



Revista de Matemática: Teoría y Aplicaciones

ISSN: 1409-2433

mta.cimpa@ucr.ac.cr

Universidad de Costa Rica

Costa Rica

Méndez-Gurrola, Iris Iddaly; Laureano–Cruces, Ana Lilia; Ramírez–Rodríguez, Javier

Reductos mínimos con GRASP

Revista de Matemática: Teoría y Aplicaciones, vol. 18, núm. 1, 2011, pp. 149-162

Universidad de Costa Rica

San José, Costa Rica

Disponible en: <http://www.redalyc.org/articulo.oa?id=45326927011>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica

Red de Revistas Científicas de América Latina, el Caribe, España y Portugal

Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

REDUCTOS MÍNIMOS CON GRASP

MINIMAL REDUCTS WITH GRASP

IRIS IDDALY MÉNDEZ–GURROLA*

ANA LILIA LAUREANO–CRUCES†

JAVIER RAMÍREZ–RODRÍGUEZ‡

*Received: 18 Feb 2010; Revised: 20 Aug 2010;
Accepted: 26 Nov 2010*

Resumen

La detección adecuada del nivel de demencia de un paciente es importante para brindar el tratamiento adecuado. El diagnóstico se basa en ciertos criterios, reflejados en los exámenes clínicos. De estos exámenes surgen las limitaciones y el grado en que se encuentra cada una de ellas en los pacientes. Para reducir el total de limitaciones a ser evaluadas, utilizamos la teoría de Conjuntos Imprecisos, esta teoría ha sido aplicada en áreas de la inteligencia artificial tales como análisis de decisiones, sistemas expertos, descubrimiento de conocimiento, clasificación con múltiples atributos. En nuestro

*Posgrado en Diseño, Línea: Nuevas Tecnologías, UAM-A y Departamento de Matemáticas Aplicadas y Sistemas, UAM-C Artificios 40, Col. Hidalgo, Del. Álvaro Obregón, México, D. F. E-Mail: iddalym@yahoo.com.mx

†Universidad Autónoma Metropolitana, Departamento de Sistemas, Avenida San Pablo 180, 02200 México D. F. E-Mail: c1c@correo.azc.uam.mx

‡Universidad Autónoma Metropolitana, Departamento de Sistemas. Misma dirección que/same address as Laureano-Cruces, E-Mail: jararo@correo.azc.uam.mx

caso se aplica para encontrar el conjunto mínimo de limitaciones o reducto que genera la misma clasificación que considerando todas las limitaciones, para ello se desarrollo un algoritmo GRASP (Greedy Randomized Adaptive Search Procedure).

Palabras clave: demencias, conjuntos imprecisos, heurísticas, reducción de datos.

Abstract

The proper detection of patient level of dementia is important to offer the suitable treatment. The diagnosis is based on certain criteria, reflected in the clinical examinations. From these examinations emerge the limitations and the degree in which each patient is in. In order to reduce the total of limitations to be evaluated, we used the rough set theory, this theory has been applied in areas of the artificial intelligence such as decision analysis, expert systems, knowledge discovery, classification with multiple attributes. In our case this theory is applied to find the minimal limitations set or reduct that generate the same classification that considering all the limitations, to fulfill this purpose we development an algorithm GRASP (Greedy Randomized Adaptive Search Procedure).

Keywords: demencias, rough set, heuristics, data reduction.

Mathematics Subject Classification: 68T20, 68T30, 68T37.

1 Introducción

La correcta detección de demencia permite aplicar el tratamiento adecuado al paciente y con ello intentar mejorar o mantener la calidad de vida tanto del mismo como de sus cuidadores. Para esta detección, los médicos, revisan tanto los datos familiares como la historia personal de cada uno de los pacientes así como un examen clínico, que comprende la aplicación de diversas pruebas. Con estos elementos es posible diagnosticar al paciente. La demencia constituye la causa principal de incapacidad a largo plazo en la tercera edad y conlleva muchos problemas además de la pérdida de memoria, por tanto realmente significa un deterioro global de la capacidad del individuo.

