

# Aplicación de CSPs no Binarios al Análisis ROC\*

M.A. Salido<sup>1</sup>, F. Barber<sup>2</sup>, C. Ferri<sup>2</sup>, J. Hernández-Orallo<sup>2</sup>

<sup>1</sup> Dpto. Ciencias de la Computación e Inteligencia Artificial, Universidad de Alicante  
Campus de San Vicente, Ap. de Correos: 99, E-03080, Alicante, Spain  
`msalido@dsic.upv.es`

<sup>2</sup> Dpto. Sistemas Informáticos y Computación, Universidad Politécnica de Valencia  
Camino de Vera s/n, 46071, Valencia, Spain  
{fbarber, cferri, jorallo}@dsic.upv.es

**Resumen** Hoy en día muchos problemas de la vida real se pueden modelar como problemas de satisfacción de restricciones (CSPs). Por ejemplo en áreas tales como inteligencia artificial, investigación operativa, bases de datos, sistemas expertos y diagnosis la importancia de los CSPs se está incrementando paulatinamente. Sin embargo la mayoría de los investigadores centran su atención en los CSPs binarios y discretos. En este artículo presentamos el algoritmo HSA aplicado al análisis ROC, que genera un CSP no binario y continuo y cuyo objetivo es obtener todas las soluciones (envoltura convexa) para calcular así el volumen que ocupa dicho conjunto de soluciones.

## 1. Introducción

Muchos problemas reales pueden ser modelados como problemas de satisfacción de restricciones (CSPs) no binarias. Sin embargo la mayoría de los investigadores que trabajan en CSPs centran su atención en restricciones binarias. Es bien conocido que todo problema de satisfacción de restricciones no binarias puede ser transformado a un problema binario equivalente [14]. Sin embargo, esta transformación de restricciones tiene ciertos inconvenientes. En primer lugar, la binarización puede tener un alto coste computacional haciendo que con frecuencia esta transformación sea impracticable [10],[4]. Por otra parte, el proceso de transformación genera nuevas variables, las cuales pueden tener dominios muy grandes, causando unos requerimientos de memoria extra para los algoritmos que lo resuelven. Por ello, en algunos casos, resolver el problema binarizado puede resultar muy ineficiente [2]. Además, una binarización forzada puede generar una formulación poco natural causando unas dificultades extras para las interfaces de los resolutores de restricciones y la comprensión del usuario humano [5].

Por lo tanto, resulta a menudo más adecuado manejar las restricciones no binarias de forma directa, de manera que los problemas mantengan sus formulaciones originales.

---

\* Este trabajo ha sido realizado durante la estancia realizada por Miguel A. Salido en la Universidad Politécnica de Valencia (PPI-02-03) como profesor invitado y parcialmente financiado por el proyecto CTIDIB/2002/112 de la Generalitat Valenciana y los proyectos CICYT DPI2001-2094-C03-03 y TIC2001-4936-E del gobierno español.

En este artículo proponemos una aplicación del algoritmo HSA (Hyperpolyhedron Search Algorithm)” [16],[15] que es una técnica heurística para manejar problemas de satisfacción de restricciones no binarias de forma directa y natural. Este algoritmo maneja el problema mediante un polítopo que representa en sus facetas las restricciones del problema, actualizándose cada vez que una nueva restricción se inserta en el problema.

En muchos problemas reales hay que tomar decisiones para llevar a cabo una acción u otra. Para ello se utilizan *clasificadores*, que hacen predicciones sobre dichas decisiones utilizando para ello experiencias previas. Sin embargo, debido a que las predicciones pueden ser erróneas, es importante conocer cual es su efecto, cuando éstas son incorrectas. En muchas situaciones los errores tienen distintas consecuencias. Algunos errores tienen unos costes más elevados que otros, especialmente en el campo de la diagnosis. Por ejemplo, un tratamiento o diagnosis erróneo puede tener diferentes costes y peligros dependiendo del tipo de error que se ha producido. Obviamente, los costes de cada clasificación errónea son dependientes del problema, pero casi nunca se da el caso de que ellos sean uniformes para un solo problema. Consecuentemente, la precisión o el error no es generalmente la mejor manera para evaluar la calidad de un clasificador o de un algoritmo de aprendizaje.

