© 2014 IJEDR | Volume 2, Issue 4 | ISSN: 2321-9939

A Genetic Algorithm Approach for Diagnosability Analysis

[1]Ruben Leal, [2]Jose Aguilar, [3]Louise Travé-Massuyès, [4]Edgar Camargo, [5]Addison Ríos
[1,4]PDVSA, [2,5]CEMISID ULA, [3]LAAS-CNRS
[1]Automation Department,
[1]PDVSA, Cabimas, Venezuela

***Abstract* - In this work we propose to use an approach based on genetic algorithms to obtain analytical redundancy relations to study the diagnosability property on a given continuous production system. Diagnosability analysis for production systems examines the detectability property (the faults are discriminable from the normal behavior of the system) and the isolability property (the faults are discriminable between them). The redundancy relations are based on the minimal test equation support and in a structural analysis over a bipartite graph. The faults analysis is studied using a multi-objective fitness function in a genetic algorithm, which describes the different constraints to be covered in order to reach the diagnosability property on the system. Our approach is tested in a theoretical example and in a real continuous system, a process of extraction of oil by gas injection.**

***Index Terms* - Genetic Algorithm, Diagnosability, Structural Analysis, Analytical Redundancy Relations, Gas Lift Well.**

I. INTRODUCTION

Most production processes require constant supervision and control of the facilities associated. Particularly, it is necessary the diagnosis and early detection of faults, in order to have enough time to counteract the consequences that could bring faults. Some of the possible operations to counteract are reconfiguration, maintenance or reparation actions. Early detection can be achieved by acquiring information, mainly using mathematical models based on measured or calculated values from the processes studied. Moreover, for fault diagnosis is important to use cause-effect relationship of each state [2, 3].

Basically, fault means any change in the behavior of any of the components of the system, so that it cannot longer fulfill the function for which it was designed [1]. Normally, a typical diagnosis system consists in detection and isolation of a set of faults. A diagnostic utilizes observations, i.e. measurements from the system under studied, to determine if a specific behavioral mode is present in the system or not.

In general, the diagnosis analysis has been studied in the literature from two points of view. The Fault Detection and Isolation (FDI) community, which bases the foundations of its solution approaches on engineering disciplines such as control theory and statistical decision making, and the Diagnosis Community (DX), which bases the foundations of its solution approaches on the fields of computer science and artificial intelligence. Each community has developed its own terminology, tools, techniques, and approaches to solve diagnosis problems. Our approach is a mixed between the two communities' theories, using a mathematical model and intelligent techniques to solve the problems.

On the other hand, when a system has the diagnosability property it can detect and isolate all considered faults. It is a very important property that a given system should meet. The problem of the analysis of diagnosability in the area of continuous processes is very hard, and there are not a lot of works [8, 12].

This paper addresses this problem. We propose an approach for the analysis of diagnosability, based on a hybrid approach between structural models and intelligent techniques. For that, we identify the different structural models of diagnosis for a diagnosability analysis in the area of continuous processes. Additionally, we use a Genetic Algorithm (GA) to find the set of analytical redundancy relations. In this way, in this paper we propose an analysis of diagnosability based on the residual generation, applying GA to find a set of redundancy relations.

We test our approach in two cases, in a theoretical example and in a real-world process of extraction of oil by gas injection. Particularly, in the oil extraction industry is essential to achieve maximal production, for which the diagnostic process must be improves and streamlined. This study considers one of the most used methods for the artificial extraction of oil, which relies on wells with gas injection [18]. Oil wells based on gas injection with highly oscillatory flow, are a major problem in the oil industry. The efforts to find low-cost solutions based on automatic control and fault diagnosis have been carried out in both the academic and the industrial communities for a long time, [17, 19, 20]. In the literature, there are several diagnostic studies related to pipelines, storage tanks and wells, but previous works do not include diagnosability analysis in gas lift wells.

This paper is organized in the following way. The next section presents some concepts about the fault diagnosis problem. The section three introduces the theoretical bases of the diagnosability problem based on residuals. The section four presents our general algorithm for the diagnosability. The section five describes our approach based on residuals and GAs for the case of diagnosability problem. Finally, the last section presents some experiments: an application to a theoretical example, and another application to a real-world process of extraction of oil by gas injection, and analyzes the results.

II. Theoretical Foundation

In a fault diagnosis system there are three key concepts: The *fault detectability* is the ability to detect certain faults. The diagnosis must be able to decide if there is a fault or not, as well as determine the instant of the apparition, from observations of the process. For that, it needs to compare the current behavior with the expected behavior of the system. The *fault isolation* is the ability to isolate a fault that has occurred, from the faults that are detectable. The *fault identification* studies the differences between the normal behavior and the current behavior in the system, in order to determine the depth and magnitude of the faults.

The fundamental concept in our work is the diagnosability. We start with its definition and the definition of the others close concepts.

Definition 1: Diagnosability. A system has this property if it can detect and isolate all considered faults [7, 8]. The faults that are isolate in a process are often referred as monitored faults, whereas the faults that not isolate are called non-monitored faults.

A typical approach for diagnosability in dynamic systems is the model based diagnosis (MBD) [9]. In the MBD, the fundamental aspects are the definition of a process model, the comparison of the functioning of the process with the model, and the analysis of the behavior of them. A possible MBD technique is based on the generation of residue [9].

There are several technics for the generation of residuals, but all consist in the measurements. If the observed situation does not meet the estimate made by the model for a given situation, then it is concluded that there is a fault, and further analysis of the differences are carried out to identify the specific component of the fault. A way to generate residuals, which is used in this paper, is based on analytical redundancy relations (ARR) [4]. It uses analytical mathematical models that characterize the system, to reproduce the behavior of the system under evaluation.

The approach for generating a residual is based on a finite sequence of calculations that ends with the evaluation of an analytically redundant equation. Similar approaches have been described and exploited in [1]. The ARR only contains measure of known variables, and is composed of a subset of equations from the model. ARR allows us to check whether the measured variables are consistent with the model.

A residual is a signal ideally zero in the non-faulty case and non-zero otherwise. A residual generator takes measured variables from the system as input, and produces a residue as output. The method for residual generation presented in this paper relies on structural analysis and sequential generation [2, 4, 6].

The residual generation approaches have in common that the sub-systems should be over-determined to include the required redundancy. Several algorithms for calculating ARRs from over-determined systems have been proposed in [2, 7, 8]. Particularly in [2] is proposed an algorithm that analyzes the structure of a system to detecting redundancy using the ARR approach and is developed in detail in [7, 8]. In [6] is proposed a sequence to generate the residuals like a post-processing [6].

In general, the diagnosability depends on the residuals that can be generated, as it depends on the redundancy embedded in the system. Decoupling of faults in a set of tests based on residuals, means that the residuals must be sensitive to, or respond to different subsets of faults. Thus, decoupling of faults is a fundamental problem in residual generation for fault isolation, if the diagnosability wants to be reached.