La teoría de conjuntos imprecisos surgió en la década de los ochenta fue propuesta por Z. Pawlak [5] ha tenido muchas aplicaciones en años recientes. El enfoque de esta teoría ha sido importante en la inteligencia artificial y las ciencias cognitivas, especialmente en áreas como sistemas inteligentes, reconocimiento de patrones, descubrimiento de conocimiento, análisis de decisiones y sistemas expertos. Esta teoría puede ser vista

como un nuevo enfoque matemático a la vaguedad. La teoría comprende el cálculo de particiones en la cual se hace una clasificación de los objetos y posteriormente la reducción y dependencias de atributos. Aplicar esta metodología a un grupo de pacientes, nos permite detectar cuales son los atributos principales que se deben tomar en cuenta para la clasificación de fases. Tendremos como variable dependiente el estado del paciente, es decir la fase en que se encuentra: inicial, intermedio o avanzado y como variables independientes tendremos la información de las limitaciones que sufren los pacientes.

El objetivo central de aplicar la teoría de conjuntos imprecisos a este caso es que de un conjunto grande de atributos relacionados con las limitaciones de los pacientes, encontremos el conjunto mínimo de atributos o reducto que genere la misma partición que considerando todos los atributos.

Se ha demostrado que el problema de encontrar el mínimo reducto de un sistema de decisión es equivalente al problema del primer implicante mínimo, el cual es un problema NP-Completo [2]. En el caso de estudio se ha utilizado un algoritmo GRASP y los reductos encontrados han reducido hasta en un 50% los atributos considerados.

2 Conceptos de la teoría de conjuntos imprecisos

La teoría de conjunto imprecisos (CI) es un enfoque que se utiliza en aplicaciones de inteligencia artificial. Esta teoría fue originalmente desarrollada en los años 80 por el investigador Z. Pawlak [5], la cual se utiliza como herramienta formal para trabajar con la incertidumbre o la vaguedad inherente a un proceso de decisión. En esta teoría se maneja el cálculo de particiones. Se basa en que a cada objeto del universo se le puede asociar alguna información (datos, conocimiento). Los objetos que tienen la misma información no son discernibles a la vista de la información disponible. La representación del conocimiento de los objetos en la teoría de CI implica un sistema de información o sistema de decisión [1].

2.1 Sistema de información

Los elementos de un sistema de información denotado $A = (U, A)$ son:

- U , un conjunto de objetos (acciones, alternativas, pacientes, candidatos, observaciones, \dots), $U \neq \emptyset$,
- A , un conjunto de atributos (características, variables, condiciones características), $A \neq \emptyset$,

- Si $a \in A$ entonces $a : U \rightarrow V_a$,
- V_a valor conjunto.

Por tanto un Sistema de Información (SI) es un conjunto de objetos descrito por atributos.

2.2 Sistema de decisión

Esta es una extensión/especialización de un sistema de información. Es utilizado el mismo símbolo, A , para representarlo: $A = (U, A \cup \{d\})$, $d \notin A$. Un sistema de decisión es un sistema de información con un atributo de decisión extra.

Los CI por tanto son una colección de objetos que no pueden ser clasificados de forma precisa en términos de valores del conjunto de atributos y entonces existen casos fronterizos. Por tanto los conjuntos imprecisos son representados por un par de conjuntos precisos, llamados la aproximación por encima y por debajo [7].

2.3 Matriz de discernibilidad

$M(A)$ es una matriz de discernibilidad $n \times n$ del sistema de información A con coeficiente c_{ij} , tal que:

$$c_{ij} = \{a \in A : a(x_i) \neq a(x_j)\} \quad i, j \in \{1, \dots, n\}$$

donde $n = |A|$ es el número de atributos.

En la matriz de discernibilidad se tiene un índice por cada combinación de dos objetos del sistema de información. Este índice contiene la lista de atributos por cada par de objetos que tienen diferentes valores. Cada atributo en esta lista puede discernir entre estos dos objetos.

