

Análisis Comparativo de Métodos Metaheurísticos de Clasificación

Trabajo de Tesis
presentado al
Departamento de Ingeniería Eléctrica y Electrónica

por

Adriana Marcucci B.

Asesor: Alain Gauthier S.

Para optar al título de
Maestría en Ingeniería Eléctrica

Ingeniería Eléctrica y Electrónica
Universidad de Los Andes
Enero de 2006

Análisis Comparativo de Métodos Metaheurísticos de Clasificación

Aprobado por:

Alain Gauthier S., Asesor

Hernando Durán

Fidel Torres

Fecha de Aprobación _____

A Juan Fernando, por lo vivido y por lo aprendido...

Agradecimientos

Gracias a Ángela y Alain por creer en mi y porque sin ustedes no habría maestría...

A mis papás y mis hermanas por estar ahí...

A Juan Fernando y Sandra por su colaboración incondicional...

Y a mis compañeros de AOM's por el apoyo y la paciencia.

Tabla de Contenido

Dedicatoria	iii
Agradecimientos	iv
Lista de Tablas	viii
Lista de Figuras	ix
Resumen	xi
I Introducción	1
II Métodos de Clasificación	3
2.1 División de los métodos de Clasificación	3
2.2 K-Means - KM	5
2.3 C-Means - CM	6
2.4 Gustafson Kessel Means - GKM	6
III Metaheurísticas	7
3.1 Búsqueda Tabú	7
3.2 Enfriamiento Simulado	8
3.3 Algoritmos Genéticos	10
3.4 Colonia de Hormigas	10
IV Métodos Metaheurísticos de clasificación con Solución Matriz de Pertenencia	12
4.1 Generación de la solución inicial	12
4.2 Número variable de clases	13
4.3 Generación de Vecinos	13
4.4 Generación de la vecindad	14
4.5 Actualización de función objetivo	14

4.6	Convergencia	15
4.7	Descripción de Algoritmos	16
4.7.1	Búsqueda Tabú - TS-U [15][37][35]	16
4.7.2	Enfriamiento Simulado - SA-U [12][18]	16
4.7.3	Algoritmos Genéticos - GA-U [39]	16
4.7.4	Colonia de Hormigas - ACO-U [32][22][38][33]	17
V	Métodos Metaheurísticos de clasificación con Solución Centros de Clase	19
5.1	Función objetivo	19
5.2	Actualización de matriz de pertenencia y centroides	19
5.3	Generación de la solución inicial	20
5.4	Generación de Vecinos	20
5.5	Generación de la vecindad	21
5.6	Descripción de Algoritmos	21
5.6.1	Búsqueda Tabú [10][3]	21
5.6.2	Enfriamiento Simulado	21
5.6.3	Algoritmos Genéticos [7]	21
VI	Implementación de los algoritmos	23
VII	Evaluación de desempeño de las técnicas metaheurísticas	30
7.1	Función objetivo y error de clasificación	30
7.2	Tiempos de convergencia	32
7.3	Comportamiento del tiempo de ejecución en función del número de individuos a clasificar	34
VIII	Aplicaciones de los algoritmos	36
8.1	Aplicación a Clasificación de Flores Iris	36
8.1.1	Función objetivo y error de clasificación	37
8.1.2	Tiempos de convergencia	38
8.2	Aplicación a Diagnóstico de fallas	39
8.2.1	Descripción del sistema	39

8.2.2	Datos Utilizados	40
8.2.3	Resultados obtenidos	40
8.3	Aplicación a Clasificación de Empresas del sector eléctrico	42
8.3.1	Learning Algorithm for Multivariate Data Analysis - LAMDA	42
8.3.2	Descriptores utilizados	44
8.3.3	Resultados	44
IX	Conclusiones	46
APÉNDICE A	— Análisis de descriptores	49
	Referencias	50

Lista de Tablas

1	Resultados con Base de datos Geométrica	31
2	Resultados con Base de datos Iris	37
3	Resultados de clasificación de empresas de transmisión	45

