Diagnosis con variables intervalares usando técnicas Max-CSP

R. Ceballos, V. Cejudo, C. Del Valle, R. M. Gasca
Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla
Avda. Reina Mercedes s/n, 41012 Sevilla (Spain)
email: {ceballos,cejudo,carmelo,gasca}@lsi.us.es

Abstract

En ingeniería son muchas las aplicaciones que usan modelos basados en variables y parámetros cuyos dominios aparecen limitados a intervalos. Estos modelos almacenan el comportamiento esperado del sistema físico. En este trabajo se propone una aproximación para automatizar la determinación de la diagnosis de este tipo de sistemas. Se compone de dos fases. En la primera se realiza un preprocesamiento del sistema para encontrar subsistemas diagnosticables independientemente. La división en subproblemas, con menos componentes que el problema original, reduce la complejidad computacional y evita cálculos innecesarios. Este paso es especialmente necesario en los sistemas con un gran número de componentes. En la segunda fase, se construye un modelo basado en la programación con restricciones para obtener la diagnosis del sistema. En esta fase, se establecen intervalos en los cuales el sistema puede funcionar, restringiendo los dominios posibles a ciertas variables. Los resultados obtenidos demuestran la validez de las técnicas de programación con restricciones para encontrar la diagnosis mínima en sistemas de componentes.

1. Introducción

El proceso de diagnosis intenta determinar por qué un sistema correctamente diseñado no trabaja conforme a la forma especificada. Su objetivo es detectar e identificar qué partes del sistema no funcionan correctamente. Se basa en la monitorización usando sensores, a través de los cuales disponemos de una representación fiel del sistema y de las desviaciones que en él se producen. En el mundo de la empresa, poder diagnosticar a tiempo es importante ya que los fallos producidos en los componentes y procesos, traen consigo paradas indeseables y deterioros en el sistema, con el consiguiente aumento de costos, disminución de la producción y efectos ambientales.

La diagnosis es un campo muy activo de investigación. Existen dos comunidades que trabajan en paralelo y de forma normalmente aislada en este campo: la metodología FDI (Fault Detection and Isolation) proveniente del campo del control automático, y la metodología DX emergida del campo de la Inteligencia Artificial. En el área DX la formalización se concretó en [Reiter, 1987] y [de Kleer *et al.*, 1992], donde se proponen las teorías generales para poder explicar las discrepancias entre los comportamientos observados y correctos de los componentes de un sistema. Las primeras implementaciones fueron DART [Genesereth, 1984] y GDE [de Kleer and Williams, 1987]; ambas detectaban fallos pero usaban diferentes mecanismos de inferencia. Los modelos centrados en componentes describen los sistemas mediante relaciones entrada/salida. La mayoría de las aproximaciones aparecidas para diagnosis de componentes caracterizan la diagnosis de un sistema como una colección de conjuntos mínimos de componentes que fallan, y que explican los comportamientos observados (síntomas). Por tanto, es muy importante disponer de un buen modelo para determinar la diagnosis de un sistema.

En la comunidad FDI, los trabajos [Staroswiecki and Declerk, 1989], [Cassar and Staroswiecki, 1997] presentan la formalización del análisis estructural, es decir, el proceso por el cual se obtienen las ARRs (Analytical Redundancy Relation) del sistema. Últimamente hay una apuesta por la integración entre técnicas provenientes de las metodologías DX y FDI. La integración de las teorías de FDI con DX (BRIDGE Task Group [Cordier *et al.*, 2000]), y las pruebas de sus equivalencias se han mostrado para varios supuestos en [Cordier *et al.*, 2000], [Gasca *et al.*, 2003]. En [Ceballos *et al.*, 2004] se proponen técnicas para realizar parte del trabajo de forma off-line, basándose en la generación de todos los posibles contextos minimales que posee el sistema y aplicando las bases de Gröbner para lograr las ecuaciones que determinan el comportamiento del problema. La diagnosis mínima se obtiene de forma on-line usando un modelo observacional. En [Pulido and González, 2004] se presentan también técnicas para conseguir realizar parte del trabajo de diagnosis de forma off-line. Se basa en almacenar, a través de hipergrafos, dependencias precompiladas entre componentes. Al aplicar un modelo observacional sobre las estructuras precompiladas es posible detectar los conflictos y en base a ellos generar la diagnosis mínima.

La propuesta que hacemos en este trabajo es una metodología para automatizar el proceso de diagnosis de un sistema cuando las variables deben tener dominios intervalares. Se basa en dos fases:

- Un preprocesamiento estructural con el objetivo de reducir la complejidad computacional del proceso de diagnosis, especialmente en los sistemas con una gran número de componentes.
- Y la generación de un modelo para diagnosticar basado en la satisfacción de restricciones, concretamente un Max-CSP (problema de maximización de satisfacción de restricciones). El objetivo de este modelo será encontrar aquellas restricciones que impiden el comportamiento especificado para un sistema en su conjunto.

La programación con restricciones (Constraint Programming) es un paradigma que permite la resolución de un CSP (Constraint Satisfaction Problem). La metodología propuesta en este trabajo se basa en la generación de un modelo CSP para encontrar la diagnosis mínima de un sistema. La programación con restricciones nos permite modelar y resolver problemas reales como un conjunto de restricciones entre variables. Un problema de satisfacción de restricciones se define basándose en un conjunto de variables $X=\{X_1,X_2...,X_n\}$ asociadas a unos dominios, $D=\{D_1,D_2,...,D_n\}$, y un conjunto de restricciones $C=\{C_1,C_2,...,C_m\}$. Cada restricción C_i es una tupla (W_i,R_i), donde R_i es una relación $R_i \subseteq D_{i1}x...xD_{ik}$ definida para el subconjunto de variables $W_i \subseteq X$. Si tenemos un CSP, el objetivo del Max-CSP (Maximization Constraint Satisfaction Problem) es encontrar una asignación que satisfaga el mayor número de restricciones, y que minimice el número de restricciones violadas. La búsqueda de soluciones con técnicas Max-CSP es muy compleja, y algunas investigaciones han intentado mejorar su eficiencia [Kask., 2000][Larrossa and Meseguer, 1999].