2.4 Reducto

Un reducto o conjunto mínimo, es el menor conjunto de atributos que mantiene la misma calidad de clasificación o las mismas particiones que el conjunto de todos los atributos. Esto se logra analizando el sistema de información o sistema de decisión y descubriendo las dependencias entre atributos (eliminando aquellos atributos que sean redundantes).

3 Selección de datos y representación

Las demencias son enfermedades en las que existe una alteración de la memoria, de las funciones cognitivas y conductuales así como alteraciones de la funcionalidad con respecto al estado previo de una persona [3].

El diagnóstico de demencia debe incluir ciertos criterios para poder establecerse. Entre ellos se encuentran:

- Deterioro de memoria a corto y largo plazo.
- Alteraciones cognitivas: afasia, apraxia, agnosia.
- Alteraciones de la funcionalidad previa del paciente.
- Trastornos en la vida social y laboral.

Existen diversos tipos de demencias: la enfermedad de Alzheimer [4], la demencia vascular (o por multi-infartos cerebrales), demencia por cuerpos de Lewy, demencias fronto-temporales, entre otras, las más frecuentes son las dos primeras.

Con relación a los datos, la población utilizada es un grupo de pacientes de un centro de día de la Cd. de México. Este centro cuenta con 22 pacientes que padecen algún tipo de demencia. En cuanto a las variables, se consideraron las limitantes que sufren los pacientes, clasificándolas básicamente en tres grupos principales: 1) Alteraciones cognoscitivas, 2) Alteraciones de la funcionalidad y 3) Alteraciones no cognoscitivas o conductuales.

Los datos utilizados para determinar los valores que incluiría en el sistema de información fueron obtenidos del análisis clínico de cada paciente. Algunas de las limitaciones seleccionadas para incluirse como atributos de cada paciente se muestran en la Tabla 1.

	Limitaciones
1	Carencia de la palabra
2	Alteración de la comprensión auditiva y escrita
3	Alteración de la expresión verbal y la comunicación
4	Alteración del léxico
5	Interacciones fonéticas anómalas
6	Desplazamiento de consonantes
...	...
65	Depresión
66	Perseverancia
67	Incontinencia urinaria o fecal
68	Conductas repetitivas
69	Vagabundeo
70	Falta de control emocional

Tabla 1: Listado de limitaciones que sufren los pacientes con demencias.

Se formó un sistema de información con los valores de las limitaciones en términos cualitativos. Ver Tabla 2. Los valores en el sistema de información determinan la frecuencia con se presenta la limitación, indicándola con una letra. Frecuencia: A (Alta), M (Media), B (Baja), N (Ninguna o No se presenta). El sistema de información consta de 22 pacientes descritos por 70 atributos.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	...	62	63	64	65	66	67	68	69	70
Paciente 1	A	A	A	A	A	A	A	A	A	A	A	A	A	A	...	A	A	N	N	N	A	N	N	A
Paciente 2	N	N	B	N	N	N	N	M	N	B	N	N	M	M	...	B	M	B	A	N	M	A	B	B
Paciente 3	A	A	A	A	A	A	A	A	A	A	M	A	A	...	A	A	A	N	A	A	B	N	A	
Paciente 4	B	B	N	B	B	B	B	B	B	B	N	M	B	...	B	B	B	B	M	N	M	M	M	
Paciente 5	A	A	A	A	A	A	A	A	A	A	N	A	A	...	A	B	B	N	N	A	N	N	N	
Paciente 6	M	M	M	M	M	M	M	B	B	B	N	M	M	...	M	M	N	M	N	A	N	N	B	
Paciente 7	B	B	B	B	B	B	B	N	B	N	N	M	M	...	M	M	N	M	N	N	N	N	N	
...																							
Paciente 18	B	B	B	B	B	B	M	B	B	N	N	M	M	...	A	A	A	A	A	M	A	A	A	
Paciente 19	M	B	M	B	B	B	M	B	B	B	N	M	M	...	A	A	M	A	A	M	A	A	A	
Paciente 20	B	M	M	M	B	B	M	M	B	M	B	N	M	M	...	B	N	B	B	A	A	M	M	A
Paciente 21	N	N	B	N	N	N	N	B	N	N	N	N	M	M	...	M	N	N	B	B	N	B	M	B
Paciente 22	M	M	M	M	B	B	B	M	M	M	B	N	A	A	...	N	N	B	B	B	M	M	M	

Tabla 2: Sistema de información de pacientes.