El aprendizaje sensible al coste es una generalización más realista del aprendizaje predictivo, y los modelos sensibles al coste permiten una mejor toma de decisiones. La calidad de un modelo se mide en términos de minimización de coste más que en términos de minimización de errores. Cuando se proporcionan a priori las matrices de coste, es decir, antes de que el aprendizaje tenga lugar, las matrices tienen que explotarse completamente para obtener modelos que minimicen el coste. Sin embargo, en muchas circunstancias, los costes no son conocidos a priori o los modelos ya están seleccionados. El análisis ROC (Receiver Operating Characteristic) [12] [18] [13] ha demostrado ser muy útil para la evaluación de los clasificadores cuando la matriz de coste no era conocida en la construcción de los clasificadores. El análisis ROC proporciona herramientas para seleccionar un conjunto de clasificadores que se comportarían óptimamente y para rechazar algunos otros clasificadores menos útiles. Para hacer esto, se construye la envoltura convexa de todos los clasificadores, dando una curva que junto a los ejes genera un polítopo convexo. esto se puede realizar de manera sencilla para clasificadores binarios (2 clases) pero hasta el momento no se ha presentado una solución para más de dos clases.

Como veremos, el análisis ROC puede ser resuelto por el algoritmo completo HSA mientras que es difícil resolverlo mediante los resolutores tradicionales debido a que estos trabajan principalmente con problemas discretos y con restricciones binarias. Además, su objetivo se centra en estudiar la consistencia del problema y en obtener una o todas las soluciones del problema. Sin embargo en un problema continuo esto es imposible. HSA maneja este tipo de problemas y obtiene todas las soluciones extremas que se necesitan para obtener el hiperpoliedro resultante y obtener así el volumen que ocupa la envoltura convexa del espacio ROC de clasificadores de  $n$  clases pudiendo ser  $n > 2$ .

## 2. Preliminares

Brevemente, el problema de satisfacción de restricciones (CSP) que HSA puede manejar consiste en:

- un conjunto de variables  $X = \{x_1, x_2, \dots, x_n\}$ ;
- un conjunto de dominios  $D = \{D_1, D_2, \dots, D_n\}$  asociados a cada variable en  $X$ . Cada dominio  $D_i$  está definido como un intervalo de la forma  $D_i = [l_i, u_i]$ ;
- un conjunto de restricciones limitando los valores que las variables pueden simultáneamente tomar.

Una solución a un CSP es una asignación a cada variable de un valor de su dominio, de tal manera que se satisfacen todas las restricciones del problema. El objetivo de HSA puede ser obtener: una solución, sin preferencia alguna; varias o todas las soluciones; los dominios mínimos de las variables;

Las restricciones que vamos a manejar para el estudio del análisis ROC son inigualdades lineales y desigualdades lineales:

$$\text{Inigualdad lineal : } \sum_{i=1}^n p_i x_i \leq b \quad (1)$$

$$\text{Desigualdad lineal : } \sum_{i=1}^n p_i x_i \neq b \quad (2)$$

De esta manera, mediante combinación de restricciones de inigualdad (1) y de desigualdad (2) podremos expresar cualquier otro tipo de expresión de la forma  $\alpha r c$  donde  $\alpha$  es un polinomio de grado 1,  $c$  una constante y  $r \in \{<, \leq, =, \neq, \geq, >\}$ .

**Definición 1** Un Polígono se define como una región finita del espacio  $n$ -dimensional acotada por un número finito de hiperplanos (de dimensión  $n-1$ ). Coxeter, en [6], define el polígono como el término general del "punto, línea, segmento, polígono, poliedro,..."