La programación con restricciones fue propuesta en [I.Mozetic *et al.*, 1993] para diagnosticar circuitos analógicos. La herramienta usada fue CLP($\Re$). Una de los problemas de esta aproximación, debido a las restricciones de CLP($\Re$), es que se limita sólo a restricciones lineales. Otra aproximación, [Martinez and Kuchcinski, 2003], se propuso para circuitos analógicos basados en aritmética intervalar. Esta aproximación combina información proveniente de test realizados a diferentes frecuencias.

La estructura que seguiremos en el artículo será la siguiente: en primer lugar se presentan dos ejemplos muy utilizados en la bibliografía relativa, y que permitirán aplicar la metodología propuesta. Después se expondrán las definiciones y notación usada en el artículo. A continuación aparecen los pasos a seguir, en primer lugar, realizar el preprocesamiento estructural del sistema, y en segundo lugar, generar el modelo basado en la satisfacción de restricciones. Por último, la metodología se aplica a los dos ejemplos propuestos, y se exponen las conclusiones y trabajos futuros.

2. Ejemplos

Para clarificar la metodología propuesta, se han escogido los siguientes ejemplos, cuyo uso es muy común en las investigaciones relativas a la diagnosis basada en modelos:

2.1. Sistema polybox

Este ejemplo es el más utilizado en la bibliografía relativa a la diagnosis basada en modelos [Reiter, 1984][de Kleer *et al.*, 1992]. Esta formado por tres multiplicadores y dos sumadores, tal como se presenta en la figura 1. Los multiplicadores están representados como M_1, M_2 y M_3, y los sumadores como A_1 y A_2. Para detectar qué componentes fallan, sólo se conocen los valores de las variables observables a, b, c, d, e, f y g, que aparecen marcadas con un circulo sombreado sobre la figura.

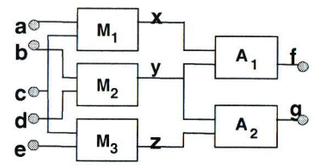

Figura 1: Sistema polybox

2.2. Sistema de intercambiadores de calor

Este sistema fue inicialmente propuesto en [Guernez, 1997]. En la figura 2 se muestra el sistema especificado, las zonas marcadas con un circulo sombreado indican que se ha colocado un sensor de flujo y temperatura en dicho punto. Consiste en seis intercambiadores de calor y tres flujos f_i de entrada, que entran en el sistema con diferentes temperaturas t_i. Es posible diferenciar tres subsistemas diferentes, cada uno formado por dos intercambiadores: {E1, E2}, {E3, E4} y {E5, E6}. Cada uno de los seis intercambiadores, y cada uno de los ocho nodos del sistema, son considerados como componentes en el proceso de diagnosis. El comportamiento del sistema puede ser descrito mediante las siguientes restricciones polinómicas provenientes de los tres diferentes posibles tipos de balances:

$\sum_i f_i = 0$: balance de masa en
$\sum_i f_i \cdot t_i = 0$: balance térmico en
$\sum_{in} f_i \cdot t_i - \sum_{out} f_j \cdot t_j = 0$: balance de e
intercambiador de calc

