^* when a feasible solution with lower cost than the current best one is found.

The deep-first search is stopped at some node when

1. the node is not a feasible solution for the sensor placement problem according to Algorithm 2
2. or the lower bound $\underline{B}$ of the node is greater than the cost of the current best solution.

These two conditions are motivated by requisites 1 and 2. According to (14) in requisite 2 and observing the Figure 1, if a node is not a feasible solution then its sub-nodes can not be feasible solutions, therefore the search is cut.

On the other hand, the lower bound $\underline{B}$ is an indicator of the best cost reachable from a node, i.e. the lowest cost of its leaf nodes, (see Figure 1). If condition 2 is fulfilled, i.e. $\underline{B}(node) > C(S^*)$, then it can be ensured that there is no sub-node with lower cost than $C(S^*)$ since (13) from requisite 2 holds along the current node until the leaf nodes. Therefore, no better solution can exist in the sub-nodes and the search is also cut.

Algorithm 3 performs recursively the optimal search for the sensor placement. The algorithm generates child nodes by removing, according to the costs, sensors in $node.R$. The deep-first search is performed by choosing first the child node with the lowest cost. If this child node does not fulfil conditions 1 and 2 then the deep-first search is not stopped and the algorithm recursively searches in a deeper child nodes. Otherwise, the search is cut (all the sub-nodes are rejected) and the algorithm moves to the next child node with the lowest cost and so on until all the child nodes have been tested. Note that Algorithm 2 is used to verify specifications. In order to make the algorithm more readable, the three-sets $(K_{det}^{spec}, \mathbb{K}_{disc}^{spec}, K_{diag}^{spec})$, of specification sets is condensed in the $spec$ variable, i.e $spec = (K_{det}^{spec}, \mathbb{K}_{disc}^{spec}, K_{diag}^{spec})$.

Let S be the set of all sensors that can be installed in the system, then the algorithm is initialised with $node.S = S$ and $node.R = S$. It is also assumed that diagnosis specifications are fulfilled with all the possible sensors installed in the system, thus the best solution is initialised as $S^* = S$. Note that if this assumption does not hold then there is no possible solution, according to (14), for the sensor placement problem.

Since all the possible branches in the search are investigated and cutting operation ensures that no better solution is missed, the global optimal solution is ensured. To clarify the proceeding of Al-

Algorithm 3
$S^* = \text{searchOpt}(node, S^*, K, spec)$

for all $s \in node.R$ ordered in decreasing cost **do**
  $childNode.S \leftarrow node.S \setminus \{s\}$
  $node.R \leftarrow node.R \setminus \{s\}$
  $childNode.R \leftarrow node.R$
  if $\underline{B}(childNode) < C(S^*)$ **and** $\text{isFeasible}(K, childNode.S, spec)$ **then**
    if $C(childNode.S) < C(S^*)$ **then**
      $S^* \leftarrow (childNode.S)$
    end if
    $S^* \leftarrow \text{searchOpt}(childNode, S^*, K, spec)$
  end if
end for
return S^*

Table 1: Optimal search example

iteration	current node	cost	cutting condition	best node	cost
0	0	15	—	0	15
1	1	8	—	1	8
2	2	3	(1)	1	8
3	6	6	(1)	1	8
4	8	7	—	8	7
5	9	10	(2)	8	7
6	13	13	(2)	8	7
7	15	14	(2)	8	7

gorithm 3, next example shows how the search is carried out by the algorithm.

Example 2 *Assume the set of four sensors* $\{v_1, v_2, v_3, v_4\}$ *with the cost mentioned above. In order to follow the search, sensor configurations fulfilling specifications are known beforehand:*

$$\{v_1, v_2, v_3, v_4\}, \{v_1, v_2, v_4\}, \{v_2, v_4\},$$
$$\{v_1, v_2, v_3\}, \{v_2, v_3\}, \{v_2, v_3, v_4\}$$

which corresponds, respectively, to nodes 0, 1, 8, 13, 14 and 15 in Figure 1. Table 1 shows the sequence that the algorithm uses to find the optimal solution. After completing the sequence, the algorithm would return node 8 (v_1, v_2) *with a cost of 7 as the optimal solution.*

6 APPLICATION

The algorithm for sensor placement summarized in section 5 has been applied to an electronic circuit (see Figure 2).