Structural Analysis Based On Residuals

The structural analysis is a set of tools to explore the fundamental properties of a system using a structural model, either in the form of a structural graph or incidence matrix [6]. In our work, a structural analysis of the system is used for the faults detection and isolation, following the approaches used by the community of Fault Detection and Isolation (FDI) [1]. At the following we present a resume about the theoretical aspects used by our approach.

A structural model is a representation of a system in which only couplings between variables and equations are retained [10, 11]. The structural model contains only the information of which variable belongs to which equation, regardless of the value of the parameters and the detailed form of the mathematical expression [4]. A structural model can be represented by a bipartite graph or an incidence matrix. Let's call this model is *M(X,Z,E,F)*, where E is a set of equations $E = \{e_1,..., e_m\}$, X is a set of unknown variables $X = \{x_1, \dots, x_n\}$, Z is a set of known variables $Z=\{z_1,...,z_p\}$. and $F=\{f_1,...,f_o\}$ is a set of fault parameters which modify the normal behaviour of the system (they are considered as unknown variables). In the case of a differential model, it is necessary to add a fifth set, $D = \{\dot{x}_1, \dots, \dot{x}_n\}$, which contains the derivatives of the variables of X.

We assume that the equations in the set E have the form.

$$e_i: h_i(\dot{x}, x, y, f) = 0,\ 1 \le i \le m$$

Where, $\dot{x}$, x, f and y are vectors of the sets D, X, F and Y, respectively.

Example 1: consider.

$$\begin{aligned} e_1&: \dot{x}_1 = -x_1 + u + f_1 \\ e_2&: \dot{x}_2 = x_1 - 2x_2 + x_3 + f_2 \\ e_3&: \dot{x}_3 = x_2 - 3x_3 \\ e_4&: y_1 = x_2 + f_3 \\ e_5&: y_2 = x_2 + f_4 \\ e_6&: y_3 = x_3 + f_5 \end{aligned}$$

The structural model of the system is as follows: $e = \{e_1, e_2, e_3, e_4, e_5\}$, $X = \{x_1, \dot{x}_1, x_2, \dot{x}_2, x_3, \dot{x}_3\}$, $D = \{\dot{x}_1, \dot{x}_2, \dot{x}_3\}$, $Y = \{y_1, y_2, y_3, u\}$, $F = \{f_1, f_2, f_3, f_4, f_5\}$.

This structure of the system is a representation as it shown in Figure 1 of which variables are involved in the different equations. This abstraction allows us to study the diagnosability properties, independently of the linear or nonlinear nature of the systems. However, it must be kept in mind that results obtained with such a structural representation are a best case scenario.

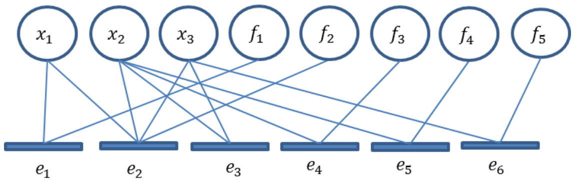


Figure 1 Bipartite Graphs of the example.

Now, we present some definitions to use the structural analysis for diagnosis purposes.

Definition 2: ARR for M(X,Y,E,F). Let *M(X,Y,E,F)* be a model, then an equation $r_i: h(y, \dot{y}, \ddot{y}, \ldots) = 0$ is an ARR for *M(E,X,Z,F)*,if for each *y* consistent with *M(E,X,Y,F)*, the equation is fulfilled [3].

These relationships can be derived only if the model has more equations than unknown variables, i.e. if the system is structurally over-determined (SO) [10].

Example 2: According to example 1, an ARR would be.

$$r_1 = y_1 - y_2 = f_3 - f_4$$

An ARR can be used to check if the observed variables *z* are consistent with *M(E,X,Z,F)*, and can be used as the basis of a residual generator.

Definition 3: Residual Generator for M(E,X,Y,F). A system taking a subset of the variables *z* as input, and generating a scalar signal r_i as output, is a residual generator for the model *M(E,X,Y,F)*, if for all *y* consistent with *M(E,X,Y,F)*, it hold that $\lim_{t\to\infty} r_i(t) = 0$ in normal behavior otherwise $r_i(t) \neq 0$ [11].

The structure of the system can be abstracted as a representation of which variables are involved in the different equations which make up the model of the system. The structural model of a system is an abstraction that allows one to study the diagnosability properties, independently of the linear or nonlinear nature of the systems. However, it must be kept in mind that results obtained with such a structural representation are a best case scenario. Causality considerations and the presence of algebraic and differential loops, determine which structural redundancies can be exploited for the design of residual generators.

Each r_i should be evaluated in order to decide if it can be used or not. Finally, the evaluation of each detection test constitutes the fault signature vector ($S = \{S_1, \ldots, S_n\}$), that is a set of vector in order to isolate the fault.

Given a set of vector $S = \{S_1, \ldots, S_n\}$ and a set of faults $F=\{f_1,\ldots,f_o\}$, the theoretical fault signature matrix can be defined codifying the effect of every fault in a residual [12].

Definition 4: The fault signature matrix of M. It is a table obtained by the concatenation of all possible signatures of faults. Each row corresponds to an *ARR* and each column to a failure mode. A "1" in position (*ij*), indicates that the fault *j* is detected by the *ARR i* [1].

F(M) is the set of faults that affect either equation in *M*, then the diagnosability ability is achieved if the system complies with the following definitions.

Definition 5: Detectability for M(E,X,Z,F). A fault F_o, where $o = 1, \ldots, n$ which belongs to *F(M)* in the diagnosis system of *M*, is detectable if there is a residue different from zero in the residual generator, i.e. $r_i \neq 0$.

Definition 6: Isolability for M(E,X,Y,F). When two signatures are identical, the related faults are considered non-decoupled, that mean they cannot be isolated [13]. Therefore, all signatures must be different from each other $S(f_o) \neq S(f_t), \forall o, t \in \{1, .., n\}, o \neq t$. The fault isolation will consist in looking for the theoretical fault signature in the fault matrix that matches with the observed signature, to distinguish between all the possible faults.

Example 3: Consider a diagnosis system containing a set of residuals $\{ARR_1, ARR_2, ARR_3\}$constructed to detect and isolate three faults $\{f_1, f_2, f_3\}$. The following fault signature matrix shows the sensitivity of ARRs to faults even in the system in normal behaviour *N*. Arr_1 is sensitive to faults f_2 and f_3, and so on. Each fault has a different signature, so we can isolate all considered faults.

Table 1 Fault Signature Matrix

	N	f_1	f_2	f_3
Arr_1	0	0	1	1
Arr_2	0	1	0	1
Arr_3	0	1	1	0

We adopt the design method of minimal structurally over determined (MSO) sets based on ARR [2]. Unobserved variables can be eliminated for the subset of equations [7].