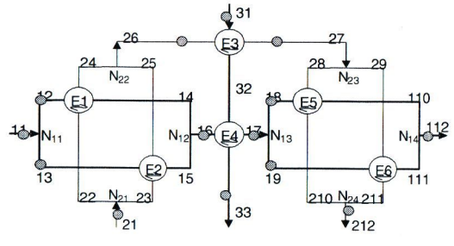

Figura 2: Sistema de los intercambiadores de calor

Tabla 1: Descripción del sistema de intercambiadores de calor

C.	Restricciones	C.	Restricciones	C.	Restricciones
N_{11}	$\mathbf{f_{11}}$-$\mathbf{f_{12}}$-$\mathbf{f_{13}}$ = 0	N_{23}	$\mathbf{f_{27}}$-f_{28}-f_{29} = 0	E_4	$\mathbf{f_{16}}$-$\mathbf{f_{17}}$ = 0
	$\mathbf{f_{11}}\cdot\mathbf{t_{11}}$-$\mathbf{f_{12}}\cdot\mathbf{t_{12}}$-$\mathbf{f_{13}}\cdot\mathbf{t_{13}}$ = 0		$\mathbf{f_{27}}\cdot\mathbf{t_{27}}$-$f_{28}\cdot t_{28}$-$f_{29}\cdot t_{29}$ = 0		f_{32}-$\mathbf{f_{33}}$ = 0
N_{12}	f_{14}+f_{15}-$\mathbf{f_{16}}$ = 0	N_{24}	f_{210}+f_{211}-$\mathbf{f_{212}}$ = 0		$\mathbf{f_{16}}\cdot\mathbf{t_{16}}$-$\mathbf{f_{17}}\cdot\mathbf{t_{17}}$+$f_{32}\cdot t_{32}$-$\mathbf{f_{33}}\cdot\mathbf{t_{33}}$ = 0
	$f_{14}\cdot t_{14}$+$f_{15}\cdot t_{15}$-$\mathbf{f_{16}}\cdot\mathbf{t_{16}}$ = 0		$f_{210}\cdot t_{210}$+$f_{211}\cdot t_{211}$-$\mathbf{f_{212}}\cdot\mathbf{t_{212}}$ = 0	E_5	$\mathbf{f_{18}}$-f_{110} = 0
N_{13}	$\mathbf{f_{17}}$-$\mathbf{f_{18}}$-$\mathbf{f_{19}}$ = 0	E_1	$\mathbf{f_{12}}$-f_{14} = 0		f_{28}-f_{210} = 0
	$\mathbf{f_{17}}\cdot\mathbf{t_{17}}$-$\mathbf{f_{18}}\cdot\mathbf{t_{18}}$-$\mathbf{f_{19}}\cdot\mathbf{t_{19}}$ = 0		f_{22}-f_{24} = 0		$\mathbf{f_{18}}\cdot\mathbf{t_{18}}$-$f_{110}\cdot t_{110}$+$f_{28}\cdot t_{28}$
N_{14}	f_{110}+f_{111}-$\mathbf{f_{112}}$ = 0		$\mathbf{f_{12}}\cdot\mathbf{t_{12}}$-$f_{14}\cdot t_{14}$+$f_{22}\cdot t_{22}$-$f_{24}\cdot t_{24}$ = 0		-$f_{210}\cdot t_{210}$ = 0
	$f_{110}\cdot t_{110}$+$f_{111}\cdot t_{111}$-$\mathbf{f_{112}}\cdot\mathbf{t_{112}}$ = 0	E_2	$\mathbf{f_{13}}$-f_{15} = 0	E_6	$\mathbf{f_{19}}$-f_{111} = 0
N_{21}	$\mathbf{f_{21}}$-f_{22}-f_{23} = 0		f_{23}-f_{25} = 0		f_{29}-f_{211} = 0
	$\mathbf{f_{21}}\cdot\mathbf{t_{21}}$-$f_{22}\cdot t_{22}$-$f_{23}\cdot t_{23}$ = 0		$\mathbf{f_{13}}\cdot\mathbf{t_{13}}$-$f_{15}\cdot t_{15}$+$f_{23}\cdot t_{23}$-$f_{25}\cdot t_{25}$ = 0		$\mathbf{f_{19}}\cdot\mathbf{t_{19}}$-$f_{111}\cdot t_{111}$+$f_{29}\cdot t_{29}$
N_{22}	f_{24}+f_{25}-$\mathbf{f_{26}}$ = 0	E_3	$\mathbf{f_{26}}$-$\mathbf{f_{27}}$ = 0		-$f_{211}\cdot t_{211}$ = 0
	$f_{24}\cdot t_{24}$+$f_{25}\cdot t_{25}$-$\mathbf{f_{26}}\cdot\mathbf{t_{26}}$ = 0		$\mathbf{f_{31}}$-f_{32} = 0		
			$\mathbf{f_{26}}\cdot\mathbf{t_{26}}$-$\mathbf{f_{27}}\cdot\mathbf{t_{27}}$+$\mathbf{f_{31}}\cdot\mathbf{t_{31}}$-$f_{32}\cdot t_{32}$ = 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}}
V_{nob}={f_{14},f_{15},f_{110},f_{111},f_{22},f_{23},f_{24},f_{25},f_{28},f_{29},f_{210},f_{211},f_{32},t_{14},t_{15},t_{110},t_{111},t_{22},t_{23},t_{24},t_{25},t_{28},t_{29},t_{210},t_{211},t_{32}}

3. Notación y definiciones

Para clarificar la presentación de la metodología propuesta se hace necesario introducir algunas definiciones extraídas de la terminología usada en la comunidad DX. La diagnosis basada en modelos requiere un modelo del sistema que represente el comportamiento del sistema en conjunto, y de cada uno de sus componentes por separado. En nuestro caso, usaremos restricciones para representar las ecuaciones que definen el comportamiento del sistema.

Definición 1. Precisión de monitorización ($\pm\Delta_M$): Se define como la exactitud que se tendrá al monitorizar las señales de entrada y de salida de los componentes. Si sabemos el valor exacto de la señal monitorizada, entonces $\Delta_M \cong 0$, aunque lo normal en los sistemas reales es que éste valor sea diferente de cero.

Definición 2. Precisión del componente ($\pm\Delta_C$): Se define como la exactitud con la que un componente se comportará tal como está especificado. Si el componente trabaja exactamente tal como se establece en su modelo, entonces podemos suponer que $\Delta_C \cong 0$, pero al igual que en la precisión de monitorización, lo normal en los sistemas reales es que este valor sea diferente de cero. La precisión de los componentes establece un concepto muy parecido al de la tolerancia cuando se establecen las ARRs en la metodología FDI.

Definición 3. Descripción del sistema (SD, System Description): La descripción del sistema vendrá dada por la tupla (P, V_{ob}, V_{nob}). P se define como un conjunto finito de ecuaciones que determinan el comportamiento de los componentes (COMPS) y del sistema en su conjunto. Las ecuaciones del sistema están formadas por variables observables (V_{ob}), obtenidas directamente de los sensores, y no observables (V_{nob}).