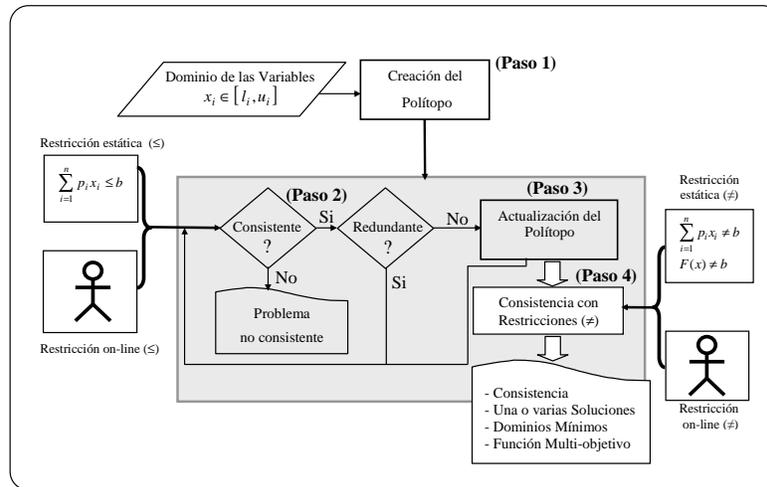
## 3. Especificación del Algoritmo HSA

El algoritmo HSA se considera como un resolutor de problemas de satisfacción de restricciones no binarias sobre dominios continuos, donde podremos obtener el polígono que englobe al conjunto de soluciones posibles del problema. En la Figura 1 se presenta la especificación de HSA de forma gráfica.

HSA genera un polígono inicial (paso 1), con  $2^n$  vértices generados mediante el producto cartesiano de los valores extremos de los dominios de las variables  $[l_1, u_1] \times [l_2, u_2] \times \dots \times [l_n, u_n]$ . Cada faceta del polígono está definida mediante  $2^n - 1$  vértices, por lo que el polígono completo tendrá  $2^n$  vértices.

En primer lugar HSA estudia las restricciones de igualdad ( $\leq$ ) (pasos 2 y 3) y posteriormente trata las restricciones de desigualdad ( $\neq$ ) (paso 4), el cual no vamos a utilizar en el análisis ROC. Para cada restricción ( $\leq$ ) de entrada, HSA lleva a cabo la prueba de la consistencia (paso 2). Si la restricción no es consistente, HSA detecta dicha inconsistencia y finaliza la ejecución. Sin embargo si la restricción es consistente, el algoritmo estudia si la restricción es redundante. Si es así, el polígono no se actualiza y continua analizando la siguiente restricción. Si no es redundante, se actualiza el polígono (paso 3). Finalmente cuando se han estudiado todas las restricciones de igualdad y el problema es consistente, HSA estudia la consistencia de las restricciones de desigualdad ( $\neq$ ) (paso 4).

Las soluciones al CSP son todos los vértices del polígono resultante, así como cualquier combinación convexa entre dos vértices del polígono que satisfagan todas las restricciones de desigualdad. Para el caso del análisis ROC, HSA mantendrá el polígono resultante para obtener el volumen correspondiente.



**Figura 1.** Especificación Gráfica de HSA.

**Teorema** El algoritmo HSA es un algoritmo completo y correcto.

*Demostración* HSA es un algoritmo completo y correcto ya que mantiene en todo momento el conjunto de soluciones en un polígono convexo cuyas facetas son las restricciones de igualdad que están presentes en el problema. Cuando este polígono es no vacío cada una de las soluciones obtenidas por HSA es correcta, y además encuentra todas las soluciones existente mediante combinación convexa.

## 4. Utilización de HSA para la obtención de polítopos ROC

En esta sección, presentamos la aplicabilidad que tiene nuestro algoritmo completo HSA para llevar a cabo la extensión real del área bajo la curva ROC (Area Under the Curve, AUC) al volumen bajo la superficie ROC que llamaremos VUS (Volume Under ROC Surface), mostrando cómo generar el hiper-poliedro que englobe a todos los clasificadores. Además compararemos el VUS real obtenido mediante HSA con aproximaciones o extensiones de la AUC para más de dos clases.

El análisis ROC y la medida AUC se han usado exhaustivamente en el ámbito de la toma de decisiones en medicina [9] [11], así como en el área de la extracción de conocimiento, en las herramientas de minería de datos, el reconocimiento de patrones [1] o la ciencia en general [18].