Definition 7: Over-determined System (SO). M is an SO if the cardinality in *E* is greater that the cardinality in *X* i.e.$|E| > |X|$.

If the cardinality of *M* is $|E| = |X|$ then it is a Just-determined system, and if the cardinality of *M* is $|E| < |X|$ then is an Under-determined system. A condition that must be satisfied is that must be at least one more equation than unknown variables, which means that the system is over-determined.

Definition 8: Minimal Structurally Over-determined (MSO). A MSO contains only one equation more than unknown variables, and each MSO is equivalent to an ARR [7].

In [2] provides an algorithm that identifies the MSO, enabling the construction of more efficient ARRs. Each ARR correspond to an MSO.

In [7] introduced an algorithm and the notion of TES (Test Equation Support) which are sets of equations which express redundancy specific to a set of considered faults. Each TES corresponds to a set of faults which influence the residual generator constructed from the TES. The corresponding quantities expressing minimal redundancies are denoted minimal TES (MTES), and the set of MTES can be seen as a subset of the set of MSOs for the set of faults of interest of the system.

Definition 9: Test Equation Support (TES). Given a model M and a set of faults $F(M)$, an equations set M is a test equation support if M is a SO set, and if $F(M')$ correspond a part of the model, being $F(M)\neq 0$ for any $M' \subseteq M$, where M' is a SO set it holds that $F(M') \subseteq F(M)$[3].

Definition 10: Minimal Test Equation Support (MTES). A TES of M is a minimal TES if is the smaller subset sensitive to a fault with the degree of redundancy equal to one, i.e. one equation more than unknown variables.

Since there is a one-to-one correspondence between MTESs and ARR, we will only focus on MTESs in this paper, to generate residuals of the process. With that, we will be able build a signature matrix. A MTES set could be used to develop a consistency check for a part of the system, and a set of $F(M)$ can be detected with this consistency check [17]. With that, we will be able build a signature matrix.

A MTES set could be used to analyze a part of the system, and a set of $F(M)$ can be detected in this analyze. In our work, we have used residuals generation method based on, a sequence of step to successively solve the unknown variables involved in the equations set (see [8, 17]). In this way, it solves the system of equations, deleting the unknown variables one at a time. Then, the residual generator is created.

Many practical problems are modeled like the interaction between two different types of objects, i.e. between equations and variables, and can be expressed like a bipartite graph problem. Unlike [7, 8] we will use a matching in a bipartite graph to solve the problem of find the possible MTES's in a system model. For that we give the following definitions:

Definition 11: Bipartite Graph. is an ordered triple $G = (Z, E, \Gamma)$ such that Z and E are sets of vertices, $Z \cap E = \emptyset$, where $Z = Y \cup Z \cup F$, and Γ are the set of arcs in ***G***.

Definition 12: Matching. is a set of edges from graph *G*, where each arc has a node from *Z* an *E*.

Definition 13: Maximal Matching. is a matching *M* of a graph *G* with the property that it is not a subset of any other matching in graph *G*. In other words, a matching *M* of a graph *G* is maximal if every edge in *G* (*Z* and *E*) has a non-empty intersection with at least one edge in *M*.

Definition 14: Perfect Matching P. is a matching in a graph G that covers all its vertices $Z \cap E$.

Definition 15: Alternative Path. $P_\Gamma = (\Gamma_1, \Gamma_{2,\dots,}\Gamma_k)$ in a graph G, determines the subset of redundancy equations. It studies the path to eliminate the unknown variables from a TES.

The above definitions will help us solve the problem defined in the previous section of search of MTES's, and thus ARR. All the definitions made in this part will be used to apply intelligent techniques in the optimizing of search of alternative paths in a bipartite graph, unlike [7, 8].

III. The Genetic Algorithms Of Our Diagnosability Analysis Approach

Several researchers have successfully implemented GA for the redundancy analysis specifically GA-based approaches for the sensors placement problem [14, 15, 16]. This paper attempts to incorporate the GA optimization features to reach the full criteria of diagnosability. That is, our approach analyses the diagnosability property and then, if it is not fulfill, it solves the sensors placement problem to reach it.

In order to fulfill this criterion we developed two algorithms which during their executions invoke several GAs. The first algorithm (called the **MAIN**) simply calls the GA call *Detection*. The aim of the Detection algorithm is to check whether all the faults in study are detected, (see lines 1-2, of the main algorithm). In the line 2, if the detectability property is not satisfied **(**$MTES(f) = \emptyset\ \forall\ f = 1, \#faults$), or the isolability property is not satisfied ($ind\_opt \neq \emptyset$, where $ind\_opt \neq \emptyset$ is a control variable of the second algorithm).

The second algorithm (called *Detection*) searches an alternative matching in the model (a matching represent a redundancy), based on the theory of MTES. The algorithm finds the possible connections between variables and equations in order to eliminate the unobserved variables to fulfill the specific requirements of an ARR. Particularly; it will find populations of MTES invoking the first GA, called GA1. This algorithm also determines the detectability and the isolability (for the last case uses a second GA, called GA2, which uses the best MTES generated by GA1 to build the fault signature matrix).

Algorithm 1: **Main(G)**

```
1   MTES=Detection(G);
    % It determines detectability and isolability of the system
2      if MTES(f) = ∅ ∀ f = 1,#faults or ind_opt≠∅
3         Print: "there is not enough redundancy"
4      end
5   end
```

In the line 2 of the second algorithm is called *GA1* to determine all the possible MTES for each fault (see line1), the lines 4 to 6 guarantee that at least one MTES has been defined for each fault. The line 6 calls *det*, which determines the set of faults cover by each MTES and includes them in the set *S*. With this information, the algorithm verifies in line 8 if all the faults are covered, using the procedure *verify(S)*. In this way, it determines the detectability of the model (all the faults can be detected). Then, in line 9 it constructs the fault signature matrix (for that, it calls the procedure *iso*). In order to verify the isolability of the system, the procedure calls a second GA (line 10), called *GA2*, with the fault signature matrix as parameter. If the system is isolable, the best individual of GA2 fulfills the isolability propriety (the system has the diagnosability property). If this property cannot be reached, then it defines $ind\_opt \neq 0$ in order to call in the main algorithm the third algorithm, called *Placement*, which is used when there are faults not detected or isolated.