Tabla 2: Descripción del sistema polybox

Componentes	Restricciones
M_1	$\lvert x - a \cdot c\rvert \leq \Delta_{C_{Mult}}$
M_2	$\lvert y - b \cdot d\rvert \leq \Delta_{C_{Mult}}$
M_3	$\lvert z - c \cdot e\rvert \leq \Delta_{C_{Mult}}$
A_1	$\lvert f - x - y\rvert \leq \Delta_{C_{Add}}$
A_2	$\lvert g - y - z\rvert \leq \Delta_{C_{Add}}$
V_{ob} = {a, b, c, d, e, f, g}	V_{nob} = {x, y, z}

En las tablas 1 y 2 aparecen las descripciones del sistema para los ejemplos propuestos en la sección 2. Algunos componentes tienen asociada una precisión, que se aplica directamente sobre las restricciones (ecuaciones) del componente. En el ejemplo propuesto en la sección 2.1 se han usado dos tipos de precisiones para los componentes del sistema, dependiendo de si se trata de un multiplicador o un sumador. La precisión establecida para los multiplicadores se denota $\Delta_{C_{Mult}}$, y para los sumadores $\Delta_{C_{Add}}$.

Definición 4. Modelo observacional (OBS): Se define como una tupla que asigna valores a las variables observables.

Definición 5. Problema de diagnosis basado en la satisfacción de restricciones (CSDP, Constraint Satisfaction Diagnosis Problem): Se define como un problema de satisfacción de restricciones (CSP) planteado en función de las restricciones obtenidas de la descripcion del sistema (SD) a diagnosticar. Al aplicar un modelo observacional, el resultado de este problema será el conjunto de posibles componentes que son posibles fallos del sistema.

El proceso de búsqueda de la diagnosis se sustenta en el concepto de comportamiento anormal. Básicamente, si un componente tiene un comportamiento no anormal, entonces su funcionamiento es correcto. En el CSDP se hará uso del predicado AB para identificar si un comportamiento es anormal o no. Dado un componente *c* perteneciente al sistema, si el predicado ¬AB(c) es verdadero, entonces el componente funciona correctamente.

Definición 6. Diagnosis [Reiter, 1987]: La diagnosis para un CSDP será un conjunto de componentes $\mathcal{D} \subseteq$ COMPS tal que SD $\bigcup$ OBS $\bigcup$ {AB(c) | c $\in \mathcal{D}$} $\bigcup$ {¬AB(c) | c $\in$ COMPS $- \mathcal{D}$} puede ser satisfecho.

El número de posibles diagnosis es exponencial, concretamente 2^{nComps}, donde *nComps* representa el número de componentes del sistema. Nuestro objetivo es refinar la diagnosis

y quedarnos con un número menor de posibilidades con la misma información.

Definición 7. Diagnosis mínima [Reiter, 1987]: La diagnosis mínima será aquella diagnosis $\mathcal{D}$ tal que $\forall\ \mathcal{D}' \subset \mathcal{D}, \mathcal{D}'$ no es una diagnosis.

4. Preprocesamiento estructural

El objetivo de esta sección es la división del sistema en subconjuntos que puedan ser diagnosticados de forma independiente. La diagnosis completa del sistema se obtendrá como la unión de las diagnosis obtenidas para todos los subsistemas.

La principal ventaja está en la reducción automática de la complejidad en la obtención de la diagnosis. Los subsistemas obtenidos son más simples que el sistema completo, y por tanto la complejidad computacional para detectar conflictos en cada uno de estos subsistemas es menor comparado con el sistema completo. Los subsistemas obtenidos serán lo que llamaremos *clusters* de componentes.

Definition 8 *Cluster de componentes*: Un conjunto de componentes T es un cluster de componentes si no existe ninguna variable no observable perteneciente a algún componente del cluster T que sea común con otro componente de otro cluster diferente, y además para todo T' $\subset$ T, se debe cumplir que T' no es un cluster de componentes.

En un cluster, todas las variables no observables (conexiones entre componentes) aparecen sólo en las restricciones asociadas a componentes que pertenecen al mismo cluster. Por tanto, todas las conexiones con componentes externos al cluster son a través de variables observables (que están monitorizadas). Un cluster de componentes está completamente monitorizado, y por esta razón, es posible la detección de los fallos dentro del cluster sin información de otros componentes externos al cluster. Cuanto menor sea el cluster de componentes, menor es el número máximo de componentes que pueden aparecer en la diagnosis, ya que como mucho pueden fallar todos los componentes que estén en el cluster. De ahí que en la definición de cluster se imponga que dentro de un cluster no pueda existir otro con menos elementos.

Los clusters de componentes tienen un número menor de componentes que el sistema completo (o igual si sólo existe un cluster de componentes). Gracias a esta propiedad, la complejidad computacional del proceso de diagnosis es siempre menor o igual usando el sistema equivalente dividido en clusters, ya que el número de posibles diagnosis se reduce. Sea X un conjunto de n componentes del sistema, y sean X_1 y X_2 clusters de $n-m$ y m componentes respectivamente tales que $X_1 \cup X_2 = X$ y $m > 0$. Entonces la complejidad computacional para detectar la diagnosis en X_1 y X_2 de forma separada es menor que la de todo el sistema completo X. El conjunto inicial de posibles diagnosis es 2^{nComp}, donde *nComp* es el número de componentes del sistema, tal como aparece en [de Kleer *et al.*, 1992]. El número de posibles diagnosis de dos clusters X_1 y X_2 respectivamente será: $(2^{n-m}) - 1 + (2^m) - 1 \leq 2^{n-m} \cdot 2^m - 2 < 2^n - 1$, que es el número de posibles diagnosis del sistema para el conjunto de componentes X.