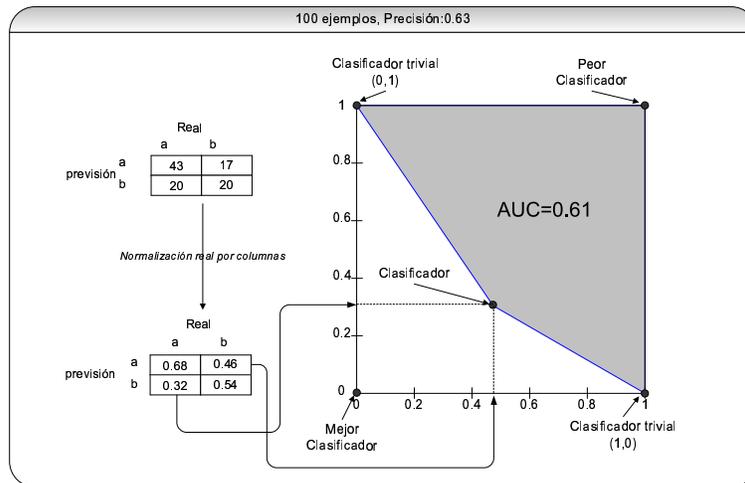
En el caso trivial, un clasificador de dos clases forma una curva ROC compuesta por cuatro segmentos que se generan con los siguientes puntos (ver Figura 2): el punto dado por el clasificador  $(0.46, 0.32)$  que proviene de los errores de clasificar como  $a$  cuando realmente es  $b$  ( $a \rightarrow b$ ) y de clasificar como  $b$  cuando realmente es  $a$  ( $b \rightarrow a$ ), los dos puntos que representan los clasificadores triviales (es decir, el clasificador que siempre predice a la clase 0 y el clasificador que siempre predice a la clase 1) y el punto origen. En la Figura 2 presentamos de forma general la representación de un clasificador con dos clases, donde posteriormente explicaremos el significado de los polígonos generados. Estos polígonos generan un área que puede ser calculada y que lógicamente variará entre 0.5 (área mínima) y 1 (área máxima). El área del polígono superior, que hemos llamado AUC, se ha convertido en una mejor alternativa que la precisión o el error para evaluar clasificadores, ya que obtiene la bondad de un clasificador para diferentes distribuciones de las clases.

Sin embargo, la aplicabilidad del análisis ROC y del AUC solamente se ha mostrado viable para problemas con dos clases. Aunque teóricamente el análisis ROC puede extenderse para manejar problemas multi-dimensionales [17], sin embargo en la práctica no se ha podido utilizar debido a la dificultad de la formulación de los clasificadores triviales. El principal inconveniente es la alta dimensionalidad.

No obstante, a pesar de la dificultad, en este trabajo vamos a ver que es posible llevar a cabo el análisis ROC para más de dos clases y el cálculo de la AUC o, más precisamente, el volumen bajo la superficie ROC (VUS). Sin embargo, en la literatura, tanto los clasificadores triviales para más de dos clases, como el volumen mínimo y máximo no se han identificado hasta la fecha.

En este artículo, presentamos cómo calcular los clasificadores triviales, y el VUS mínimo y máximo para clasificadores de más de dos clases. Nosotros compararemos experimentalmente el VUS real obtenido utilizando el algoritmo HSA, con las extensiones de AUC obtenidas por Hand&Till [8].

Veamos a continuación las equivalencias que se generan cuando pasamos del estudio de clasificadores de dos clases a tres clases:



**Figura 2.** Representación de un clasificador con dos clases

- El polígono generado bajo la curva ROC de los clasificadores de dos clases, se transforma en un hiper-poliedro de 6 dimensiones para los clasificadores de tres clases, por lo que su visión gráfica se hace imposible.
- El área bajo la curva ROC (AUC) se calcula con los cuatro puntos necesarios para generar el polígono en clasificadores de dos clases. En cambio, para tres clases, se transforma en el volumen bajo la superficie (VUS) de un hiper-poliedro de 6 dimensiones del cual se necesitan todos los puntos extremos para dicho cálculo. Pasemos, en primer lugar, a calcular el volumen máximo y mínimo.

#### 4.1. Volumen máximo bajo la superficie (VUS) para tres clases

El volumen máximo debería representar todos los posibles clasificadores.

Para obtener todos los posibles clasificadores, deberemos resolver el siguiente CSP continuo:

- Variables:  $x_1, x_2, x_3, x_4, x_5, x_6$ ;
- Dominios continuos:  $x_i \in [0, 1], \forall i : 1.,6$ ;
- Restricciones:
  - $x_3 + x_5 \leq 1$ ;
  - $x_1 + x_6 \leq 1$ ;
  - $x_2 + x_4 \leq 1$ ;

Un punto dentro del hiper-poliedro resultante representa a un clasificador con tres clases.