Lista de Figuras

1	División de Métodos de Clasificación según su procedimiento	4
2	Clases de Mitra	24
3	Diagrama de clases de métodos metaheurísticos solución matriz de pertenencia	25
4	Diagrama de clases de métodos metaheurísticos solución centroide	27
5	Diagrama de clases de métodos metaheurísticos difusos	28
6	Interfaz Gráfica de Mitra 1.3	29
7	Base de datos Geométrica	30
8	Resultados Clasificación Datos Geométricos. (a)KM, (b)CM, (c)TS-U, (d)SA-U, (e)GA-U, (f)ACO-U, (g)TS-Cent, (h)SA-Cent, (i)GA-Cent, (j)TS-dif, (k)SA-dif, (l)GA-dif, (m)GKM	32
9	Convergencia de F.O. con Matriz de Pertenencia para Geométrica	33
10	Convergencia de F.O. con Centros de Clase para Geométrica	33
11	Convergencia de F.O. con solución Difusa para Geométrica	33
12	Tiempo de Ejecución vs. Individuos con Matriz de Pertenencia	34
13	Tiempo de Ejecución vs. Individuos con Centros de Clase	35
14	Tiempo de Ejecución vs. Individuos con solución Difusa	35
15	Base de datos Iris	36
16	Convergencia de F.O. con Matriz de Pertenencia para Geométrica	38
17	Convergencia de F.O. con Centros de Clase para Geométrica	38
18	Convergencia de F.O. con solución Difusa para Geométrica	39
19	Disposición de los tanques de prueba	39
20	Datos Utilizados para la detección de fallas en Tanques (a) Nivel del tanque (b) Derivada del nivel del tanque 1	40
21	Resultados de métodos determinísticos en detección de fallas. (a)Datos, (b)TS-U, (c)SA-U, (d)GA-U, (e)ACO-U, (f) Datos, (g)KM, (h)TS-Cent, (i)SA-Cent, (j)GA-Cent	41

22 Resultados de métodos difusos en detección de fallas. (a)Datos, (b)CM, (c)TS-Fuzzy, (d)SA-Fuzzy, (e)GA-Fuzzy 41

Resumen

En este documento se presenta la aplicación de técnicas metaheurísticas de optimización al problema de clasificación o clustering de datos.

Se implementaron dos tipos de métodos, los cuales emplean diferentes representaciones de la solución, en el primer tipo se usa la matriz de pertenencia y, en el segundo los centros de clases. Los métodos que tienen como representación de la solución la matriz de pertenencia son determinísticos y los que emplean los centros de clase son tanto determinísticos como difusos.

Se realiza una comparación del desempeño de los métodos metaheurísticos de clasificación con K-means y C-Means.

Se presentan los resultados de los métodos metaheurísticos de clasificación para tres aplicaciones: clustering de flores tipo iris, diagnóstico de fallas en un sistema hidráulico y agrupamiento de empresas de energía eléctrica.

CAPÍTULO I

Introducción

El problema de clasificación consiste en buscar el agrupamiento de objetos, descritos mediante un vector de atributos, ya sea construyendo las clases o asignando los objetos a clases previamente definidas. En muchos casos, los algoritmos de clasificación son métodos en los cuales se lleva a cabo la optimización (minimización) de cierto criterio de disimilitud. Por esta razón, se hace interesante evaluar la utilización de algunos métodos de optimización metaheurísticos para resolver problemas de clustering.

El objetivo principal de esta tesis es desarrollar algoritmos en los cuales se emplean técnicas metaheurísticas de optimización para la solución de problemas de clasificación, implementar dichos algoritmos y analizar su desempeño.

Se implementaron dos clases de métodos, los cuales difieren en la representación de las soluciones:

1. Métodos con representación Matriz de pertenencia: En este grupo se desarrollaron métodos determinísticos, es decir, métodos en los cuales un individuo pertenece a una sola clase. Las metaheurísticas empleadas son:
 - (a) Búsqueda Tabú
 - (b) Enfriamiento Simulado
 - (c) Algoritmos Genéticos
 - (d) Colonia de Hormigas
2. Métodos con representación Centros de Clase: En este grupo se desarrollaron métodos, tanto determinísticos como difusos; estos últimos son métodos en los cuales cada individuo tiene un nivel de pertenencia a cada una de las clases. Las metaheurísticas empleadas son:

- (a) Búsqueda Tabú
- (b) Enfriamiento Simulado
- (c) Algoritmos Genéticos

Para la evaluación de los algoritmos metaheurísticos de optimización se hace una comparación con otras técnicas de clasificación: K-means (KM) y C-Means (CM).

