INTERVAL MODEL-BASED DIAGNOSIS USING CONSTRAINT PROGRAMMING

R. CEBALLOS , R. M. GASCA, C. DEL VALLE, AND M. TORO
Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla
Escuela Técnica Superior de Ingeniería Informática,
Avenida Reina Mercedes s/n 41012 Sevilla(Spain)

ABSTRACT

In engineering applications many models are based on constraints with interval parameters and variables. The model is based on the knowledge of the behavior of the system to diagnose. Inputs and outputs of components are represented as variables in the constraints, and they can be observable and non-observable depending on the situation of the sensors in the system.
In this work, we propose a new approach to automate the determination of the minimal diagnosis. This approach has two phases. In the first phase, we determine components clusters in the system in order to reduce drastically the number of components to consider. This is specially necessary in high density systems where components compose independent sets in themselves. In the second phase, we construct a constraint satisfaction diagnosis problem. In this phase we use interval variables (based on the domain of the variables). The results obtained in the studied cases are very promising.
KEYWORDS: constraint satisfaction problem, constraints, clusters, diagnosis, interval models.

1. INTRODUCTION

Diagnosis allows to determine why a system correctly designed does not work like it was expected. It is based on the monitoring of a system, using sensors which are integrated and supposed to work correctly. The diagnosis aim is to detect and to identify the reason of the unexpected behavior, or in other words, to identify the parts which fail in a system. In order to explain a wrong behavior, the diagnosis process uses a determined set of observations and a model of the system. These faults has to be avoided if we want to keep a system within the desired production and security level. Two communities works in parallel and usually separated in diagnosis: FDI (Automatic Control) and DX (Artificial Intelligence). Nevertheless, the integration of FDI and DX theories (BRIDGE Task Group) has been shown in recent works (as [Cordier00] and [Gasca03]).

Both communities are based on the use of models. In the area of DX, the first work related to diagnosis was presented with the aim of identifying faults in the component systems, based on the structure and its behavior [Reiter84]. DART [Genesereth84] and GDE [de Kleer87] were the first implementations to perform diagnosis, both detect possible faults using different inference mechanisms. In [Reiter87] and [de Kleer92] a general theory was proposed for the problem of explaining the discrepancies between the observed and correct behavior that the mechanisms subject to the diagnosis process (logical-based diagnosis) have. These two paper presented the diagnosis formalization.

In this work, we propose a new approach to automate and to improve the determination of interval model-based diagnosis . This work is based in two steps:

- A structural pre-treatment in order to reduce drastically the computational complexity, specially in high density systems where components compose independent sets in themselves.
- A Maximization Constraint Satisfaction Problem (CSP) for modelling and solving the diagnosis problem as a set of constraints among interval variables. The diagnosis aim is to find what constraints are not satisfied and therefore must be modified. Constraint programming has the capability to solve systems of linear and polynomial equations and inequalities.

Many techniques exploit the topological structure of the system using a problem's constraint graph. For example, in [Krysander02], in order to reduce the computational complexity, they proposes a two-step approach: First, the system is analysed to find overdetermined submodels, and then, all of these submodels are transformed to consistency relations.

A Constraint Satisfaction Problem (CSP) is a framework for modelling and solving real-problems as a set of constraints among variables. A CSP is defined by a set of variables $X=\{X_1,X_2...,X_n\}$ associated with a set of discrete-valued, $D=\{D_1,D_2,..., D_n\}$ (where every element of D_i is represented by set of v_i), and a set of constraints $C=\{C_1,C_2,...,C_m\}$. Each constraint C_i is a pair (W_i,R_i), where R_i is a relation $R_i \subseteq D_{i1}x...xD_{ik}$ defined in a subset of variables $W_i \subseteq X$. If we have a CSP, the Max-CSP aim is to find an assignment that satisfies most constraints, and minimizes the number of violated constraints. The diagnosis aim is to find what constraints are not satisfied and therefore must be modified. The solutions searched with Max-CSP techniques is very complex. Some investigations have tried to improve the efficiency of this problem, [Kask00] and [Larrosa99].