Algorithm 2: **Detection (G)**

1 **for** $k = 1, k = \#faults |\ f \in F$
2 $\quad MTES(f_k) = GA1(f_{k,} G)$;
$\quad$ % Determines the MTES's for each fault
3 **end**
4 **if** at least $\exists\ MTES(f) \neq \emptyset \mid \forall f \in F$
5 $\quad$ **for** $i = 1, i = \#faults |\ f \in F$
6 $\quad\quad S(i) = \det(i, MTES\ (f_i))$;
$\quad\quad$ % It determines the set of faults cover by the
$\quad\quad$ % MTES's of each fault
7 $\quad$ **end**
8 $\quad$ **if** verify(S) $= True$
$\quad\quad$ % Determines the detectability in the model
9 $\quad\quad S_m = \text{iso}(f_i, S)$;
$\quad\quad$ % Constructs the fault signature matrix
10 $\quad\quad ind\_opt = GA2(S_m)$
$\quad\quad$ % Determines the isolability in the faults ($ind\_opt = 0$)
11 $\quad$ **end**
12 **end**

GA is one of the most powerful heuristics tools for solving optimization problems, based on natural selection and the process of biological evolution [14]. A GA repeatedly modifies a population of individual solutions. At each step, the GA selects individuals randomly from the current population to be parents, and uses them to produce the descent for the next generation. Over successive generations, the population "evolves" towards an optimal solution. GAs can be applied to solve a variety of optimization problems that are not well suited for standard optimization algorithms, including problems in which the objective function is discontinuous, non-differentiable, stochastic, or highly nonlinear. In the next subsection we are going to describe our Gas.

GA1

Given a model *M*, its diagnosability property can be analyzed using a structural model of the system as bipartite graph (*G*), and is represented by the union of all the faults and all the unknown variables $G = G_X \cup G_F$. From G_F, the fault f_k is extracted, GA1 will generate a population of individuals with random combinations of active arcs in the model (an active arc is when the relation between this equation and the variable is selected to compose the possible MTES defined by this individual for this fault f_k, see below for more details). That is, each individual is going to define a possible MTES. The objective function measures if (possible MTES) reaches the detectability proprieties, the best individuals will be the input of the GA2, in order to study the isolability property.

An individual in GA1 is represented by a bits-string, where each bit is used to represent the arcs connecting the vertices in the bipartite graph, a_{ij}, a is the arc between vertices, i (the unknown variables *x* and *f* in the bipartite graph) and j (the equation *e* in the bipartite graph). If an arc is selected (active arc) $a_{ij} = 1$, else $a_{ij} = 0$ (it represents that this arc does not exist in the bipartite graph or is not an active arc). For our example in the section I, an individual is shown in Fig 2.

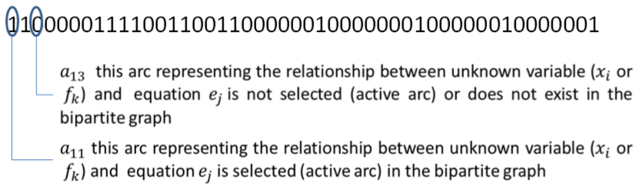


Figure 2. Bits string representation of an the individual in GA1

Hence, for every possible MTES for this fault *f* there is a unique bits-string sequence. The MTES (individual) is evaluated using fitness function which includes all the important design criteria of a MTES. The general algorithm of GA1is:

Genetic Algorithm 1: GA1(f_k, G)

```
1   begin
2       pop=generate_pop(G);
        % Generate initial population
3       Evaluation(FF_1,pop);
        % compute the fitness function to each individual
4       while not FF_1 = ∅ or Nb_generation do
5           begin
            % reproductive cycle begins
6               Reproduction (pop)
7               Evaluation (FF1, pop)
8               selection (pop)
9           end
10      end
11  end
```

Line 2 generates the initial population and line 3 evaluates this population using the fitness function FF_1. Then, between lines 4 and 11 there is a classical evolutionary process (reproduction and selection of individuals). This evolutionary process is stopped when is reached a number of generation, or when an individual reaches a value of $FF_1 = 0$ (this MTES represents a residue of the fault f_k).

$FF_1$is a multi-objective function fitness in order to find the MTES for the fault *f*. A weighted sum approach has been used to aggregate all the optimization criteria according to the equation below, gives a numerical value of the quality of each possible solution (MTES) of the optimization problem.

$$FF_1 = Min\left\{\sum_{n=1}^{3} w_n P_n\right\} \qquad (3)$$

Where, w_n is the corresponding weight and P_n is the criterion to be optimized, the different criteria to be optimized are in table 2.

Table 2 Optimization criteria

Objective	Optimization criteria
P_1	Number of Faults
P_2	Studied Fault
P_3	System Redundancy

P_1 ensures that only is studied one fault f_k at a given time: It is described by the sum of the active arcs of the faults that are in the individual, different of the faults to study. If this sum is different from zero is penalized the equation with a weight w_1.

$$P_1 = \sum_{\substack{f_i \in F \\ f_i \neq f_k}} \sum_{j=1}^{m} a_{kj} \qquad (4)$$

Where, $a_{kj}$are the arcs between the faults and the equations, $f_k$is a select fault and m is the number of equations.

P_2 ensures that the fault studied f_k is considered by the individual: it will make a sum of the active arcs from the studied fault presents in the individual. Normally, only one active arc from the studied fault must exist to guarantee a TES, otherwise this value is different from zero and the individual must be penalized by a weight w_2.

$$P_2 = \left|\sum_{j=1}^{m} a_{f_k j} - 1\right| \qquad (5)$$

Where, $a_{f_k j}$ are the arcs from the studied fault.

P_3verifies that the degree of redundancy of the model is equal to one, that is, there is one equation more than unknown variables. For that, it checks the cardinality of the active arcs for the case of the variables and equations, and this difference must be one to ensure the propriety of MSO and MTES. This function must also shed as a result the value of zero, otherwise the individual must penalized by weight w_3.

$$P_3 = \left|\left(Card(J^*) - Card(I^*)\right) - 1\right|_{\substack{I^*=\{i | \exists a_{ij}=1\ \forall\ j=1,m\} \\ J^*=\{j | \exists a_{ij}=1\ \forall\ i=1,n\}}} \qquad (6)$$

Where, I^* are the active arcs in unknown's variables and J^* are the active arcs in equations.

In this fitness function the importance of each criterion is defined by weight w_n. The values are determined according to the design requirements and experimentation see table 3.

Table 3 Optimization criteria

Objective	Optimization criteria	Optimized Value
w_1	Number of Faults	100
w_2	Faults in study	1000
w_3	System Redundancy	10000

Particularly, in GA1 three elite individuals (individuals with the best fitness values) of each generation are chosen in order to ensure that the current best individuals always survived in the next generation.

Theorem 1: GA1 ensures that if there is an individual with $FF_1 = 0$ it is a MTES of f_k, and the set of individuals with $FF_1 = 0$ will be the set of MTES of the studied fault f_k.

Proof of theorem 1; C_1 represents one individual in G with $FF_1 = 0$, that is a MTES sensitive a f_1

GA2

The resulting population from GA1 involves perhaps a large number of MTES. As there is a one to one relationship among the ARRs and MTESs, and if the detectability property is reached in the system, the line 10 of the Detection algorithm builds the signature matrix and calls GA2, in order to determine a set of ARR that guaranteeing the isolability property of the system, if there exist.