Dados 6 valores bajo una distribución uniforme entre 0 y 1, representado por  $U(0,1)$ , tenemos que la probabilidad de que un punto formado por estos 6 valores satisfaga las tres restricciones anteriores es:

$$VUS^{max} = P(U(0,1) + U(0,1) \leq 1) \cdot P(U(0,1) + U(0,1) \leq 1) \cdot P(U(0,1) + U(0,1) \leq 1) = P(U(0,1) + U(0,1) \leq 1)^3.$$

Es fácil ver que la probabilidad de que la suma de dos números aleatorios bajo la distribución uniforme  $U(0,1)$  sea menor que 1 es exactamente  $1/2$ , es decir,

$$P(U(0,1) + U(0,1) \leq 1) = 1/2$$

Por lo tanto, el volumen máximo teórico es:

$$VUS^{max} = (1/2)^3 = 1/8$$

Sin embargo, ¿cuáles son los puntos cuya envoltura convexa componen este volumen?

Para ello resolvemos el CSP continuo anterior mediante el algoritmo HSA obteniendo 41 puntos extremos, donde el volumen que genera la envoltura convexa de estos 41 puntos, utilizando Qhull [3], es  $1/8$ , tal y como hemos visto teóricamente (Vease también [7]).

#### 4.2. Volumen mínimo bajo la superficie (VUS) para tres clases

Veamos la manera de obtener el VUS mínimo. Sin pérdida de generalidad podemos construir clasificadores triviales como se muestra en la Figura 3 (izquierda), donde  $h_a + h_b + h_c = 1$ .

		Real					Real		
		a	b	c			a	b	c
previsión	a	$h_a$	$h_a$	$h_a$	previsión	a	$v_{aa}$	$v_{ba}$	$v_{ca}$
	b	$h_b$	$h_b$	$h_b$		b	$v_{ab}$	$v_{bb}$	$v_{cb}$
	c	$h_c$	$h_c$	$h_c$		c	$v_{ac}$	$v_{bc}$	$v_{cc}$

**Figura 3.** Clasificador trivial y clasificador cualquiera

Esto, obviamente incluye los tres clasificadores triviales extremos: 'todo es  $a$ ', 'todo es  $b$ ' y 'todo es  $c$ '.

Dado un clasificador cualquiera como el mostrado en la Figura 3 (derecha), nosotros podemos descartar este clasificador si y sólo si:

$$\exists h_a, h_b, h_c \in \mathbb{R}^+ : h_a + h_b + h_c = 1 \text{ y además } v_{ba} \geq h_a; v_{ca} \geq h_a; v_{ab} \geq h_b; v_{cb} \geq h_b; v_{ac} \geq h_c; v_{bc} \geq h_c$$

De aquí podemos derivar el siguiente teorema [7]:

**Teorema.** Un clasificador  $(x_1, x_2, x_3, x_4, x_5, x_6)$  se puede descartar si y sólo si  $r_1 + r_2 + r_3 \geq 1$ , donde  $r_1 = \min(x_1, x_2)$ ,  $r_2 = \min(x_3, x_4)$  y  $r_3 = \min(x_5, x_6)$

Con las propiedades previas, nosotros solamente tenemos que calcular el espacio de aquellos clasificadores que cumplen la condición de que  $r_1 + r_2 + r_3 \geq 1$ , donde  $r_1 = \min(x_1, x_2)$ ,  $r_2 = \min(x_3, x_4)$  y  $r_3 = \min(x_5, x_6)$ , para así obtener el volumen mínimo correspondiente a la ausencia total de información.

De esta manera el CSP no binario y continuo que deberemos resolver mediante el algoritmo HSA es el siguiente:

- Variables:  $x_1, x_2, x_3, x_4, x_5, x_6, r_1, r_2, r_3$ ;
- Dominios continuos:  $x_i \in [0, 1], i : 1..6$  y  $r_j \in [0, 1], j = 1, 2, 3$ ;
- Restricciones:
  - $x_3 + x_5 \leq 1$
  - $x_1 + x_6 \leq 1$
  - $x_2 + x_4 \leq 1$
  - $r_1 + r_2 + r_3 \geq 1$ , donde  $r_1 = \min(x_1, x_2)$ ,  $r_2 = \min(x_3, x_4)$  y  $r_3 = \min(x_5, x_6)$

Este CSP no binario se resuelve mediante HSA y obteniendo 25 puntos, donde el volumen que genera la envoltura convexa de estos 25 puntos es  $1/180$ , coincidiendo con el que experimentalmente obtenemos mediante el método de Monte Carlo.