The constraint programming(CP) is a paradigm with the capability to solve CSP. CP has been proposed in order to diagnose analog circuit [Mozetic93]. They show how analog circuits can be modelled using this paradigm. The tool used was CLP($\Re$). One of the restrictions of this approach is the single fault assumption, this is due to the limitation of the CLP($\Re$) to linear constraints. Another approach for diagnosis of analog circuits is presented using the CP and interval arithmetic [Fuentes03]. They uses combined information from tests at different frequencies

Our paper has been organized as follows. In section 2 we will show two examples, the simple problem example and the six heat exchangers example. In section 3 it appears definitions and notation in order to clarify concepts for our approach. In section 4, we present the structural pretreatment and their usefulness. Then in section 5, we give a description of the constraint satisfaction problem and how to solve it. Finally, conclusions and future works are presented.

2. EXAMPLES

In order to explain the methodology, we will use the following systems that are very often-used examples in the bibliography concerning model-based diagnosis.

2.1 A Well-Known Example: Simple problem

A very often used example in the bibliography concerning model-based diagnosis [Reiter84] and [de Kleer92] is the one formed by three multipliers and two adders, as it is presented in figure 1. The multipliers are represented in figure 1 as M_1, M_2 and M_3, and the adders as A_1 and A_2. In this system, the component or components that are failing have to be identified, taking into account that the only observable values are the ones represented as a, b, c, d, e, f and g.

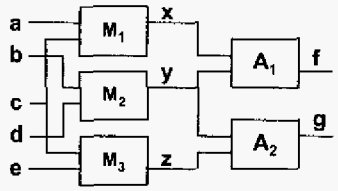

Figure 1. Circuit formed by three multipliers and two adders

2.2 A System of Heat Exchangers

This system proposed in [Guernez97], consists of six heat exchangers, three flows f_i come in at different temperatures t_i. This example defines three different subsystems, each one formed by two exchangers: E_1, E_2, E_3, E_4, E_5 and E_6. Each of the six exchangers and each of the eight nodes of the system are considered as components to verify their correct functioning. The normal functioning of the system can be described by means of polynomial constraints, coming from three kinds of balances:

$$\sum_i f_i = 0 : \text{mass balance at each node,}$$

$$\sum_i f_i \cdot t_i = 0 : \text{thermal balance at each node,}$$

$$\sum_{in} f_i \cdot t_i - \sum_{out} f_j \cdot t_j = 0 : \text{enthalpic balance for each heat exchanger.}$$

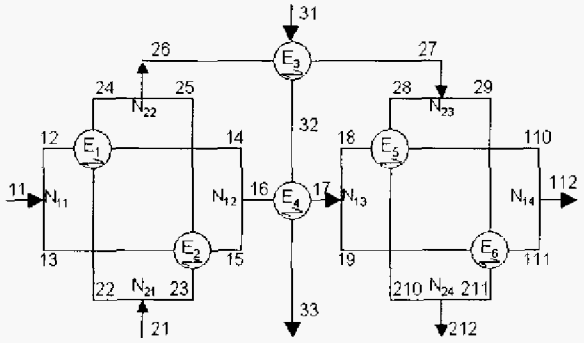

Figure 2. System of heat exchangers

3. DEFINITIONS AND NOTATION

In order to clarify the diagnosis process we need to expose some definitions and notation. Model-based diagnosis requires a system model which represents the behavior of the system and each model component. In our case, we will only deal with the case which has a model of system constraints that derives from its own structure, and which has links between components (structural model) and the behavior of each model component.

Definition 1. A System of Components (COMPS): It's the set of all the components of the system.

Definition 2. Monitored Precision ($\pm \Delta_{MONITORED}$): It's the accuracy that we have when we monitored inputs or outputs of components. If we know exactly the value of a monitored point, then $\Delta_{MONITORED}=0$, but this is very difficult in real systems.

Definition 3. Component Precision ($\pm \Delta_{COMPONENT}$): It's the accuracy of the behaviour of a system component. If the component works exactly like it's supposed in its model, then $\Delta_{COMPONENT} \equiv 0$, but this is very difficult in real components.

Definition 4. The **System Description** (SD): It can be defined as a finite set of polynomial equality constraints (P) which determine the system behaviour. This is done by means of the relations between the system non-observable variables (V_{nob}) and the system observable variables (V_{ob}) which are directly obtained from sensors that are supposed to work correctly. Then, the following tuple for a system description is obtained SD (P, V_{ob} ,V_{nob}).