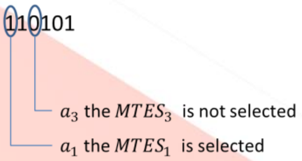


Figure 3. Bits string representation of the individual in GA2

An individual in the GA2 is also represented by a bits string. In this case, each bit represents each MTES (ARR) found by GA1 for all the faults, GA2 randomly defines individuals to be subsequently evaluated by the objective function, where a bit with value of "1 " means that a MTES is chosen in the fault signature matrix otherwise its value is "0" (see figure 3 for the example of the section 1). GA2 studies the sensitivity of the faults for the MTES's selected in a given individual, and observes if in this individual the isolability property is reached. The objective function of GA2 ensures that its value is zero because the individual has isolability property. Otherwise, this individual cannot isolate the faults (isolability property). There are two conditions in the objective function.

$$FF_2 = \sum_{i=1}^{\#f} \sum_{k=i+1}^{\#f} O_{ij}^1 + \sum_{i=1}^{\#f} O_i^2 \qquad (7)$$

Where O_{ik}^1 y O_i^2

$$O_{ik}^1 = \begin{cases} 0 & \text{if } \forall j = 1, \#ARRs\ \exists\, V_{ij} \neq V_{kj} \\ 10 & else \end{cases} \qquad (8)$$

$$O_i^2 = \begin{cases} 0 & \sum_{j=1}^{\#\,ARRs} V_{ik} \neq \emptyset \\ 10000 & else \end{cases} \qquad (9)$$

The first condition O^1 ensures that the fault signature vector f_i in the matrix constructed by the individual is different than the fault signature vector f_k, otherwise it penalizes with a low value, interpreting than just f_i cannot be isolate f_k. O^2 condition ensures that the sum of each vector in the fault signature matrix is non-zero, otherwise it cannot distinguish the behavior of a system with fault of a normal behavior, penalizing the individual with a high value.

```
Genetic Algorithm 2: GA2(S_m)
begin
    pop=generate_pop(S_m);
    % Generate initial population chosen some ARR from S_m
    Evaluation(FF_2, pop);
    % compute the fitness function to each individual
    while not FF_2 = ∅ or Nb_generation do
        begin
        % reproductive cycle begins
            Reproduction(pop)
```

```
            Evaluation (FF2, pop)
            selection (pop)
        end
    end
end
```

This GA generates the initial individuals using S_m, and the rest of the procedure is a classical GA. If we look at the individual shown in Figure 3, we can see that this individual selected the $MTES_1$, $MTES_2$, $MTES_4$ and $MTES_6$, the GA2 will check the signatures of each fault in the set of MTES selected using the objective function FF_2.

Theorem 2: GA2 ensures that if there is an individual with $FF_2 = 0$, it will get a set of MTES which accomplish with the isolability propriety

IV. EXPERIMENTS

In this section we are going to prove the different theorems. The *Main* algorithm invokes the *Detection* algorithm for the location of a population of sensitive MTES for each fault in the system, in order to reach its detectability property. In addition, it constructs the failure signature matrix with the minimum set of MTES to meet the isolability property in the system. The *Detection* algorithm invokes to GA1 and GA2 respectively, to perform this task. We are going to studied two problems, a theoretical example to describe our approach and a real continuous system.

A theoretical example to test our GA approach

For the experiment, we evaluate the example 1 that we have been using through this paper and structurally defined in section II.

Detection Problem: Evaluation of GA1

The initial population of GA1 is composed by fifty individuals. These individuals are sub-models of the main structural model, which is sufficient to fulfill the initial requirements for MTES. These individuals are evaluated using the fitness function FF_1, and in each generation the two best individuals are selected as parents, to execute the crossover operator with probability 0.7, and the mutation operator with probability 0.9. The stopping criterion is: if the individual with the best fitness value does not improve in 50 iterations AG1 stops. Table 4 and figure 4 are a representation of an individual $G = G_X \cup G_F$.

Table 4 Representation of the model in GA1

	G
G_X	[1,0,0;1,1,1;0,1,1;0,1,0;0,1,0;0,0,1];
G_F	[1,0,0,0,0;0,1,0,0,0;0,0,0,0,0;0,0,1,0,0;0,0,0,1,0;0,0,0,0,1];

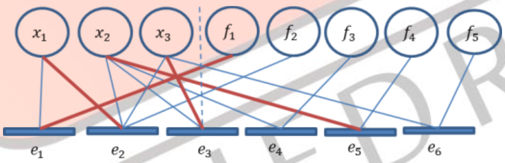


Figure 4, A possible individual of GA1

Graphically, the figure 4 shows an individual (C_1) in which the objective function FF_1 can be evaluated.

C_1: 010000000001000100010000000000000000000000000000000

This individual has, as shown in the figure 4, red lines (they are the active arcs between the vertices of equations and the vertices of the variables). In C_1 is represented by a value "1". If we evaluate de equations 4, 5 y 6.

$$P_1 = \sum_{\substack{f_i \in F \\ f_i \neq f_1}} \sum_{j=1}^{6} a_{kj} = 0 \qquad (10)$$

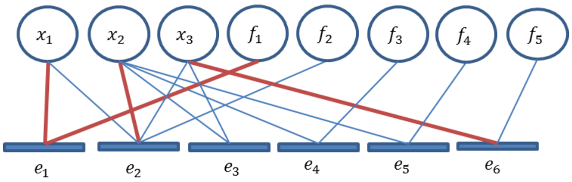
$$P_2 = \left| \sum_{j=1}^{6} 1 - 1 \right| = 0 \qquad (11)$$

$$P_3 = \left| \left(Card(3) - Card(4) \right) - 1 \right| = 0 \qquad (12)$$

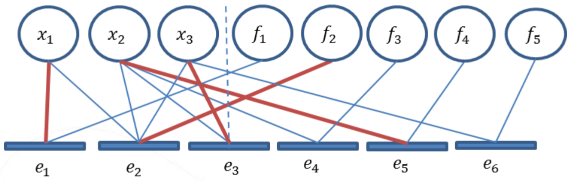
$$FF_1 = 100 * (0) + 1000 * (0) + 10000 * (0) = 0 \qquad (13)$$

P_1 means that there is only one active arc from the set of vertices belong to F. P_2 ensures that there is only one active arc from the studied fault. Finally the system redundancy is evaluated by P_3, where in the individual C_1 there are 4 equations and 3 variables with active arcs.

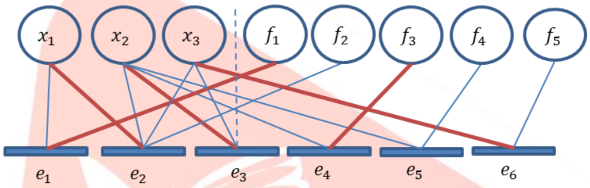
Now, we describe the behavior of our algorithm GA1 for other individuals. C_2 and C_3 are individuals which have two active arcs in the same vertex (see figure 5). In C_2, vertex of variables x_2 has two active arcs, and in C_3 e_2 has two active arcs. P_4 in FF_1 will penalize these individuals for this reason. On the other hand, P_3 will penalize C_2 and C_3 because the degree of redundancy of the individuals is different to one (cardinality constraint, see Table 5). In C_4, the active fault from the vertex of faults is not the fault in study, in this case the active arc is f_2. P_2 will penalize that in FF_1. In C_5, there are two active arcs from the vertex of faults (f_1 and f_3 are actives), P_1 will penalize this. Also P_3 penalizes the degree of redundancy, see Table 5.