Ejemplo: En el sistema de los intercambiadores de calor,

```
identificacionCluster(C) return T
  E = {}
  T = {}
  // Detectar todas las conexiones entre componentes
  foreach x ∈ C
    foreach y ∈ C
      if x ≠ y ∧ nonObsVar(x) ∩ nonObsVar(y) ≠ {}
        E = E ∪ {{x,y}}
      endif
    endforeach
  endforeach
  // Generar los clusters con un componente
  foreach x ∈ C
    T = T ∪ {{x}}
  endforeach
  // Agrupar los clusters de componentes
  foreach {x,y} ∈ E
    if ∃ S1,S2 | S1 ∈ T ∧ S2 ∈ T ∧ S1 ≠ S2
          ∧ x ∈ S1 ∧ y ∈ S2
      T = T \ S1
      T = T \ S2
      T = T ∪ {S1 ∪ S2}
    endif
  endforeach
```

Figura 3: Algoritmo para seleccionar los clusters de componentes

el intercambiador E_3 no está completamente monitorizado porque no es posible observar directamente el valor de f_{32} y t_{32}. Igualmente, el intercambiador E_4 tampoco está completamente monitorizado porque no es posible saber el valor de las variables f_{32} y t_{32}. Sin embargo, es posible monitorizar ambos componentes de forma conjunta, ya que todas las conexiones de estos dos componentes con el exterior son observables.

Algoritmo: El siguiente pseudocódigo (figura 3) define la función *identificacionCluster(C)*, que recibe *C*, el conjunto de todos los componentes del sistema, y devuelve *T*, el conjunto de clusters obtenidos del sistema. El algoritmo primero almacena en el conjunto *E* todas las parejas de componentes que tienen una variable no observable en común. A continuación crea tantos clusters como componentes existen en el sistema (n). Todos estos clusters tendrán inicialmente sólo un componente. Después, por cada elemento de *E*, es decir, por cada conexión entre dos componentes $x \in S_1$ e $y \in S_2$, donde S_1 y $S_2 \in A$, el algoritmo fusionará los conjuntos S_1 y S_2. Cuando el proceso haya acabado todos los componentes tendrán asignado un cluster. La función auxiliar *nonObsVar*(x) devuelve el conjunto de variables no-observables del componente x.

Para el ejemplo de los intercambiadores de calor visto en la sección 2.2, se obtienen cinco clusters de componentes que aparecen en la tabla 3.

Tabla 3: Clusters de componentes

CC	Restricciones	CC	Restricciones
1	$\{N_{11}\}$	4	$\{N_{14},N_{23},N_{24},E_5,E_6\}$
2	$\{N_{13}\}$	5	$\{E_3,E_4\}$
3	$\{N_{12},N_{21},N_{22},E_1,E_2\}$		

Tabla 4: Diagnosis del sistema polybox usando la metodología DX, CSP y Max-CSP

DX		CSP		Max−CSP	
SD:	¬AB(A1) ⇒ f = x + y ¬AB(A2) ⇒ g = y + z ¬AB(M1) ⇒ x = a * c ¬AB(M2) ⇒ y = b * d ¬AB(M3) ⇒ z = c * e	Restricciones:	(AB(A1) ∨ f = x + y)∧ (AB(A2) ∨ g = y + z)∧ (AB(M1) ∨ x = a * c)∧ (AB(M2) ∨ y = b * d)∧ (AB(M3) ∨ z = c * e)	Restricciones:	$\neg AB(A1) = (f = x + y)$ $\neg AB(A2) = (g = y + z)$ $\neg AB(M1) = (x = a * c)$ $\neg AB(M2) = (y = b * d)$ $\neg AB(M3) = (z = c * e)$
OBS:	a = 3, b = 2, c = 2, d=3, e=3, f=10, g=12	Dominios:	a = 3, b = 2, c = 2, d = 3, e = 3, f = 10, g = 12	Dominios:	a = 3, b = 2, c = 2, d = 3, e = 3, f = 10, g = 12
COMPS:	A1, A2, M1, M2, M3		AB(M1), AB(M2), AB(M3), AB(A1), AB(A2) = {verdadero, falso} x, y, z = libre		AB(M1), AB(M2), AB(M3), AB(A1), AB(A2) = {verdadero, falso} x, y, z = libre
				F. objetivo:	Max(N c : c ∈ {M1, M2, M3, A1, A2} : ¬AB(c) = verdadero)

5. Diagnosis utilizando técnicas CSP

La programación con restricciones nos permite modelar y resolver problemas reales a través de un conjunto de variables asociadas a unos dominios, y un conjunto de ecuaciones definidas para un subconjunto de variables. Al plantear un problema CSP el objetivo es encontrar para las variables una asignación que satisfaga todas las restricciones. Pero existen determinados problemas en los que es inviable satisfacer todas las restricciones. El objetivo al plantear un problema Max-CSP es encontrar una asignación que satisfaga el mayor número de restricciones, y que minimice el número de restricciones violadas.

Básicamente se trata de definir una función objetivo que el motor de inferencias debe intentar maximizar mientras busca la solución para el CSP planteado. Para este tipo de problemas se utilizan restricciones del tipo *reified constraint* (restricciones de satisfacción concretada), cuya definición es la siguiente:

Definición 9. *Reified constraint*: Será una restricción que tiene asociada una variable que tomará el valor verdadero o falso según se dé la satisfacción o no de la propia restricción. Esta variable booleana la llamaremos *reified variable*(variable de satisfactibilidad), y tomará el valor verdadero si la restricción asociada es satisfecha.

Para plantear la función objetivo de un problema Max-CSP, las restricciones que puedan ser violadas deben estar definidas como *reified constraints*. Al alcanzar la solución para el problema Max-CSP, todas las variables de satisfacción concretada tomarán el valor adecuado. La cardinalidad del conjunto de variables de satisfactibilidad que toman el valor verdadero será la función objetivo que buscamos maximizar, o lo que es lo mismo, satisfacer el mayor número de restricciones.