En la siguiente sección se hace una presentación de los fundamentos de los diferentes métodos de clasificación, posteriormente se presentan los algoritmos desarrollados y se hace una evaluación de los métodos, en cuanto a tiempos de convergencia y función objetivo encontrada; y, por último, se emplean los métodos en tres aplicaciones: clustering de flores tipo iris, diagnóstico de fallas en un sistema hidráulico y agrupamiento de empresas de energía eléctrica. Adicionalmente se presenta un capítulo en el que se hace una descripción de la implementación de los algoritmos.

CAPÍTULO II

Métodos de Clasificación

Una clase representa un conjunto de individuos descritos mediante un vector de n atributos [21]. El objetivo de los métodos de clasificación es realizar de manera automática el agrupamiento de los individuos en dichas clases. [27][26]

En general, sea $X = \{x_1, x_2, \dots, x_N\}$ un conjunto con N vectores (individuos) en un espacio n -dimensional (n atributos), y sea Ω un conjunto con c clases $\Omega = \{\omega_1, \omega_2, \dots, \omega_c\}$, tal que $2 \leq c \leq N$; entonces, los métodos de clasificación definen una función de X a Ω [21]. Para definir esta función, se tienen en cuenta ciertas medidas de semejanza entre los datos, tales como distancia, intensidad y conectividad, entre otras. [27][26]

Al realizar la clasificación se busca que cada clase contenga individuos que sean similares entre sí, e individuos de clases distintas deben ser “no similares”. El criterio utilizado es, generalmente, la optimización de una función objetivo relacionada con las semejanzas entre los datos que sirve como índice del desempeño de la clasificación. No obstante, tal como explica Trejos et. al., “muchos autores han propuesto métodos para realizar la partición de los datos, sin embargo, muchos de ellos son variaciones de búsquedas locales, con el inconveniente de que estos métodos pueden quedar atrapados en óptimos locales.” [37]

Cabe resaltar que la clasificación de individuos tiene un amplio rango de aplicaciones [28], entre las que se encuentran: diagnóstico médico y psicológico, aplicaciones en economía, supervisión y diagnóstico de fallas en sistemas automáticos complejos y modelaje de sistemas no lineales, entre otros.

2.1 División de los métodos de Clasificación

Existen varias divisiones de los métodos de clasificación, entre las que se encuentran:

1. Supervisada y No supervisada:

- Clasificación supervisada: En este tipo de clasificación se tiene un conjunto de datos de entrenamiento con clases previamente definidas y el objetivo es establecer unas reglas para asignar los individuos a una de esas clases.
- Clasificación no supervisada: En este tipo de clasificación las clases se determinan a partir de ciertas características de los datos con los que se cuenta. Lo que se busca es agrupar a los individuos que tengan la mayor similitud entre sí.

2. Según el procedimiento: En la figura 1 se presenta una división de los métodos teniendo en cuenta el procedimiento de clasificación.