Some polynomial equality constraint has assigned a component precision, depending on the type of component. For example, in the example proposed in section 2.1 we have supposed that the components work correctly but with a component precision. In this example we studied two different kind of component precision, for the multipliers the component precision is named $\Delta_{C\text{-}MULT}$, and for the adders the component precision is named $\Delta_{C\text{-}ADD}$. In the example proposed in section 2.2 we supposed that it is not necessary anyone component precision. In tables 1 and 2 appears the system description for the examples proposed in section 2.

Definition 5. Observational Model (OM): A tuple that assigns values to the observable variables. Every observable variable has assigned a monitored precision ($\pm \Delta_{MONITORED}$) which represent the accuracy that we have when we monitored this variable. If we know exactly the value of a monitored point, then $\Delta_{MONITORED}=0$, but this is very difficult in real systems. In the first example there exists only one kind of component precision, for monitored signal($\Delta_{M\text{-}SIGN}$) at each input or output of every component. In the Heat Exchangers example there exists two different kind of component precision, for monitored temperature ($\Delta_{M\text{-}TEMP}$) at each node or heat exchanger, and for monitored flow (Δ_{M_FLOW}) at each node or heat exchanger.

System Description			
Component.	Constraints	Component.	Constrains
M_1	$\|x-a*c\| \leq \Delta_{C\text{-}MULT}$	A_1	$\|f-x+y\| \leq \Delta_{C\text{-}ADD}$
M_2	$\|y-b*d\| \leq \Delta_{C\text{-}MULT}$	A_2	$\|g-y+z\| \leq \Delta_{C\text{-}ADD}$
M_3	$\|z-c*e\| \leq \Delta_{C\text{-}MULT}$		
V_{ob}	a, b, c, d, e, f, g		

Table 1. Simple Problem System Description.

System Description			
C.	Constraints	C.	Constrains
N_{11}	$f_{11}+f_{12}+f_{13}=0$ $f_{11}*t_{11}+f_{12}*t_{12}+f_{13}*t_{13}=0$	N_{21}	$f_{21}-f_{22}-f_{23}=0$ $f_{21}*t_{21}-f_{22}*t_{22}-f_{23}*t_{23}=0$
N_{12}	$F_{14}+f_{15}-f_{16}=0$ $f_{14}*t_{14}+f_{15}*t_{15}-f_{16}*t_{16}=0$	N_{22}	$f_{24}+f_{25}-f_{26}=0$ $f_{24}*t_{24}+f_{25}*t_{25}-f_{26}*t_{26}=0$
N_{13}	$F_{17}-f_{18}-f_{19}=0$ $f_{17}*t_{17}-f_{18}*t_{18}-f_{19}*t_{19}=0$	N_{23}	$f_{27}-f_{28}-f_{29}=0$ $f_{27}*t_{27}-f_{28}*t_{28}-f_{29}*t_{29}=0$
N_{14}	$f_{110}+f_{111}-f_{112}=0$ $f_{110}*t_{110}+f_{111}*t_{111}-f_{112}*t_{112}=0$	N_{24}	$f_{210}+f_{211}-f_{212}=0$ $f_{210}*t_{210}-f_{211}*t_{211}-f_{212}*t_{212}=0$
E_1	$f_{12}-f_{14}=0$ $f_{22}-f_{24}=0$ $f_{12}*t_{12}-f_{14}*t_{14}+f_{22}*t_{22}-f_{24}*t_{24}=0$	E_2	$f_{13}-f_{15}=0$ $f_{23}-f_{25}=0$ $f_{13}*t_{13}-f_{15}*t_{15}+f_{23}*t_{23}-f_{25}*t_{25}=0$
E_3	$f_{26}-f_{27}=0$ $f_{31}-f_{32}=0$ $f_{26}*t_{26}-f_{27}*t_{27}+f_{31}*t_{31}-f_{32}*t_{32}=0$	E_4	$f_{16}-f_{17}=0$ $f_{32}-f_{33}=0$ $f_{16}*t_{16}-f_{17}*t_{17}+f_{32}*t_{32}-f_{33}*t_{33}=0$
E_5	$f_{18}-f_{110}=0$ $f_{28}-f_{210}=0$ $f_{18}*t_{18}-f_{110}*t_{110}+f_{28}*t_{28}-f_{210}*t_{210}=0$	E_6	$f_{19}-f_{111}=0$ $f_{293}-f_{211}=0$ $f_{19}*t_{19}-f_{111}*t_{111}+f_{29}*t_{29}-f_{211}*t_{211}=0$
V_{ob}	$f_{11}, f_{12}, f_{13}, f_{16}, f_{17}, f_{18}, f_{19}, f_{112}, f_{21}, f_{26}, f_{27}, f_{212}, f_{31}, f_{33}, t_{11}, t_{12}, t_{13}, t_{16}, t_{17}, t_{18}, t_{19}, t_{112}, t_{21}, t_{26}, t_{27}, t_{212}, t_{31}, t_{33}$		