This electronic circuit is modeled by 13 constraints and 14 unknown variables. The corresponding bi-adjacency matrix which represents the structural model (Blanke *et al.* [2006]) is given in Table 2. For this example, it is assumed that all the variables can be measured, however different sensors costs are assigned in order to represent preferences at installing sensors. Next table shows the cost of installing each sensor:

	v_0	v_1	v_2	v_3	v_4	i_1	i_2	i_3	i_4	v_{1a}	v_{1b}	v_{1c}	v_{4a}	v_{4b}
cost	1	4	4	4	2	1	8	8	8	4	4	4	4	4

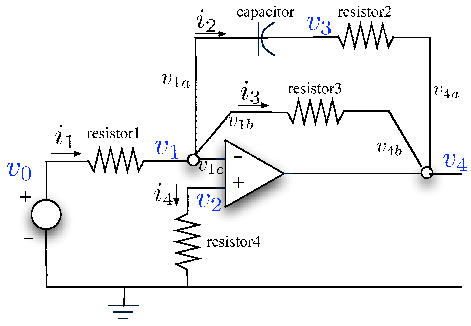

Figure 2: Scheme of the electronic circuit

Table 2: Structural model of the electronic circuit

	v_0	v_1	v_2	v_3	v_4	i_1	i_2	i_3	i_4	v_{1a}	v_{1b}	v_{1c}	v_{4a}	v_{4b}
k_1			×									×		
k_2						×	×	×						
k_3		×								×				
k_4		×									×			
k_5		×										×		
k_6	×	×				×								
k_7				×			×			×				
k_8				×			×						×	
k_9		×						×						×
k_{10}			×						×					
k_{11}					×								×	
k_{12}					×									×
k_{13}	×													

Let consider, for this example, the following diagnosis specifications:

$$K_{det} = \{k_1, k_6, k_7, k_8, k_9, k_{10}, k_{13}\}$$
$$\mathbb{K}_{disc} = \{\{k_6\}, \{k_7\}, \{k_8\}, \{k_9\}, \{k_{10}\}, \{k_{13}\}\}$$
$$K_{diag} = \{k_1\}$$

In order to find the cheapest sensor placement that satisfies specifications, Algorithm 3 is used and it returns the following result:

$$S^* = \{v_0, v_2, v_3, v_4, i_1, i_4, v_{1c}\}$$

with a minimal cost $C(S^*) = 24$.

Algorithm 3 was implemented in Matlab and it lasts 0.7 seconds to find the solution. The number of possible sensors configuration is $2^{14} = 16384$, however the algorithm only visits 190 nodes. 62 nodes out of these 190 need to be tested whether diagnosis specifications are fulfilled, i.e. the Algorithm 2 receives 62 calls. Only 28 visited nodes do not cut the search.

7 CONCLUSION

An optimal approach for sensor placement satisfying diagnosis specifications has been proposed. In contrast to other approaches, diagnosis specifications concern fault detectability, discriminability and diagnosability which are attained by means of the Dulmage-Mendelsohn decomposition. The computation of this decomposition has no computational complexity, thus Algorithm 2, developed

to test specifications, does not present computational time problems.

Furthermore, a simple but efficient algorithm based on heuristic rules has been developed in order to carry out the optimal solution search. Thanks to the proposed algorithm, cost optimal sensor placements satisfying diagnosis specifications are possible without designing minimal testable subsystems a priori. Algorithm 3 has, in general, theoretically speaking a computational exponential time. Compared to the other existing method, it is able to rapidly find an optimal solution.

Given a set of n candidate sensors, the worst case appears when the algorithm has to traverse all the nodes in the upper half part of the lattice of all the sensor combinations, i.e. $2^n/2$ nodes are visited. This case is present when the optimal solution involves $n/2$ (or $(n-1)/2$ if n is odd) sensors, and no cutting operations can be performed (conditions 1 and 2 are not met) during the search. This situation implies that the sensor cost is the same for all the n sensors and, furthermore, any sensor set involving more that $n/2$ sensors is a feasible solution (condition 1 is not met), whereas the remaining set of sensors involving less that $n/2$ sensors are not feasible solutions (condition 2 is never applied to cut the search). Hence, the efficiency of the algorithm strongly depends on how the sensor costs are set.