Resumiendo, tenemos el siguiente VUS para tres clases:

	Teórico	Monte Carlo ( $10^6$ Casos)	Exacto (HSA)
$VUS^{max}$	1/8	0.12483	0.125
$VUS^{min}$	1/180	0.555523	0.55555556

### 4.3. Clasificadores Triviales

A pesar de los buenos resultados obtenidos, parece intuitivo (o al menos así ocurre en clasificadores con dos clases) que si cogemos el mínimo y le añadimos el punto origen (el mejor clasificador) deberíamos obtener el máximo. Sin embargo esto no ocurre, ya que con el mínimo obtenido con un volumen de  $1/180$  y añadiendo el  $(0,0,0,0,0,0)$  obtenemos 10 puntos con un volumen de  $1/120$  que es muy inferior al  $1/8$  obtenido como máximo. Esto parece contradictorio, ya que se supone que tener el mejor clasificador daría el máximo. Lógicamente si se tiene un clasificador  $(0,0,0,0,0,0)$ , cualquier clasificador que tenga un valor mayor de 0 para cualquier coordenada es descartable y esto debería dar  $1/8$ .

La cuestión es que cuando se añade un clasificador deberíamos ver las condiciones que forma. El clasificador perfecto genera las ecuaciones de descarte  $x_i \geq 0, \forall i = 1..6$ , que son ecuaciones nulas, ya que las variables están acotadas en dominios  $[0, 1]$ .

De esta manera dado un clasificador  $C_1$ , nos podemos plantear la siguiente pregunta ¿qué podemos descartar?

Podremos descartar cualquier clasificador  $C_2$  tal que supere a  $C_1$  combinado con los triviales, es decir, que  $C_2$  tenga cada valor mayor en cada una de las casillas.

En realidad lo que tenemos que mirar es la combinación lineal de los tres triviales con el que tenemos:

$$h_a \cdot (1, 1, 0, 0, 0, 0) + h_b \cdot (0, 0, 1, 1, 0, 0) + h_c \cdot (0, 0, 0, 0, 1, 1) + h_d \cdot (z_{ba}, z_{ca}, z_{ab}, z_{cb}, z_{ac}, z_{bc})$$

que lo podemos descartar si:

$$\begin{aligned} \exists h_a, h_b, h_c, h_d \in \mathbb{R}^+ : h_a + h_b + h_c + h_d = 1 \text{ y ademas} \\ v_{ba} \geq h_a + 0 + 0 + h_d \cdot z_{ba} \\ v_{ca} \geq h_a + 0 + 0 + h_d \cdot z_{ca} \\ v_{ab} \geq 0 + h_b + 0 + h_d \cdot z_{ab} \\ v_{cb} \geq 0 + h_b + 0 + h_d \cdot z_{cb} \\ v_{ac} \geq 0 + 0 + h_c + h_d \cdot z_{ac} \\ v_{bc} \geq 0 + 0 + h_c + h_d \cdot z_{bc} \end{aligned}$$

Esto da lugar a un CSP no binario y continuo con 10 incognitas que deberemos resolver mediante el algoritmo HSA.

Del resultado obtenido, nos quedamos con las 6 variables de todos los puntos obtenidos, pudiendo calcular el volumen que genera la envoltura convexa de todos ellos.

Veamos a continuacion una tabla comparativa, en la que evaluamos, para distintos ejemplos, nuestra herramienta llamada HSA+QHULL (HSA), con otras aproximaciones al VUS como Macro-average (MAC), 1-P Trivial (TRIV), y el metodo experimental generado mediante el metodo de Monte Carlo (MONT).