Table 2. Heat Exchangers. System Description.

Definition 6. Constraint Satisfaction Diagnosis Problem (CSDP): It can be defined by means of a CSP problem formed by a System Description (SD) and Observational Model (OM). The result of this CSP problem will be a set of elements that belong to the set of the system faults which reflect in a minimal way the information of the possible failing components.

To obtain these components, it is necessary to define the notion of abnormal (AB): AB(c) is a predicate which holds when component $c \in$ COMPS is abnormal. The model for the correct behavior of a component is written as if the component is not abnormal

Definition 7. Diagnosis: The diagnosis for a CSDP will be a set of components $\Delta \subseteq$ COMPS that verifies SD $\cap$ OM $\cap$ $\{AB(c) \mid c \in \Delta\} \cap \{\neg AB(c) \mid c \in$ COMPS-$\Delta\}$[de Kleer87]. There may be an exponential number of diagnoses (2^{COMPS}).

Definition 8. Minimal diagnosis: A diagnosis Δ is minimal if $\forall \Delta' \subset \Delta$, Δ' is not a diagnosis[de Kleer87].

4. STRUCTURAL PRETREATMENT: IDENTIFICATION OF COMPONENTS CLUSTERS

The first step is to isolate independent subsystems. This structural pretreatment will give us a partition of the system into independent subsystems. The independence between subsystems guarantees us that the minimal diagnosis of the system can be obtained with the minimal diagnosis of all independent subsystems. The subsystems obtained are much smaller than the whole system, and therefore the computational complexity to detect conflicts from each subsystem is lower compared to the whole system. The partition of the system guarantees a smaller computational cost.

Definition 9. Components cluster (CC): A set of components C belong to the total system is a components cluster, if the following predicates are true:
- For all non-observable inputs and outputs of each component of C, these inputs and outputs are always linked with only components of C.
- It does not exist another set of components C' with less elements than C which validates the first predicate and it is include in C.

With the first predicate we look for the independence between conflicts of different components clusters. This predicate guarantees us that we are able to detect a minimal diagnosis in a components cluster without information about other components clusters. This is possible because in a components cluster all the non-observable inputs and outputs are between components of the same cluster, and therefore, there is not anyone connection with other component which is not monitored. Each components cluster is a set of components where we can detect conflicts.

We look for to divide our system in the biggest possible number of subsystems in order to obtain a smaller computational cost. The second predicate guarantees us that the sets will be as small as it is possible, because it prevents that a set of components (components cluster) is composed of two or more independent sets of components. In this predicate we guarantee that a set C' will not exist inside C, because if C' exists, then another independent set C'' with components $C \setminus C'$ could exist.

Example: For example in the heat exchangers problem the component E_3 is not completely monitored because we are not able to know the value of outputs f_{32} and t_{32}. Likewise, E_4 is not completely monitored because we are not able to know the value of inputs f_{32} and t_{32}. But we can monitored these two component if we think in these two components as if they were a subsystem, with the same observable inputs and outputs that they have separately.