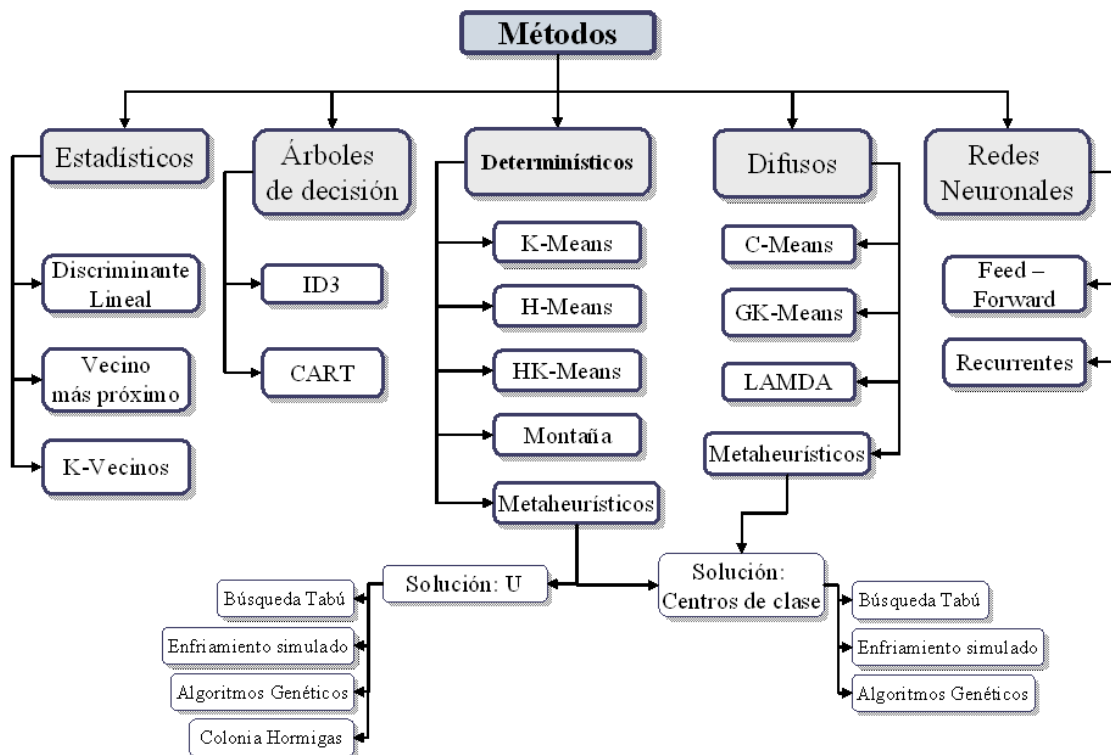


Figure 1: División de Métodos de Clasificación según su procedimiento

- (a) Estadísticos: La clasificación estadística se realiza, en la mayoría de los casos, a partir de la suposición de que los atributos tienen una distribución de probabilidad que depende de la clase del individuo [1]. Uno de los métodos más conocidos es el discriminante lineal de Fisher [20].

- (b) árboles de decisión: Es un método supervisado, en el cual la clasificación se realiza a partir de una secuencia de preguntas en la cual la siguiente pregunta depende de la respuesta actual. El procedimiento se hace desde la raíz hacia las hojas y en cada nodo se evalúa un atributo y se baja por la rama asociada al valor de la instancia. El proceso se repite hasta llegar a una hoja en donde se encuentra la clase a asignar al individuo.[1]
- (c) Determinísticos y difusos: Una división importante de los algoritmos de clasificación es aquella que los separa entre determinísticos y difusos. En los algoritmos del primer grupo cada individuo pertenece exclusivamente a una clase, mientras que en los difusos existen grados de pertenencia de los individuos a las clases.

Para representar la pertenencia de los individuos a las clases, se emplea una matriz de pertenencia U , de dimensión $c \times N$, en la que el elemento u_{ij} indica la pertenencia del j -ésimo individuo a la clase i y está dado por:

- Métodos determinísticos:

$$u_{ij} = \begin{cases} 1 & \text{si el } j\text{-ésimo individuo } \in \text{ a la clase } i \\ 0 & \text{de lo contrario} \end{cases} \quad (1)$$

- Métodos difusos:

$$u_{ij} \in [0, 1] \quad (2)$$

donde 1 implica pertenencia total, y 0 ninguna pertenencia.

- (d) Redes neuronales artificiales: Pretenden imitar el funcionamiento del cerebro humano en cuanto a su estructura y el funcionamiento de cada neurona [5].

A continuación se describen tres algoritmos de clasificación, que se utilizarán para comparar el desempeño de los métodos metaheurísticos desarrollados.