La diagnosis de sistemas busca encontrar qué componentes tienen un comportamiento anormal, es decir, cuáles de los componentes tienen un comportamiento diferente al especificado en las ecuaciones que forman la descripción del sistema. Si pensamos en las ecuaciones asociadas a los componentes como un conjunto de restricciones, podremos plantear la diagnosis de sistemas mediante un problema CSP. En la tabla 4 aparece, para el ejemplo del sistema polybox, cómo a partir del modelo del sistema se pueden obtener directamente cuáles serían las restricciones para un problema CSP. Cada una de las ecuaciones lógicas del modelo del sistema se traduce a una restricción de tipo OR que debe cumplirse al alcanzar la solución para el problema CSP. En el caso de la primera restricción del ejemplo, debe cumplirse al menos una de las dos siguientes condiciones: o bien que el componente tenga un comportamiento anormal, o bien que la salida f del componente sea igual a la suma de las entradas x e y. Con los valores del modelo observacional es posible acotar los dominios de las variables implicadas en las restricciones. Sin embargo, otras variables quedarán libres, como las variables no observables x, y, y z, así como el predicado $AB(C_i)$ aplicado a cada componente.

Encontrar la diagnosis mínima implica encontrar un conjunto de cardinalidad mínima de componentes con funcionamiento anormal. Pero esta propiedad de la diagnosis mínima no la estamos utilizando si planteamos el problema como un simple problema CSP. Para optimizar y guiar la búsqueda de la diagnosis mínima, se debe plantear como un problema Max-CSP. Un componente tendrá un comportamiento anormal si es diferente al especificado en las ecuaciones que hacen referencia a dicho componente, es decir, si las restricciones asociadas a dicho componente no pueden satisfacerse. Por tanto, el valor del predicado AB vendrá dado por la satisfacción o no de las restricciones asociadas a dicho componente. Tal como aparece en la tabla 4, para el ejemplo del sistema polybox, plantearemos un problema Max-CSP donde los predicados $AB(C_i)$ actuarán como *reified variables*. La función objetivo debe maximizar el número de predicados $\neg AB(C_i)$ que tomen el valor verdadero. De esta forma, aunque los predicados $AB(C_i)$ parten inicialmente como variables libres, tendrán su dominio intrínsecamente ligado a la satisfacción de las restricciones asociadas a los componentes.

Con la resolución del problema Max-CSP planteado se ten-

Tabla 5: CSDP para el sistema polybox

	Variable	Restriccines
Componentes	M_1	$\neg AB(M_1) = [\|x - a \cdot c\| \leq \Delta_{C_{Mult}}]$
	M_2	$\neg AB(M_2) = [\|y - b \cdot d\| \leq \Delta_{C_{Mult}}]$
	M_3	$\neg AB(M_3) = [\|z - c \cdot e\| \leq \Delta_{C_{Mult}}]$
	A_1	$\neg AB(A_1) = [\|f - x - y\| \leq \Delta_{C_{Add}}]$
	A_2	$\neg AB(A_2) = [\|g - y - z\| \leq \Delta_{C_{Add}}]$
Dominios	AB	$AB(M_1), AB(M_2), AB(M_3), AB(A_1), AB(A_2) = \{$verdadero, falso$\}$
	V_{nobs}	x, y, z = {libre}
	V_{obs}	a = 3, b = 2, c = 2, d = 3, e = 3, f = 10, g = 12
	Δ	$\Delta_{C_{Add}} = 0.1$, $\Delta_{C_{Mult}} = 0.1$, $\Delta_{M_{Sign}} = 0.1$
Función objetivo	goal	Max (N c : c $\in \{M_1, M_2, M_3, A_1, A_2\}$: $\neg$AB(c) = verdadero)

Tabla 6: CSDP para el cluster 3 del sistema de intercambiadores de calor

	Variable	Restricciones
Componentes	N_{12}	$\neg AB(N_{12}) = [(f_{14}+f_{15}-f_{16} = 0) \wedge (f_{14}\cdot t_{14}+f_{15}\cdot t_{15}-f_{16}\cdot t_{16} = 0)]$
	N_{21}	$\neg AB(N_{21}) = [(f_{21}-f_{22}-f_{23} = 0) \wedge (f_{21}\cdot t_{21}-f_{22}\cdot t_{22}-f_{23}\cdot t_{23} = 0)]$
	N_{22}	$\neg AB(N_{22}) = [(f_{24}+f_{25}-f_{26} = 0) \wedge (f_{24}\cdot t_{24}+f_{25}\cdot t_{25}-f_{26}\cdot t_{26} = 0)]$
	E_1	$\neg AB(E_1) = [(f_{12}-f_{14} = 0) \wedge (f_{22}-f_{24} = 0) \wedge (f_{12}\cdot t_{12}-f_{14}\cdot t_{14}+f_{22}\cdot t_{22}-f_{24}\cdot t_{24} = 0)]$
	E_2	$\neg AB(E_2) = [(f_{13}-f_{15} = 0) \wedge (f_{23}-f_{25} = 0) \wedge (f_{13}\cdot t_{13}-f_{15}\cdot t_{15}+f_{23}\cdot t_{23}-f_{25}\cdot t_{25} = 0)]$
Dominios	AB	$AB(N_{12}), AB(N_{21}), AB(N_{22}), AB(E_1), AB(E_2) = \{$verdadero, falso$\}$
	V_{nobs}	$f_{14}, f_{15}, f_{22}, f_{23}, f_{24}, f_{25}, t_{14}, t_{15}, t_{22}, t_{23}, t_{24}, t_{25} = \{$libre$\}$
	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	$\Delta_{M_{Flow}} = 1$, $\Delta_{M_{Temp}} = 1$
Función objetivo	goal	Max (N c : c $\in \{N_{12}, N_{21}, N_{22}, E_1, E_2\}$: $\neg$AB(c) = verdadero)

drán los valores asociados a los predicados $AB(c_i)$ para cada uno de los componentes. Estos valores definirán el grupo de componentes que forman la diagnosis. Para el problema planteado en la tabla 4, se buscarán primero aquellas soluciones que permitan alcanzar un funcionamiento correcto del sistema modificando un sólo componente. En este caso resultan ser $\{A_1\}$ y $\{M_1\}$. Pero, ¿es ésta la única diagnosis posible?. Evidentemente no, el proceso Max-CSP en principio ha conseguido maximizar la función objetivo utilizando sólo un componente, pero existe la posibilidad de que sean dos o más componentes los que fallen simultáneamente.