Algorithm: The following pseudo-code (see figure 3) defines the function *clustersIdentification(C)* which takes C, the set of component of all the system, and returns A, the set of components clusters. The algorithm previously will store into the set E all pairs of components which have an in common non-observable variable. The algorithm begins creating as many sets as n, where n is the number of components of the system. All these sets have one component. Then, for each element of E, which is a connection between two components $x \in S_1$ and $y \in S_2$, where S_1 and $S_2 \in A$, the algorithm merges sets S_1 and S_2. When the process is finished all components have assigned one components cluster.

Auxiliary function of the algorithm:
- nonObsVar(x): This function returns the set of non-observable variables of a component x.

For the example presented in section 2.2, we obtained five components clusters, which are $A=\{\{N_{11}\},\{N_{13}\},\{N_{12},N_{21},N_{22},E_1,E_2\},\{N_{14},N_{23},N_{24},E_5,E_6\},\{E_3,E_4\}\}$. And, for the example presented in section 2.1, we obtained only one components clusters, which is $A=\{\{M_1,M_2,M_3,A_1,A_2\}\}$

```
clustersIdentification(C) return A
  E = {}
  A = {}
// Detect all connections between components
  foreach x ∈ C
    foreach y ∈ C
      if x ≠ y ∧ nonObsVar(x) ∩ nonObsVar(y) ≠ {}
        E = E ∪ {{x,y}}
      endif
    endforeach
  endforeach
// Generate clusters with only one component
  foreach x ∈ C
    A = A ∪ {{x}}
  endforeach
// Detect all components clusters
  foreach {x,y} ∈ E
    if ∃ S1,S2 | S1 ∈ A ∧ S2 ∈ A ∧ S1 ≠ S2
          ∧ x ∈ S1 ∧ y ∈ S2
      A = A \ S1
      A = A \ S2
      A = A ∪ {S1 ∪ S2}
    endif
  endforeach
```

Figure 3. ClusterIdentification Algorithm

5. SOLVING THE CSDP

For everyone subsystem obtained if section 4, we build a different and independent CSDP. This structural pretreatment guarantees a smaller computational cost, because the subsystems obtained are much smaller than the whole system.

In a CSP problem the goal is to satisfy all constraints. But in many problems this is impossible. The goal in this kind of problems (Max-CSP, Maximization Constraint Satisfaction Problem) is to satisfy the most bigger number of constraints. To obtain this goal, we have to define a goal function that a solver have to maximize while is looking for a solution for the CSP problem.

In order to obtain the goal function, we have to define all the constraint that can be not satisfied like reified constraints. A reified constraint has associated a boolean value (called reified variable) which stored if the constraint is satisfied or not. The number of reified variables that are true will be the goal function that we want to maximise. We will use the predicated AB(c) as a reified variable that stored if the component *c* work correctly.

In table 3 appears the CSDP for the example presented in section 2.1, and in table 4 appears the CSDP for the components cluster 3 ($\{N_{12}, N_{21}, N_{22}, E_1, E_2\}$) of the example presented in section 2.2. Like it appears in tables 3 and 4, a component fails if someone of its constraints is not satisfied. With the OM we can define the domains of observable variables, but non-observable variables are free (like the predicate AB applied to components).

Our objective is to find minimal diagnosis. This implies that our objective is to maximize the number of predicates AB(c) which appear as true, where c are components of the components

cluster. The CSP problem have many solutions, but we are interesting in the solutions which implies to change the minimal number of components. For example, in the example of the section 2.1, firstly we look for solutions that allow us to obtain a correctly work of the system with only one component change. In order to find firstly these solutions, we solve this problem like a Max-CSP problem. Both of them, CSDP and function to maximize, constitute a Max-CSP problem. Solving these Max-CSP problem we will obtain the values of the predicate AB(c) for every component *c*. The false value of these predicates define the set of components that constitutes the diagnosis, and also, this set has the minimal cardinality. Then we look for the diagnosis changing two components, three components…or more components. To implement this search of solutions we used ILOG-Solver TM tool (Constraint Library of commercial C++ [Ilog02])