To take advantage of cutting condition 2, it is recommended to set the sensor cost as differed as possible among them. In this way, once a lower cost solution is found, large branches of subnodes can be pruned from the search tree, leading to a more efficient search. Therefore, the algorithm can handle, in general, situations where the number of possible sensors is large. For instance, the proposed algorithm has been tested in a system with 45 possible sensors where a solution has been found in 15 minutes approximately, whereas such system is insolvable with other existing approaches.

To make simpler the approach, it is assumed that each fault can only effect one constraint or a sensor has to be modelled by the sensor constraint in (1). However, these assumptions could be easy extended to more general cases.

Furthermore, the proposed approach is not able to handle faults in the sensors that may be introduced to fulfilled diagnosability specifications. A future extension will be an improvement of the presented method in order to extend the approach to diagnosis specifications related to possible added sensors.

REFERENCES

M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki. *Diagnosis and Fault-tolerant Control*. Springer-Verlag, 2006.

C. Commault, J.-M. Dion, and S. Yacoub Agha. Structural analysis for the sensor location problem in fault detection and isolation. In *SAFEPROCESS'2006*, Beijing, China, Aug. 30th-Sep.1st 2006.

A. L. Dulmage and N. S. Mendelsohn. A structure theory of bi-partite graphs of finite exterior extension. *Transactions of the Royal Society of Canada*, 53(III):1–13, 1959.

Amir Fijany and Farrokh Vatan. A new efficient algorithm for analyzing and optimizing the system of sensors. In *Proc. 2006 IEEE Aerospace Conference*, Big Sky, Montana, USA, March 4–11, 2006.

Mattias Krysander and Erik Frisk. Sensor placement for fault diagnosis. *IEEE Trans. Syst., Man, Cybern. A*, 38(6):1398–1410, 2008.

M. Krysander, J. Åslund, and M. . Nyberg. An efficient algorithm for finding minimal overconstrained subsystems for model-based-diagnosis. *IEEE Trans. Syst., Man, Cybern. A*, 38(1):197–206, 2008.

F. Madron and V. Veverka. Optimal selection of measuring points in complex plants by lineair models. *AICheE*, 38(2):227–236, 1992.

D. Maquin, M. Luong, and J. Ragot. Fault detection and isolation and sensor network design. *European Journal of Automation*, 31(2):393–406, 1997.

K. Murota. *Matrices and Matroids for Systems Analysis*. Springer-Verlag, 2000.

S. Ploix, A. Yassine, and J.-M. Flaus. An improved algorithm for the design of testable subsystems. In *The 17th IFAC World Congress*, Seoul, Corea, 2008.

B. Pulido and C. Alonso. Possible conflicts, arrs, and conflicts. In *13th International Workshop on Principles of Diagnosis (DX02)*, pages 122–128, May 2002.

A. Rosich, R. Sarrate, V. Puig, and T. Escobet. Efficient optimal sensor placement for model-based FDI using and incremental algorithm. In *Proc. 46th IEEE Conference on Decision and Control*, pages 2590–2595, New Orleans, USA, December 12–14, 2007.

A. Rosich, R. Sarrate, and F. Nejjari. Optimal sensor placement for FDI using binary integer linear programming. 20th International Workshop on Principles of Diagnosis, DX'09, 2009.

L. Travé-Massuyès, T. Escobet, and X. Olive. Diagnosability analysis based on component supported analytical redundancy relations. *IEEE Trans. Syst., Man, Cybern. A*, 36:1146 – 1160, 2006.

A. Yassine, S. Ploix, and J.-M. Flaus. A method for sensor placements taking into account diagnosability criteria. *Applied Mathematics and Computer Science*, 18(4), 2008.

A. Yassine, S. Ploix, and J.-M. Flaus. Structural approach for sensor placement with diagnosability purpose. In *The 17th IFAC World Congress*, Seoul, Corea, 2008.