2.2 K-Means - KM

Es un método de clasificación determinístico no supervisado que permite agrupar en c clases un conjunto de individuos. El objetivo de este algoritmo es obtener, con una población de N datos, los centroides de cada una de las c clases $\{\omega_1, \omega_2, \dots, \omega_c\}$, y a partir de estos centroides clasificar los datos asignándole la clase del centroide más cercano. El procedimiento se basa en

la minimización de una función objetivo de la forma [4] [34]:

$$\sum_{i=1}^c J_i = \sum_{i=1}^c \sum_{j=1}^N u_{ij} d_{ij}^2 \quad (3)$$

donde d_{ij} es la distancia euclidiana entre el individuo x_j y el centroide i y u_{ij} es la pertenencia del j -ésimo individuo a la clase i . Como se emplea la distancia euclidiana las clases generadas tienen forma de hiperesferas.

2.3 C-Means - CM

Es un método de clasificación difuso no supervisado propuesto por Bezdek en 1973 como una mejora del algoritmo K-means, y que asocia a cada individuo cierto nivel de pertenencia a cada una de las c clases. Este nivel de pertenencia se representa mediante la matriz U en la que el elemento u_{ij} es el grado de pertenencia del j -ésimo individuo a la clase i .

El proceso llevado a cabo en este método se basa en la minimización de una función objetivo que mide la distancia entre los individuos de una misma clase, dada por [4] [34]:

$$\sum_{i=1}^c J_i = \sum_{i=1}^c \sum_{j=1}^N u_{ij}^m d_{ij}^2 \quad (4)$$

donde d_{ij} es la distancia euclidiana entre el individuo x_j y el centroide de la clase i y m es el grado de difusión entre clases, que se define exógenamente. Debido al tipo de distancia empleada, la agrupación se realiza en hiperesferas.

2.4 Gustafson Kessel Means - GKM

GK-Means es una modificación hecha por Gustafson y Kessel en 1979 al algoritmo CM [4] que busca minimizar la misma función objetivo de C-means (ver ec. 4), pero en lugar de usar la distancia euclidiana emplea una norma adaptiva, dada por [34]:

$$d_{ij} = (x_j - c_i)^T A_i (x_j - c_i) \quad (5)$$

donde A_i es una matriz definida positiva, conocida como matriz de norma adaptiva, que se adapta de acuerdo a la forma de cada grupo. Este tipo de distancia empleada se conoce como distancia de Mahalanobis [4], cuyo uso permite tener clusters con forma de hiperelipses rotadas con respecto a los ejes del espacio de datos.

CAPÍTULO III

Metaheurísticas

3.1 Búsqueda Tabú

Esta técnica metaheurística de optimización tiene sus orígenes hacia los años 70, pero oficialmente su nombre y metodología fueron introducidos por Fred Glover en 1989 [14].

El algoritmo se basa en la idea básica de que “es mejor una mala decisión basada en información que una buena decisión al azar” [16]. Para esto se emplea una memoria, llamada lista Tabú, la cual contiene información sobre soluciones anteriores o movimientos que se han realizado en iteraciones anteriores. De esta manera, la lista Tabú impide que el algoritmo vuelva sobre soluciones previamente visitadas, restringiendo la vecindad de las soluciones.

El algoritmo de Búsqueda Tabú para la minimización de una función objetivo se describe a continuación [16]:

1. Inicializar la lista Tabú: T .
2. Elegir una solución inicial x_0 , con función objetivo $F(x_0)$.
3. Inicializar la solución actual y la solución óptima:
 $x = x_0$
 $x^* = x_0$
4. Buscar la vecindad de la solución actual x : $N(x)$
5. Encontrar el conjunto tabú de la vecindad de la solución, es decir, los vecinos de x que son Tabú:
 $T(x) = N(x) \cap T$
6. Encontrar el conjunto aspirante de la vecindad de la solución, es decir, las soluciones Tabú cuya función objetivo es mejor que la función objetivo de x^* :

$$A(x) = \{x_i : x_i \in T(x) \wedge F(x_i) < F(x^*)\}$$

7. Definir la vecindad restringida como la vecindad sin el conjunto Tabú pero incluyendo el conjunto aspirante

$$N^*(x) = N(x) - T(x) + A(x)$$

8. Escoger como solución actual la mejor solución de la vecindad restringida $N^*(x)$: \hat{x} . Esta solución puede ser mejor o peor que la mejor solución encontrada (x^*):

$$x = \hat{x}$$

9. Almacenar en la lista Tabú información relacionada con la movida realizada para alcanzar la nueva solución actual, o información sobre la solución anterior. Cuando la lista Tabú se llena, se debe comenzar a eliminar elementos, para lo cual existen varias formas: eliminar el más antiguo, el que tenga la mejor función objetivo, el que tenga la peor función objetivo, entre otras.