4 Análisis

Para el análisis del sistema de información fue necesario primero construir la matriz de discernibilidad. La matriz se construye tomando en cuenta las combinaciones de todos los pares de objetos (las primeras dos columnas). En las siguientes columnas se representan valores binarios con tantos elementos como número de atributos existentes. Un 1 en una posición del arreglo indica que este particular atributo distingue entre estos dos pares de objetos. Un 0 indica lo contrario. El orden de los atributos en la matriz de discernibilidad es el mismo que se encuentra en el sistema de información. Parte de la matriz de discernibilidad construida se muestra en la Tabla 3.

Con base en esta representación se desarrolló un algoritmo GRASP para encontrar los reductos.

5 Generación del algoritmo

En esta sección se describe el algoritmo propuesto. Un procedimiento de búsqueda miope aleatorizado y adaptativo (GRASP) [6], es una meta-

		Arreglo de atributos															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...
1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	...
1	3	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	...
1	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	...
1	5	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	...
1	6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	...
...

Tabla 3: Matriz de discernibilidad.

heurística para encontrar soluciones aproximadas a problemas de optimización combinatoria. GRASP es un método multiarranque que consta de dos fases: una fase de construcción en la que se genera una solución factible buena y una fase de búsqueda local, usando la solución construida en la fase previa se examinan vecindades de la solución hasta llegar a un óptimo local. El algoritmo desarrollado se muestra en la Figura 1.

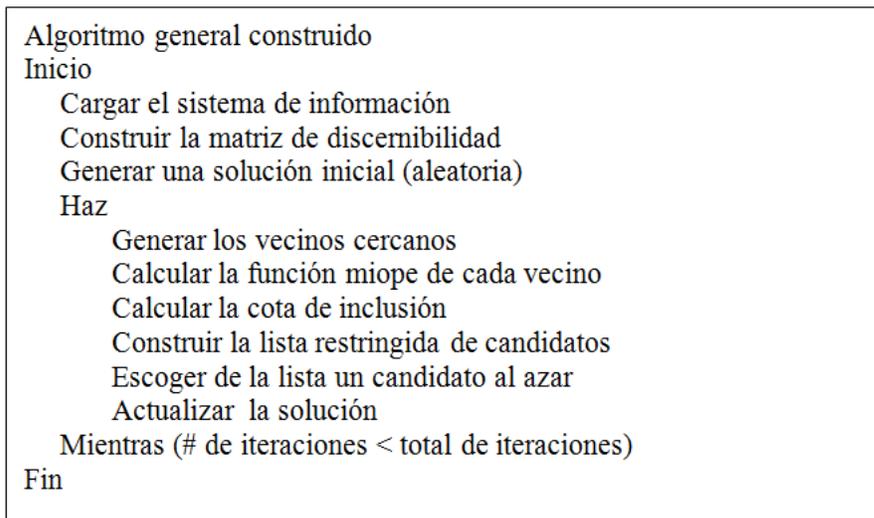


Figura 1: Procedimiento de búsqueda miope aleatorizado y adaptativo.

Cada uno de los pasos en este algoritmo general es un procedimiento. A continuación explicaremos cada procedimiento.

5.1 Cargar el sistema de información

Para el funcionamiento del procedimiento se debe incluir un archivo con el sistema de información como se detalla en la Tabla 2. El procedimiento carga este archivo en una matriz $m \times n$ inicial, donde m es el número total de objetos y n es el número total de atributos que describen a cada objeto.

5.2 Construir la matriz de discernibilidad

La matriz de discernibilidad se construye en base al análisis del sistema de información, este análisis se encuentra en la sección 4 del presente documento. El resultado será una matriz como la que se muestra en la Tabla 3.