C_2: 100000010000000001000000000000000000000000000000



C_3: 100000000010001000000000010000000000000000000000



C_4: 010000001000000001100000000000000000000100000000

Figure 5. Individuals of $GA1$

Table 5 Evaluation of the fitness function of GA1

Populations		**Evaluation of the fitness function**			
		P_1	P_2	P_3	*Value of* FF_1
1	C_1	0	0	0	0
2	C_2	0	0	10000	10000
3	C_3	0	1000	0	1000
4	C_4	100	1000	20000	21100

Finally, GA1 proposes a population of sensitive MTES for all considered faults. The table 6 shows that the algorithm finds nine MTES, fulfilling with the theorem 1.

Table 6 All MTES found by GA1

MTES for all considered faults	
MTES	[3 5 6]
	[3 4 6]
	[4 5]
	[4 6]
	[1 2 4 6]
	[1 2 5 6]
	[1 2 3 6]
	[1 2 3 5]
	[1 2 3 4]

Detection Problem: Evaluation of GA1

Continuing with the same example, we will evaluate the fault signature matrix showed in the table 1, to proof the theorem 2 (isolability of the model).

GA2 starts with a population of 20 individuals evaluated by the objective function FF2. Then, like GA1, the best individuals are chosen as parents in each generation to apply the crossover and mutation operators. The process to generate new individuals is performed carefully, because each individual must contain valid arcs among the possible arcs of the original bipartite graph. The stop criterion in this case is when the best individual does not improve after 20 iterations.

As we said before, an individual in GA2 is represented by a bit string. For this experiment we will evaluate tree individuals A_i, where 1 in an individual means that this ARR in the signature matrix is chosen.

A_1: 011100
A_2: 111111
A_3: 001011

In the case of A_1 only are selected $Arr_2,\ Arr_3, Arr_4$, different than A_2 where all the Arr are selected. In table 7 we will show the fitness function of each individual for the first condition. It will compare the signature of each fault.

Table 7 Evaluation of $O^1$of the individuals chosen.

O^1		Evaluation of the fitness function									
		f_1, f_2	f_1, f_3	f_1, f_4	f_1, f_5	f_2, f_3	f_2, f_4	f_2, f_5	f_3, f_4	f_3, f_5	f_4, f_5
1	A_1	10	10	0	0	0	0	0	0	0	10
2	A_2	10	0	0	0	0	0	0	0	0	0
3	A_3	10	0	0	0	0	0	0	0	0	0

The best result en O^1 is “0” that is when an individual has the isolability property. In this case none of the individual fulfill with this proprieties because the signature of f_1 and f_2 are similar. In Table 8 is studied the second condition.

Table 8 Evaluation of $O^1$of the individuals chosen

O^2		Evaluation of the fitness function				
		f_1	f_2	f_3	f_4	f_5
1	A_1	0	0	0	0	0
2	A_2	0	0	0	0	0
3	A_3	10000	10000	0	0	0

We can see that the sum of the values for $A_1$and A_3 are equal to 0, means that those individual are different of the model of the normal behavior. With respect to A_2, it has the same model of the normal behavior, that means the value is different to zero. Below is the result of the evaluation of the fitness function of GA2.

$$FF_2(A_1) = 30 + 0 = 30$$
$$FF_2(A_3) = 10 + 0 = 10$$
$$FF_2(A_2) = 10 + 20000 = 10010$$

According to theorem 2, for a full diagnosability, the value of FF_2must be equal to 0, all values more than 0 means that at least one fault is not isolable from other faults, and all the values more than 10000 means that there is a signature equal a the model in normal behavior, making no detectable the fault. Table 9 shows the final signature matrix as the result of GA2.

Table 8 Signature Matrix as the Result

	N	f_1	f_2	f_3	f_4	f_5
Arr_1	0	1	1	0	0	1
Arr_2	0	1	1	0	1	0
Arr_3	0	0	0	0	1	1
Arr_4	0	0	0	1	0	1

Analysis Of Diagnosability For A Gas Lift Well

In this section we present the method of production of oil by gas injection, the structural analysis of the process, and then apply our approach for fault diagnosis.

According to [19], the method of extraction of oil by gas injection is a method using compressed gas as energy source, for carrying the reservoir fluids from downhole to the surface; thus, the main consideration to select a group of oil wells, is the availability of a cost-effective source of high pressure gas.

In [17] indicates that gas lift is one of the primary methods used in the production of fluids from a well, which consists in the continuous injection of high pressure gas, to lighten the oil column in the production tubing. In other words, this method involves injecting gas at high pressure (through the compressor plant) at a preset rate, to lighten the column of oil, and thus improving the production of wells with reservoir pressure lower than the head at different depths. It is considered by experts as the most similar to the natural flow [18].

Gas from the annulus starts to flow into the tubing, as the gas enters into the tubing the pressure in the tubing falls, accelerating the inflow of gas lift. The gas pushes the major part of the liquid out of the tubing, while the pressure in the annulus falls dramatically. The annulus is practically empty, and the gas flow into the tubing is blocked by liquid accumulating in the tubing. Due to the blockage, the tubing becomes filled with liquid and the annulus with gas. Eventually, the pressure in the annulus becomes high enough for gas to penetrate into the tubing, and a new cycle starts.

In our approach, the diagnosability is developed based in residuals generation schemes derived from ARRs. Firstly, with the mathematical analysis described in this section, we identify equations governing the process defined in previous works [18, 19]. They are based on three state variables: x_1 is the mass of gas in the annulus, x_2 is the mass of gas in the tubing, and x_3 is the mass of oil in the tubing. With them, we can define the first equations set which represents the dynamics of the flow of each variable defined previously.