Constraints	Comp.	Constraints
	M_1	$\neg AB(M_1) = [\lvert x-a*c \rvert \leq \Delta_{C\text{-}MULT}]$
	M_2	$\neg AB(M_2) = [\lvert y-b*d \rvert \leq \Delta_{C\text{-}MULT}]$
	M_3	$\neg AB(M_3) = [\lvert z-e*e \rvert \leq \Delta_{C\text{-}MULT}]$
	A_1	$\neg AB(A_1) = [\lvert f-x+y \rvert \leq \Delta_{C\text{-}ADD}]$
	A_2	$\neg AB(A_2) = [\lvert g-y+z \rvert \leq \Delta_{C\text{-}ADD}]$
Domains	AB	$AB(M_1), AB(M_2), AB(M_3), AB(A_1), AB(A_2) = \{true, false\}$
	V_{nobs}	$x, y, z = \{free\}$
	V_{obs}	a=3, b=2, c=2, d=3, e=3, f=10, g=12
	Δ	$\Delta_{C\text{-}MULT} = 0.1$, $\Delta_{C\text{-}ADD} = 0.1$, $\Delta_{M\text{-}SIGN} = 0.1$
Goal Function	Max ($\mathcal{N}$ c : c ∈ $\{M_1, M_2, M_3, A_1, A_2\}$:¬AB(c) = true)	

Table 3. CSDP for the Simple Problem

Constraints	Comp.	Constraints
	N_{12}	$\neg AB(N_{12}) = [\ (f_{14}+f_{15}-f_{16}=0) \wedge (f_{14}*t_{14}+f_{15}*t_{15}-f_{16}*t_{16}=0)\ ]$
	N_{21}	$\neg AB(N_{21}) = [\ (f_{21}-f_{22}-f_{23}=0) \wedge (f_{21}*t_{21}-f_{22}*t_{22}-f_{23}*t_{23}=0)\ ]$
	N_{22}	$\neg AB(N_{22}) = [\ (f_{24}+f_{25}-f_{26}=0) \wedge (f_{24}*t_{24}+f_{25}*t_{25}-f_{26}*t_{26}=0)\ ]$
	E_1	$\neg AB(E_1) = [\ (f_{12}-f_{14}=0) \wedge (f_{22}-f_{24}=0) \wedge (f_{12}*t_{12}-f_{14}*t_{14}+f_{22}*t_{22}-f_{24}*t_{24}=0)\ ]$
	E_2	$\neg AB(E_2) = [\ (f_{13}-f_{15}=0) \wedge (f_{23}-f_{25}=0) \wedge (f_{13}*t_{13}-f_{15}*t_{15}+f_{23}*t_{23}-f_{25}*t_{25}=0)\ ]$
Domains	AB	$AB(N_{12}), AB(N_{21}), AB(N_{22}), AB(E_1), AB(E_2) = \{true, false\}$
	V_{nobs}	$f_{14}, f_{15}, f_{22}, f_{23}, f_{24}, f_{25}, t_{14}, t_{15}, t_{22}, t_{23}, t_{24}, t_{25} = \{free\}$
	V_{obs}	f_{16}=95, f_{21}=100, f_{26}=100, f_{12}=50, f_{13}=50, t_{16}=45, t_{21}=60, t_{26}=45, t_{12}=30, t_{13}=30
	Δ	$\Delta_{M\_FLOW} = 1$, $\Delta_{M\_TEMP} = 1$
Goal Function	Max ($\mathcal{N}$ c : c ∈ $\{N_{12}, N_{21}, N_{22}, E_1, E_2\}$:¬AB(c) = true)	

Table 4. CSDP for the components cluster $\{N_{12}, N_{21}, N_{22}, E_1, E_2\}$ for the Heat Exchanger

To study two particular cases of diagnosis, there are two observational models, one for the Simple Problem and other one for the Heat Exchanger example. These two OM are used in tables 3 and 4 in order to obtain the domains of every observable variable.

For the two observational models, we verify the satisfaction of the constraints described in tables 3 and 4. The minimal diagnosis appears in table 5. We detect firstly simple faults and then multiple faults using the mentioned CSDP and the function to maximize.