5.3 Generar una solución inicial

La primera solución es generada aleatoriamente, esta solución es representada mediante un arreglo unidimensional donde el total de elementos en el arreglo es igual al número total de atributos en el sistema de información. Un valor de 1 en una posición en el arreglo indicará la presencia de ese atributo en la solución. Un valor de 0 indicará lo contrario.

5.4 Generar los vecinos cercanos

En este procedimiento lo que se realiza es generar la lista de vecinos más cercanos. Partiendo de la mejor solución hasta el momento, (en el caso de la primera corrida, la mejor solución será la generada como solución inicial del procedimiento anterior), el primer vecino se construye al intercambiar el valor (de 1 a 0 ó viceversa) del primer atributo de la mejor solución y los demás valores permanecen igual. El segundo vecino será el que se genera de intercambiar el valor del segundo atributo y los demás valores permanecerán igual. Para los siguientes vecinos se seguirá el mismo método hasta generar un total de tantos arreglos (vecinos) como atributos tenga el sistema de información. El procedimiento solo utiliza esta técnica de generación de vecinos.

5.5 Calcular la función miope de cada vecino

Durante la fase de construcción de GRASP cada paso del proceso se genera una solución parcial. Un elemento que pueda seleccionarse como parte de una solución parcial se llama elemento candidato, para determinar que elemento se debe seleccionar, se hace uso de una función miope.

Una función miope mide la contribución local de cada elemento a la solución parcial. En este caso hemos seleccionado parte de la función de

aptitud descrita en [2, 8] y que se utiliza para encontrar reductos mediante algoritmos genéticos. Aplicado a nuestro problema hemos tomado la segunda parte de esta función de aptitud para utilizarla como función miope.

La función de aptitud es la siguiente:

$$F(\vec{v}) = \frac{N - L\vec{v}}{N} + \frac{C\vec{v}}{(m^2 - m)/2}$$

donde

- \vec{v} = candidato reducto
- N = número de atributos
- $L\vec{v}$ = número de 1's en \vec{v}
- $C\vec{v}$ = número de combinaciones de objetos que \vec{v} puede discernir
- m = número de objetos.

La función de aptitud consiste en dos partes. La segunda parte que es la que utilizamos, determina en qué medida el candidato puede discernir entre los objetos. (Un reducto puede discernir entre todos los objetos). Calculamos para cada vecino cercano la función de aptitud, esto servirá para encontrar los mejores candidatos a reducto.

5.6 Calcular la cota de inclusión

Para construir una lista con los mejores candidatos previamente se calcula una cota de inclusión.

La cota de inclusión será la siguiente: $c^* - \alpha(c^* - c_*)$.

c^* y c_* , respectivamente son los valores mayor y menor de la función miope para los elementos candidatos y α es un número real tal que $0 \leq \alpha \leq 1$. Notar que si $\alpha = 0$, este esquema de selección es un algoritmo miope, en tanto que si $\alpha = 1$, entonces es un esquema totalmente aleatorio. Este valor de α es una especie de “porcentaje de tolerancia” donde se pueden aceptar no sólo al mejor elemento candidato, si no también a los valores más cercanos a dicho elemento. Valores usuales de α son 0.1 y 0.2, en nuestro caso seleccionamos $\alpha = 0.2$.

5.7 Construir la lista restringida de candidatos

Dado que el procedimiento de elección mediante el mejor elemento con base en la función miope no tiene aleatoriedad, una forma de introducirla es mediante el uso de una lista restringida de candidatos (LRC) [6]. Esta

lista contiene un conjunto de elementos candidatos con los mejores valores de la función miope.

La lista se construyó con los elementos con los valores de la función miope dentro de un rango dado. La LRC consiste en todos los elementos candidatos e cuyo valor de función miope $c(e)$ es tal que $c(e) \geq$ cota de inclusión. Calcular una cota de inclusión y construir una lista restringida de candidatos forman parte del procedimiento de construcción GRASP: LRC basado en valor, descrito con mayor detalle en [6].