Example 4: Gas lift well model.

$$\dot{x_1} = w_{gc} - w_{iv}$$
$$\dot{x_2} = w_{iv} - w_{pg}$$
$$\dot{x_3} = w_r - w_{po}$$
$$w_{gc} = C_{iv}\sqrt{(P_{plp} + 14.7).P_{gldp}} + f_1$$
$$w_{iv} = C\sqrt{p_{a,inj}.(P_a - P_{t,inj})}$$
$$w_{pc} = C\sqrt{p_m.(P_t - P_s)} + f_2$$
$$w_{pc} = \frac{x_2}{x_2+x_3}.w_{pc}$$
$$w_{pc} = \frac{x_3}{x_2+x_3}.w_{pc}$$
$$w_{pc} = C_r.(P_r - P_{t,b}) + f_3$$
$$p_{a,inj} = \frac{M}{RT_a}.P_a$$
$$p_m = \frac{x_2+x_3-\rho_o L_r A_r}{L_t A_t}$$
$$P_a = \left(\frac{RT_a}{M_o} + \frac{gL_a}{V_a}\right)$$
$$P_t = \frac{M}{RT_a}.\frac{x_2}{L_r A_r + L_t A_t - x_3 v_o}$$
$$P_{t,inj} = P_t + \frac{g}{A_r}.(x_2 + x_3)$$
$$P_{t,b} = P_{t,inj} + \rho_o g L_r$$

Where w_{gc} is the mass flow rate of gas lift into the annulus, w_{iv} is the mass flow rate of gas lift from the annulus into the tubing, w_{pg} is the mass flow rate of gas through the production choke, w_r is the oil mass flow rate from the reservoir into the tubing, w_{po} is the mass flow rate of gas through the production choke, and w_{pc} is a mixed mass flow rate produced through the production choke. Unlike the model presented in [18, 19] w_{gc} is not considered as constant but rather calculated through the equation of flow by orifice plate denote by *e4*. This way, is checking the consumption of gas in each well. Also we consider two more variables: P_{glp} is the pressure in the system distribution of gas, and P_{gldp} is the differential pressure of gas through an orifice plate.

C_{iv}, C and C_r are constants, $\rho_{a,inj}$ is the density of gas in the annulus at the injection point, ρ_m is the density of the oil/gas mixture at the top of the tubing, P_a is the pressure in the annulus at the injection point, $P_{t,inj}$ is the pressure in the tubing at the gas injection point, P_t is the pressure at the top of the tubing, P_s is the pressure at the separator, P_r is pressure in the reservoir, and $P_{t,b}$ is the pressure at the bottom of the tubing. The reservoir pressure, P_r, is assumed to be slowly varying, and therefore treated as constant. Note that flow rates through the valves are restricted to be positive. M_o is the molar weight of the gas, R is the gas constant, T_a is the temperature in the annulus, T_t is the temperature in the tubing, V_a is the volume of the annulus, L_a is the length of the annulus, L_t is the length of the tubing, A_t is the cross-sectional area of the tubing above the injection point, L_r is the length from the reservoir to the gas injection point, A_r is the cross-sectional area of the tubing below the injection point, g is the gravity constant, ρ_o is the density of the oil, and, v_o is the specific volume of the oil.

Gas lift well analysis with our GA Approach

To comply with the technique described in the previous section, to detect and isolate faults that are really relevant for us in a real-world process of extraction of oil by gas injection, is necessary associate the faults in the equation where occurs. For this reason, f_1, f_2, f_3, f_4, f_5 are added to the equations of the model, f_1 which is the fault in the flow of gas injected into the annular, f_2 which is the fault in the mixed flow to the separator in the production line, f_3 is the fault in the mixed fluid into the tubing, f_4 is the fault in pressure at the bottom of the tubing, and f_5 is the fault in the tubing at the gas injection point. Table 10 shows the new individual composed by GX with a length of 456 bits, and GF with a length of 120 bits.

Table 10 Representation of the structural model

	G
G_X	[1,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0;0,1,0,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0;0,0,1,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0;0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,1,0;0,0,0,0,0,1,0,0,0,1,1,1,0,0,0,0,0,0,0;0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,1,0,1;0,1,1,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0;0,1,1,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0;0,0,0,0,0,0,0,0,1,0,0,0,0,1,1,0,0,0,0;0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0;0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1;1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0;0,1,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0;0,1,1,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0;0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,0,0;1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0;0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0;0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0;0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0;0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0;0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0;0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1;0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0]
G_F	[0,0,0,0,0;0,0,0,0,0;0,0,0,0,0;1,0,0,0,0;0,0,0,0,0;0,1,0,0,0;0,0,0,0,0;0,0,0,0,0;0,0,0,0,0;0,0,0,0,0;0,0,0,0,0;0,0,0,0,1;0,0,0,1,0;0,0,0,0,0;0,0,1,0,0;0,0,0,0,0;0,0,0,0,0;0,0,0,0,0;0,0,0,0,0;0,0,0,0,0;0,0,0,0,0;0,0,0,0,0;0,0,0,0,0;0,0,0,0,0];

The results of GA1 are shown in Table 11 for a group of 30 runs. GA1 gets a population of sensitive MTES for each considered fault. This is a very important result of our proposal, since it allows a wide analysis of diagnosability of a system. Subsequently, the GA2 determines a fault signature matrix with the isolability property of the complete system, which is shown in Table 12, choosing a MTES for each fault (which cannot be obtained in [12]).

Table 11 Population MTES achieved by AG1 for the five faults

	Population of MTES
MTES for $\boldsymbol{f_1}$	[1,2,3,4,5,7,8,9,11,14,18,19,20,21]
	[1,2,3,4,5,7,8,10,11,14,19,20,21,23]
	[3,4,5,7,8,9,11,14,16,17,18,19,20,21,22,24]
	[1,2,3,4,5,7,8,9,11,14,16,17,19,20,21,22,23,24]
MTES for $\boldsymbol{f_2}$	[1,2,3,5,6,7,8,10,11,14,16,17,18,19,23,24]
	[1,2,3,5,6,7,8,9,10,11,14,16,17,18,19,20,23,24]
	[5,6,10,11, 14,23,24]
MTES for $\boldsymbol{f_3}$	[1,2,3,5,7,8,9,10,11,14,15,16,17,18,19,20,21,22,23,24]
	[3,5,9,10,11,15,22,23]
MTES for $\boldsymbol{f_4}$	[1,2,5,7,8,9,10,11,13,14,16,17,18,19,21,23,24]
	[1,2,3,5,7,9,10,11,13,14,16,17,18,19,20,21,23]
MTES for $\boldsymbol{f_5}$	[1,2,5,7,8,9,11,12,14,16,17,21]
	[1,2,3,5,7,10,12,14,16,17,19,22,23]

In general, we see how our approach is able to generate a wide-sensitive population of MTES, to a fault or combinations of them (GA1), enabling better analysis of the property of detectability. Subsequently, based on this population, it is able to perform a thorough analysis of the isolability property in the system. Specifically, it determines the optimal set of MTES to verify that property (GA2) in the system (when it is available).

The techniques in the literature do not allow that [7, 8], particularly because obtain MTES is a combinatorial problem that our approach can solve very well (see GA1). This new population of MTES gives ample possibilities to build the fault signature matrix that reaches the isolability property in the system (see GA2), which does not happen in the others proposed in the literature, by the limited number of MTES that they obtained.