Para seguir buscando el resto de posibles diagnosis mínimas, el problema Max-CSP debe continuar buscando más soluciones. Pero previamente resulta necesario añadir como restricciones que los predicados $AB(c_i)$ correspondientes a los componentes que ya han sido obtenidos en el proceso de diagnosis, deben ser falsos obligatoriamente, y de esta forma garantizar que no se generarán nuevas soluciones cuyo conjunto de componentes englobe los componentes que formen alguna de las diagnosis mínimas ya obtenidas.

Para el ejemplo de la tabla 4 se obtendrían como diagnosis mínima utilizando dos componentes los grupos $\{A_2, M_2\}$ y $\{M_2, M_3\}$. Si aún quedaran componentes por cubrir, seguiríamos buscando soluciones formadas por grupos de tres, cuatro, o más componentes hasta utilizar todos, añadiendo las restricciones necesarias para que no se repitan las diagnosis mínimas obtenidas previamente. En la tabla 4 se ha usado un modelo observacional que no incorpora dominios intervalares para las variables del sistema. En el siguiente apartado se usarán dominios intervalares para algunas variables de los ejemplos propuestos.

6. Diagnosis de sistemas con variables de dominios intervalares

Para estudiar casos en los cuales aparezcan variables con dominios intervalares, se han utilizado dos modelos observacionales, uno para cada uno de los ejemplos presentados.

Para cada uno de los clusters obtenidos se debe construir un CSDP diferente e independiente, para garantizar un coste computacional menor. Por tanto, para cada uno de los cinco clusters obtenidos en el ejemplo de los intercambiadores de calor se debe construir un CSDP diferente. En la tabla 5 aparece el CSDP asociado al ejemplo del sistema polybox. En la tabla 6 aparece el CSDP correspondiente al cluster número tres ($\{N_{12}, N_{21}, N_{22}, E_1, E_2\}$) del ejemplo de los intercambiadores de calor.

Con el modelo observacional se definen los dominios posibles para las variables observables, sin embargo las variables no observables quedarán libres para tomar cualquiera de los valores posibles en sus dominios iniciales. Cada variable observable tiene asignada una precisión de monitorización que representa la fiabilidad con la que es posible monitorizar una variable. En el primer ejemplo se ha definido sólo un tipo de precisión de monitorización para las variables observables, $\Delta_{M_{Sign}}$. En el ejemplo de los intercambiadores de calor existen dos tipos diferentes de precisiones de monitorización, para las temperaturas $\Delta_{M_{Temp}}$ y para los flujos $\Delta_{M_{Flow}}$, en cada nodo o intercambiador de calor. En el primer ejemplo para cada tipo de componente se ha definido además una precisión de componente: la precisión establecida para los multiplicadores se ha denotado como $\Delta_{C_{Mult}}$, y para los sumadores $\Delta_{C_{Add}}$.

Tabla 7: Diagnosis mínima para los ejemplos de la sección 2 (usando el CSDP de las tablas 5 y 6)

Sistema	Diagnosis Mínima
Polybox	$\{M_1\}, \{A_1\}, \{M_2, M_3\}$ $\{M_2, A_2\}$
Intercambiadores de calor	$\{N_{12}\}, \{E_1\}, \{E_2\}$ $\{N_{21}, N_{22}\}$

Tal como aparece en las tablas 5 y 6, para cada componente c, el predicado AB(c) será usado como una variable de satisfacción concretada. El predicado AB(c) tomará el valor verdadero si el componente tiene un comportamiento anormal, es decir, si las restricciones asociadas a dicho componente no pueden cumplirse junto a las otras del sistema. Estamos interesados en las soluciones que implican cambiar el menor número de componentes, lo que implica maximizar el valor de los predicados AB(c) que tomen el valor verdadero.

Para implementar la búsqueda de las soluciones se ha utilizado la herramienta ILOG-Solver^{TM}([ILOG, 2002]). Los dominios de las variables de han definido como flotantes para poder almacenar intervalos continuos. La herramienta Ilog-Solver permite afrontar un ámplio espectro de problemas de una manera fácil y versatil. El proceso de diagnosis que hemos presentado permite encontrar la solución siempre. El tiempo requerido para encontrar las soluciones depende del número de componentes del cluster y del modelo observacional.

Para los dos modelos observacionales, se han detectado primero los fallos simples y luego los fallos múltiples, obteniendo los resultados que aparecen en la tabla 7. Para el ejemplo 1 el proceso de diagnosis ofrece los componentes M_1 y A_1 como posibles diagnosis. Cualquiera de estos dos componentes podría ser el causante del valor incorrecto en la salida f, que debería ser 12 en lugar de 10. Tambien aparecen como fallos dobles $\{M_2, M_3\}$ y $\{M_2, A_2\}$ para el ejemplo del sistema polybox. En el ejemplo de los intercambiadores de calor el proceso de diagnosis establece como diagnosis mínima los fallos simples en los componentes N_{12}, E_1, E_2 , o bien un fallo doble en $\{N_{21}, N_{22}\}$. Si se cambian estos componentes es posible modificar el valor final de f_{16}, y por tanto, satisfacer que el valor correcto es 100 en lugar de 95.