5.8 Escoger de la lista un candidato al azar y actualizar la solución

Una vez construida la LRC se elige un candidato al azar y posteriormente se actualiza al mejor candidato como el candidato seleccionado. Con este nuevo candidato (el mejor hasta el momento) se genera una nueva lista de vecinos cercanos, esta fase será el proceso de búsqueda local, en donde se explora repetidamente la vecindad de una solución en busca de un óptimo local. Todos estos procedimientos forman parte de las dos fases del algoritmo GRASP, tanto construcción como búsqueda local iteran hasta encontrar una buena solución.

6 Resultados experimentales

Se detallan a continuación un par de corridas del algoritmo descrito anteriormente, probando con un sistema de información que contiene únicamente tres objetos y tres atributos, esto debido a que por cuestiones de espacio no es posible incluir arreglos grandes (70 objetos o más) en cada corrida.

Corrida 1.

- Primera solución (Aleatoria): 1 1 0
- Soluciones vecinas y función de aptitud:

Solución vecina	Función de aptitud
0 1 0	1
1 0 0	0.66
1 1 1	1

- La cota de inclusión: 0.93
- La lista restringida de candidatos:

LRC
0 1 0
1 1 1

- Al azar un número de la lista restringida de candidatos: 1
- Solución actualizada: 0 1 0

Corrida 2.

- Solución actualizada: 0 1 0
- Soluciones vecinas y función de aptitud:

Solución vecina	Función de aptitud
1 1 0	1
0 0 0	0
0 1 1	1

- La cota de inclusión: 0.8
- La lista restringida de candidatos:

LRC
1 1 0
0 1 1

- Al azar un número de la lista restringida de candidatos: 1
- Solución actualizada: 1 1 0

En este ejemplo en particular se realizaron pruebas con un total de 10 iteraciones con $\alpha = 0.1$ y 10 iteraciones con $\alpha = 0.2$. Se utilizaron por tanto solo los valores usuales para α y con ambos obtuvimos iguales resultados en la reducción de atributos, por tanto podíamos seleccionar cualquiera de los dos, elegimos para los siguientes ejemplos $\alpha = 0.2$.

Ejecutando el algoritmo con el caso particular del sistema de información de pacientes con demencias (ver Tabla 2) se han encontrado varios subconjuntos mínimos de atributos independientes que aseguran la misma partición que la totalidad del conjunto. Algunos de estos reductos generados se muestran en la Tabla 4.

Se puede notar que cada uno de estos reductos contiene entre 29 y 40 atributos, respecto de los 70 originales, esto supone en cierta medida una reducción importante de atributos redundantes, que en algunos casos es de hasta más de la mitad.

El algoritmo propuesto fue también probado con otros ejemplos, en particular el descrito en [2]. En él se considera un sistema de información

	Reductos
R1=	1 0 1 1 1 1 0 0 1 1 0 0 1 1 1 0 0 1 0 0 0 0 0 1 1 1 1 1 1 1 0 0 1 0 1 1 1 0 0 0 1 0 1 1 0 1 1 1 1 1 1 0 1 1 0 1 0 0 0 1 0 1 0 1 0 0 1 1 1 1 0
R2=	1 0 0 0 1 1 1 1 1 1 0 1 1 0 0 1 1 1 1 0 1 1 0 1 0 0 1 0 1 1 1 1 0 1 0 0 1 1 0 1 0 1 1 0 0 0 1 1 0 1 1 0 0 0 0 1 0 1 0 0 0 1 1 0 0 0 1 0 0 0
R3=	1 1 0 1 0 0 0 1 1 0 0 0 1 0 1 0 0 0 0 1 0 0 1 1 0 0 0 0 1 0 1 0 1 1 0 0 0 0 0 1 1 1 0 0 0 1 1 1 1 1 0 0 1 0 0 0 0 0 1 1 0 1 0 1 0 1 0 0 0 1
R4=	0 1 0 0 1 0 0 0 1 1 1 0 1 0 1 0 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 1 1 0 0 1 1 1 0 1 0 1 0 1 1 0 1 0 0 1 0 0 0 1 0 1 1 1 1 1 0 0 1 0 1 0 1 0 0
R5=	1 0 1 1 0 1 0 1 1 1 0 0 1 1 0 0 0 1 1 0 1 1 0 0 1 1 0 0 1 0 1 0 1 1 0 1 1 0 1 0 1 1 1 1 0 0 0 0 1 1 1 1 1 1 0 1 0 0 0 1 1 0 0 0 0 0 1 1 1 1 1