Simple Problem		Heat Exchangers	
O.M	Minimal Diagnosis	O.M	Minimal Diagnosis
a=3, b=2, c=2, d=3, e=3, f=10, g=12	$\{M_1\}$, $\{A_1\}$, $\{M_2, M_3\}$, $\{M_2, A_2\}$	f_{16}=95, f_{21}=100, f_{26}=100, f_{12}=50, f_{13}=50, t_{16}=45, t_{21}=60, t_{26}=45, t_{12}=30, t_{13}=30	$\{N_{12}\}$, $\{E_1\}$, $\{E_2\}$, $\{N_{21}, N_{22}\}$,

Table 5. Minimal Diagnosis for section 2 examples

For the example 1 the diagnosis process offers us components M_1 and A_1 as possible component faults. If we change components M_1 or A_1 we will be able to modify the final result of f_1 and therefore, to satisfy the correct value, which has to be 12 instead of 10. For the example 2 the diagnosis process offers us components N_{12}, E_1 and E_2 as possible component faults. If we change these components we will be able to modify the final result of f_{16}, and therefore, to satisfy the correct value, which has to be 100 instead of 95.

6. CONCLUSIONS AND FUTURE WORKS

This paper shows that constraint programming can be a good solution in order to obtain de minimal diagnosis in a system which use interval variables. The determination of components clusters of the system reduces drastically the computational complexity to detect conflicts. The use of components clusters allow us to reduce the set of minimal possible conflict.

As future works we want to improve our methodology using a constraint database in order to store polynomial constraints. A constraint database allow us to use the power of SQL in order to query the database. We have used only one observational model to carry out the diagnosis, but we think that the use of a greater number of observational models will improve our methodology to obtain minimal diagnosis.

7. ACKNOWLEDGEMENTS

This work has been partially funded by the Spanish Ministerio de Ciencia y Tecnologia under grant DPI2003-07146-C02-01, and the European Regional Development Fund (ERDF/ FEDER).

8. REFERENCES

[Cordier00] M. Cordier, F. Lévy, J. Montmain, L. Travémassuy´ es, M. Dumas, M. Staroswiecki, and P. Dague, 'A comparative analysis of AI and control theory approaches to model-based diagnosis', 14th European Conference on Artificial Intelligence, 136–140, 2000.

[Fuentes03] Fuentes Martinez, A. and Kuchcinski K. Multifrequency Test and Diagnosis of Analog Circuits Using Constraint Programming and Interval Arithmetic,Procs. of DDECS 2003

[Gasca03] R.M. Gasca, C. Del Valle, R. Ceballos, and M. Toro, 'An integration of FDI and DX approaches to polinomial models', DX, 2003.

[Genesereth84] M. Genesereth, 'The use of design descriptions in automated diagnosis', Artificial Intelligence 24, 411–436, 1984.

[Guernez97] C. Guernez, "Fault detection and isolation on non linear polynomial systems" 15th IMACS World Congress on Scientific, Computation, Modelling and Applied Mathematics, 1997.

[Ilog02] ILOG: ILOG Solver 5.1 User's Manual. ILOG 2002.

[Kask00] K. Kask., 'New Search Heuristics for Max-CSP', in Proceeding of CP'2000, pg. 262–277, 2000.
[de Kleer87] J. De Kleer and B.C. Williams, 'Diagnosing multiple faults', Artificial Intelligence, 1987.
[de Kleer92] J. De Kleer, A. Mackworth, and R. Reiter, 'Characterizing diagnoses and systems', Artificial Intelligence 56, 2-3, 197–222, 1992.
[Krysander02] M. Krysander and M. Nyberg, 'Structural analysis utilizing mss sets with application to a paper plant', Proc. of the Thirteenth International Workshop on Principles of Diagnosis, Austria, 2002.
[Larrosa99] J. Larrossa and P. Meseguer., 'Partition-based lower bound for Max-CSP', in Proceedings CP, pages 303–315, 1999.
[Mozetic93] I.Mozetic, F.Novak, M.Santo-Zarnik, A.Biasizzo, 'Diagnosing analog circuits Designed-for-testability by using CLP(R) ', in Proc. 4th Int'l Workshop on Principles of Diagnosis, pp.105-120, Aberystwyth, UK, 1993.
[Reiter84] R. Davis, 'Diagnostic reasoning based on structure and behavior', Artificial Intelligence 24, 347–410, 1984.
[Reiter87] R. Reiter, 'A theory of diagnosis from first principles', Artificial Intelligence 32, 1, 57–96, 1987.