Como se puede observar, el proceso de diagnosis se basa en detectar cuáles son los síntomas, es decir, las diferencias entre las predicciones hechas por el motor de inferencias y las observaciones obtenidas por la monitorización. Los síntomas indican que componentes pueden estar fallando, y cuáles son los conflictos, es decir, los conjuntos de componentes que no funcionan bien conjuntamente. En definitiva, se busca realizar una hipótesis de porqué el sistema difiere del modelo especificado. Determinados problemas exigen que ciertas restricciones deban cumplirse preferentemente, en este caso se podrán utilizar pesos para las diferentes variables de satisfactibilidad dentro de la función objetivo.

7. Conclusiones y trabajos futuros

En este trabajo se ha mostrado cómo la programación con restricciones puede ser una buena solución para obtener la diagnosis mínima en un sistema que utilice variables intervalares. El preprocesamiento del sistema permite obtener los clusters de componentes pertenecientes al sistema, y dividir el proceso de diagnosis en varios procesos independientes. La diagnosis del sistema se obtiene como unión de las diagnosis generadas para cada uno de los clusters de componentes. La división en clusters permite una reducción de la complejidad computacional para la detección de la diagnosis mínima. La programación con restricciones y las herramientas que la sustentan, permiten un desarrollo rápido y versatil de un entorno para la detección de la diagnosis de un sistema. Frente a otros entornos poco adaptables y difíciles de aprender, la programación con restricciones proporciona sencillez y desarrollos rápidos.

Nuestras líneas de investigación se centran ahora en ampliar la metodología a modelos con sistemas dinámicos. Además estamos trabajando con sistemas más complejos, en los cuales la aplicación de la metodología no es tan directa.

8. Reconocimiento

Este trabajo ha sido financiado por el Ministerio de Ciencia y Tecnología de España (DPI2003-07146-C02-01) y el European Regional Development Fund. (ERDF/FEDER).

Referencias

[Cassar and Staroswiecki, 1997] J. Cassar and M. Staroswiecki. A structural approach for the design of failure detection and identification systems. In *IFAC-IFIP-IMACS Conf. on Control of Industrial Processes, Belfort, France*, 1997.

[Ceballos *et al.*, 2004] R. Ceballos, M.T. Gomez, R.M. Gasca, and S. Pozo. Determination of possible minimal conflict sets using components clusters and grobner bases. In *DX04*, pages 21–26, Carcassonne, France, June 2004.

[Cordier *et al.*, 2000] M. Cordier, F. Lévy, J. Montmain, L. Travémassuyés, M. Dumas, M. Staroswiecki, and P. Dague. A comparative analysis of AI and control theory approaches to model-based diagnosis. In *14th European Conference on Artificial Intelligence*, pages 136–140, 2000.

[de Kleer and Williams, 1987] J. de Kleer and B.C. Williams. Diagnosing multiple faults. *Artificial Intelligence*, 1987.

[de Kleer *et al.*, 1992] J. de Kleer, A. Mackworth, and R. Reiter. Characterizing diagnoses and systems. *Artificial Intelligence*, 2-3(56):197–222, 1992.

[Gasca *et al.*, 2003] R.M. Gasca, C. Del Valle, R. Ceballos, and M. Toro. An integration of FDI and DX approaches to polinomial models. In *DX03, Proceedings of the 14th International Workshop on Principles of Diagnosis*, 2003.

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

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

[ILOG, 2002] ILOG. *ILOG Solver 5.1 User's Manual*. ILOG Press, 2002.

[I.Mozetic *et al.*, 1993] I.Mozetic, F.Novak, M.Santo-Zarnik, and A.Biasizzo. Diagnosing analog circuits designed-for-testability by using CLP(R). In *DX93, Proceedings of the 4th International Workshop on Principles of Diagnosis*, pages 105–120, Aberystwyth, UK, 1993.

[Kask., 2000] K. Kask. New search heuristics for max-CSP. In *Sixth International Conference on Principles and Practice of Constraint Programming, Singapore*, pages 262–277, September 2000.

[Larrossa and Meseguer, 1999] J. Larrossa and P. Meseguer. Partition-based lower bound for MAX-CSP. In *Fifth International Conference on Principles and Practice of Constraint Programming, Virginia*, pages 303–315, October 1999.

[Martinez and Kuchcinski, 2003] A. Fuentes Martinez and K. Kuchcinski. Multifrequency test and diagnosis of analog circuits using constraint programming and interval arithmetic. In *DDECS00, IEEE Workshop on Design and Diagnostics of Electronic Circuits and Systems*, 2003.

[Pulido and González, 2004] B. Pulido and C. Alonso González. Possible conflicts: A compilation technique for consistency-based diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics. Part B: Cybernetics*, 34(5):2192–2206, October 2004.

[Reiter, 1984] R. Reiter. Diagnostic reasoning based on structure and behavior. *Artificial Intelligence 24*, pages 347–410, 1984.

[Reiter, 1987] R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence 32*, 1:57–96, 1987.

[Staroswiecki and Declerk, 1989] M. Staroswiecki and P. Declerk. Analytical redundancy in non linear interconnected systems by means of structural analysis. In *IFAC Advanced Information Processing in Automatic Control (AIPAC-89)*, pages 51–55, Nacy, France, June 1989.