	<b>MONT</b>	<b>HSA</b>	<b>MAC</b>	<b>TRIV</b>
(0,0,0,0,0,0)	0.125	0.125	1	1
(1,1,1,0,0,0)	0.0055	0.0055	0	0.333
(0.5,0.5,0,0,0,0)	0.0327	0.0342	0.666	0.666
(0.5,0,0.5,0,0,0)	0.0327	0.03105	0.666	0.666
(0.75,0.75,0.75,0,0,0)	0.0076	0.00832	0.25	0.25
(0.33,0,0.33,0,0,0)	0.05218	0.05221	0.777	0.777

**Tabla 1.** Estudio comparativo de distintas aproximaciones

## 5. Conclusiones

En este articulo hemos presentado el algoritmo completo HSA aplicado a la resolucion de problemas de clasificadores con tres clases. Hasta ahora, no se ha trabajado con este tipo de problemas por la complejidad que conlleva la obtencion de los puntos extremos y por el hecho de que los clasificadores con

más de dos clases son intratables visualmente. Actualmente, estamos trabajando en el manejo de clasificadores con más de tres clases, para poder generalizarlo a cualquier dimensión. Una vez hecha la extensión de dos a tres clases, aplicarlo a cualquier número de clases consiste en ampliar las desigualdades y obtener los puntos mediante HSA.

## Referencias

1. N.M. Adams and D.J. Hand, 'Comparing classifiers when the misallocation costs are uncertain', *Pattern Recognition*, **32**, 1139–1147, (1999).
2. F. Bacchus and P. van Beek, 'On the conversion between non-binary and binary constraint satisfaction problems', *In proceeding of AAAI-98*, 311–318, (1998).
3. C.B. Barber and H.T. Huhdanpaa, 'The Quickhull algorithm for convex hulls', *ACM Trans. on Mathematical Software*, (1996).
4. C. Bessière, 'Non-binary constraints', *In Proc. Principles and Practice of Constraint Programming (CP-99)*, 24–27, (1999).
5. C. Bessière, P. Meseguer, E.C. Freuder, and J. Larrosa, 'On forward checking for non-binary constraint satisfaction', *In Proc. Principles and Practice of Constraint Programming (CP-99)*, 88–102, (1999).
6. H.S.M. Coxeter, *Regular Polytopes*, 3rd ed., New York: Dover, 1973.
7. C Ferri, J Hernandez-Orallo, and M.A. Salido, 'Volume under the ROC surface for multi-class problems', *LNAI: Proceeding of 14th European Conference on Machine Learning (ECML)*, (2003).
8. D.J. Hand and R.J. Till, 'A simple generalisation of the area under the ROC curve for multiple class classification problems', *Machine Learning*, **45**, 171–186, (2001).
9. J.A. Hanley and B.J. McNeil, 'The meaning and use of the area under a receiver operating characteristic (ROC) curve', *Radiology*, **143**, 29–36, (1982).
10. J. Larrosa, *Algorithms and Heuristics for total and partial Constraint Satisfaction*, Phd Dissertation, UPC, Barcelona, 1998.
11. D. Mossman and E. Somoza, 'ROC curves, test accuracy, and the description of diagnostic tests', *J. Neuropsychiatr. Clin. Neurosci.*, **3**, 330–3., (1991).
12. F. Provost and T. Fawcett, 'Analysis and visualization of classifier performance: Comparison under imprecise class and cost distribution', *Proc. of KDD-97*, 43–48, (1997).
13. F. Provost and T. Fawcett, 'Robust classification for imprecise environments', *Machine Learning*, **42**, 203–231, (2001).
14. F. Rossi, C. Petrie, and V. Dhar, 'On the equivalence of constraint satisfaction problems', *In proceeding of European Conference of Artificial Intelligence*, 550–556, (1990).
15. M.A. Salido and F. Barber, 'An incremental and non-binary CSP solver: The Hyperpolyhedron Search Algorithm', *In Proc. of 7th International Conference on Principles and Practice of Constraint Programming (CP-01)*, 799–780, (2001).
16. M.A. Salido, A. Giret, and F. Barber, 'Constraint Satisfaction by means of Dynamic Polyhedra', *Operational Research Proceedings 2001. Ed. Springer Verlag*, **1**, 405–412, (2001).
17. A. Srinivasan, *Note on the Location of Optimal Classifiers in N-dimensional ROC Space*, Technical Report PRG-TR-2-99, Oxford University Computing Laboratory, Oxford, 1999.
18. J. Swets, R. Dawes, and J. Monahan, 'Better decisions through science', *Scientific American*, 82–87, (2000).