Tabla 4: Ejemplos de reductos encontrados.

describiendo algunos perros (en la Tabla 5 se encuentran los resultados de este SI 2). En ese trabajo utilizaron algoritmos genéticos. El ejemplo fue probado con el algoritmo desarrollado encontrando exactamente el mismo reducto (Reducto= 1 1 0) que con algoritmos genéticos. Cabe mencionar que ese reducto encontrado es el único dentro de la lista de candidatos a reducto.

6.1 Comparación de tiempos

El tiempo de cómputo probando el algoritmo propuesto para algunos ejemplos se muestra en la Tabla 5. Como se puede ver en esta tabla el tiempo de cómputo es pequeño, ninguno de los ejemplos de SI sobrepasa un segundo.

	m	N	T
SI 1	3	3	0.098
SI 2	4	3	0.109
SI 3	16	35	0.166
SI 4	22	70	0.591

Tabla 5: Comparativo de tiempos. m : número de objetos, N : número de atributos y T : tiempo de cómputo en segundos para 50 iteraciones con un $\alpha = 0.2$.

7 Conclusiones y trabajo futuro

Este trabajo propone un procedimiento GRASP para encontrar reductos mínimos, utilizando además parte de una función de aptitud formulada para trabajar con algoritmos genéticos. Es un método alternativo para

encontrar reductos procesando información tanto cualitativa como cuantitativa.

Más experimentos e investigación futura en este procedimiento son necesarios. Una de las acciones que continúan en este proyecto consiste en encontrar el menor reducto para generar reglas de diagnóstico que apoyen la clasificación de niveles de demencias en los pacientes.

Reconocimientos

Este trabajo forma parte de la investigación que está desarrollando Iris Iddaly Méndez Gurrola para obtener el grado de Doctor en Diseño, Línea: Nuevas Tecnologías de la Universidad Autónoma Metropolitana-Azcapotzalco. Además forma parte del proyecto Computación Suave y Aplicaciones financiado por la misma Universidad.

Referencias

- [1] Bjorvand, A.T. (1997) “Rough enough – A system supporting the rough sets approach”, *Sixth Scandinavian Conference on Artificial Intelligence*, Helsinki, Finland: 290–291.
- [2] Bjorvand, A.T.; Komorowski, J. (1997) “Practical applications of genetic algorithms for efficient reduct computation”, *Wissenschaft & Technik Verlag* **4**: 601–606.
- [3] Lezak, M.D.; Howieson, D.B.; Loring, D.D.; Hannay, H.J.; Fisher, J.S. (2004) *Neuropsychological Assessment*. Oxford University Press, New York.
- [4] Méndez-Gurrola, I.I.; Laureano-Cruces, A.L.; Sánchez de Antuñano, J.; de Arriga-Gómez, F. (2009) “Aplicaciones de apoyo a pacientes con Alzheimer”, *XXII Congreso Nacional y VIII Congreso Internacional de Informática y Computación de la ANIEI*, Ensenada, Baja California: 337–342.
- [5] Pawlak, Z.; Skowron, A. (2007) “Rudiments of rough sets”, *Information Sciences* **177**(1): 3–27.
- [6] Resende, M.G.C.; González Velarde, J.L. (2003) “GRASP: Greedy randomized adaptive search procedures”, *Revista Iberoamericana de Inteligencia Artificial* **19**: 61–76.
- [7] Segovia-Vargas, M.J.; Gil-Fana, J.A.; Heras-Martínez, A.; Vilar-Zañon, J.L. (2002) “Predicción de insolvencias con el método Rough Set”, *X Jornadas de Asepuma*, Madrid.

- [8] Wróblewski, J. (1995) “Finding minimal reducts using genetic algorithm”, *Proc. of the Second Annual Joint Conference on Information Sciences*, Wrightsville Beach, NC: 186–189.