Table 12 Population MTES achieved by AG1 for the five faults

	N	$\boldsymbol{f_1}$	$\boldsymbol{f_2}$	$\boldsymbol{f_3}$	$\boldsymbol{f_4}$	$\boldsymbol{f_5}$
Arr_1	0	1	0	0	0	0
Arr_2	0	0	1	0	0	0
Arr_3	0	0	0	1	0	0
Arr_4	0	0	0	0	1	0
Arr_5	0	0	0	0	0	1

V. Conclusions

The capability to detect a fault on time in a system provides security, availability and reliability. The fault diagnosis mechanisms used in this paper is based on the principles of redundancy. This paper analyzed the technique of structural analysis, in order to propose a hybrid approach for the diagnosability and sensor placement problems for continuous processes. Our approach, obtains a structural model based on the dynamic process model based in GA.

In this work, we use several GAs to solve our problem. A first GA has a population of bipartite graphs, the result is the set of MTES to detect all the faults in the system. The second GA allows reach the isolability property, selecting the minimum required population of MTES in the fault signature matrix.

Our approach is an efficient tool to solve the combinatorial problem behind of the determination of MTESs, given a structural model as a bipartite graph. Additionally, it determines where new sensors must be placed, in order to reach the diagnosability property. That means, our approach carries out several studies in an automatic way: determination of the detectability and isolability of the faults of a system (diagnosability property). We have shown that GA's is a powerful tool for providing the answers to these questions. The execution time of the GAs can be controlled with their parameters, in order to obtain good solutions in a short computing time.

Particularly, our GA1 studies the detectability property based on the structural analysis of the studied system. It is a hard optimization problem, because there are a lot of alternative paths in a system, which increase according to its number of variables and equations. These paths must be analyzed to determine the detectability property using the structural analysis approach. The GA2 studies the isolability property analyzing the fault signature matrix, in order to determine the MTES necessary to reach it. This process of selection of MTES is the optimization problem solves by GA2. If these properties are not reached, the diagnosability property is not reached. These different optimization problems are very well solved with ours GAs; additionally, theirs execution times can be controlled through theirs parameters (number of generations, numbers of individuals, etc.).

Future work will combine this approach with a sensor placement approach, in order to reach the diagnosability in the cases where this property is not reached according to the approach proposed in this work.

VI. Acknowledgment

This work has been supported by Postgraduate Cooperation Program (PCP) between Venezuela and France entitled "Supervision and maintenance task in a shared organizational environment", contract No. 2010000307, between the University of Los Andes, Venezuela with the cooperation of LAAS- CNRS, University of Toulouse, France.

References

[1] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki, "Diagnosis and fault tolerant control". Berlin: Springer 2003.

[2] M. Krysander, J. Åslund, and M. Nyberg, An efficient algorithm for finding minimal over-constrained subsystems for model-based diagnosis, IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans, 38(1), pp 197-206, 2008.

[3] S. Indra, L. Travé-Massuyès, and E. Chanthery. A decentralized FDI scheme for spacecraft: Bridging the gap between model based FDI research & practice, IFAC (SAFEPROCESS 2012), pp.283-288, 2012.

[4] J. Armengol, A. Bregon, T. Escobet, E. Gelso, M. Krysander, M. Nyberg, X. Olive, B. Pulido, and L. Travé-Massuyès, "Minimal structurally overdetermined sets for residual generation": A comparison of alternative approaches, In Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, pages 1480–1485, 2009

[5] M. Staroswiecki and P. Declerck. "Analytical redundancy in nonlinear interconnected systems by means of structural analysis". In Proceedings of IFAC AIPAC'89, pp 51–55, 1989.

[6] L. Travé-Massuyès, T. Escobet, and X. Olive. "Diagnosability analysis based on component supported analytical redundancy". IEEE Trans. on Systems, Man, and Cybernetics Part A: Systems and Humans, 36(6), pp 1146–1160, 2006.

[7] M. Krysander, J. Åslund, and E. Frisk, "A structural algorithm for finding testable sub-models and multiple fault isolability analysis", In Proceedings of the 21st International Workshop on Principles of Diagnosis (DX-10), 2010.

[8] C. Svard and M. Nyberg, "Residual generators for fault diagnosis using computation sequences with mixed causality applied to automotive systems", IEEE Transactions on Systems Man and Cybernetics Part A, 40(6), pp 1310–1328, 2010.

[9] L. Travé-Massuyès, T. Escobet, and R. Milne. "Model-based diagnosability and sensor placement. application to a frame 6 gas turbine subsystem." In DX01 twelfth international workshop on principles of diagnosis, pp 205–212, 2001.

[10] M. Nyberg,."Automatic design of diagnosis systems with application to an automotive engine". Proceedings Control Engineering Practice, 87(8): pp 993–1005, 1999.

[11] C. Svärd and M. Nyberg "A Mixed Causality Approach to Residual Generation Utilizing Equation System Solvers and Differential-Algebraic Equation Theory" Technical Report Nr. LiTH-ISY-R-2854, Department of Electrical Engineering, Linköpings Universitet, 2008.

[12] M. Krysander and M. Nyberg. "Structural analysis utilizing MSS sets with application to a paper plant". Proceedings of the Thirteenth International Workshop on Principles of Diagnosis, 2002.

[13] V Puig, J. Quevedo, T. Escobet and B. Pulido, "On the integration of detection and isolation in model based fault diagnosis". Proceeding In DX04, 15th International Workshop on Principles of Diagnosis, 2004.

[14] S. Sen, S. Narasimhan and K. Deb, "Sensor network design of linear processes using genetic algorithms", Comput. Chem. Eng. 22 (3), pp. 385–390, 1998.

[15] S. Jin, M. Zhou and A.S. Wu, "Sensor network optimization using a genetic algorithm", in: 7th World Multiconference on Systemics, Cybernetics, 2003.

[16] A. Bhondekar, R. Vig, M. LalSingla, C. Ghanshyam and P. Kapur. "Genetic Algorithm Based Node Placement Methodology For Wireless Sensor Networks" Proceedings of the International MultiConference of Engineers and Computer Scientists 2009 Vol I IMECS, 2009.

[17] R. Leal, E. Camargo, J. Aguilar, A. Ríos and L. Travé "Analysis of diagnosability for gas lift Wells", Ingeniería y Ciencias Aplicadas: Modelos Matemáticos y Computacinales (Ed. E. Dávila, J. Del Río, M. Cerrolaza, R. Chacón), Sociedad Venezolana de Métodos Numéricos en Ingeniería, pp. EC91-EC97, 2014.

[18] G. Hernández, E. Alí and T. Sharon. Manual del curso de levantamiento artificial por gas avanzado. Segunda edición. PDVSA Intevep, Venezuela. 2001.

[19] L. Imsland, "Topics in nonlinear control: Output feedback stabilization and control of positive systems". Ph.D. Thesis, NTNU, Norway, 2002.

[20] G.O. Eikrem, O.M. Aamo, H. Siahaan and B. Foss, "Anti-slug control of gas-lift wells experimental results". Proceedings volume from the 6th IFAC, 2004.