10. Actualizar la mejor solución encontrada:

$$\text{Si } F(x) < F(x^*) \Rightarrow x^* = x$$

11. Si no se cumple el criterio de parada volver al paso (4).

3.2 Enfriamiento Simulado

Este algoritmo, propuesto por Kirkpatrick en los años 70, simula el proceso de enfriamiento aplicado a los metales después de haberles realizado modificaciones a altas temperaturas [13]. Para realizar este proceso, se comienza con una solución inicial a una temperatura alta, la cual se va disminuyendo, a la vez que se realizan un número determinado de iteraciones por cada nivel de temperatura. En todas las iteraciones, a partir de una solución actual, se busca la vecindad y el mejor vecino. Si la función objetivo del mejor vecino es peor que la función objetivo actual, ésta se acepta con una probabilidad proporcional a la temperatura actual e inversamente proporcional al tamaño de la desmejora. Una vez concluidas las iteraciones correspondientes a la temperatura actual, ésta se disminuye, disminuyendo así la probabilidad de aceptar peores soluciones en el futuro. De esta forma, el algoritmo permite una intensa diversificación en las primeras etapas de búsqueda, que luego da paso a una intensificación de la solución en las últimas etapas [36].

El algoritmo de enfriamiento simulado para un problema de minimización se presenta a continuación [13]:

1. Definir la temperatura inicial t_0 , la temperatura final t_f ($t_f \ll t_0$), el porcentaje de disminución de temperatura α y el número de repeticiones por temperatura rep_t .
2. Elegir una solución inicial x_0 , con función objetivo $F(x_0)$.
3. Inicializar la solución actual, la solución óptima y la temperatura actual:
 $x = x_0$ y $F(x) = F(x_0)$
 $x^* = x_0$ y $F(x^*) = F(x_0)$
 $t_{\text{actual}} = t_0$
4. $i = 0$
While $i < rep_t$
 - Buscar la vecindad de la solución actual x : $N(x)$
 - Buscar la mejor solución de la vecindad:
 $\hat{x} \in N(x)$
 - $\Delta = F(x) - F(\hat{x})$
 - **if** $\Delta > 0$
 $x = \hat{x}$
if $F(x) < F(x^*) \Rightarrow x^* = x$
 - else**
 $p = e^{-\Delta/t_{\text{actual}}}$
Generar $u = \text{unif}(0, 1)$
if $u \leq p \Rightarrow x = \hat{x}$
 - $i = i + 1$
EndWhile
 $t_{\text{actual}} = \alpha t_{\text{actual}}$
5. Actualizar la mejor solución encontrada:
Si $F(x) < F(x^*) \Rightarrow x^* = x$
6. **If** $t_{\text{actual}} > t_{\text{final}}$ volver al paso (4).
else END

3.3 Algoritmos Genéticos

Esta metaheurística fue desarrollada por Holland en 1975. Es un algoritmo que simula el proceso de selección natural que se lleva a cabo en la evolución de los seres vivos [29]. A diferencia de los anteriores algoritmos, en éste se trabaja con un grupo de soluciones que constituyen una población. Cada una de estas soluciones tiene una representación conocida como cromosoma, al igual que en la naturaleza, es decir, una codificación de la solución que se lleva a cabo mediante una serie de genes.

En algoritmos genéticos se realiza la evolución de una población inicial, aplicando a la población de cada generación cuatro operadores: Elitismo, Selección, Cruce y Mutación; y generando, de esta manera, la siguiente población [29].

Estos operadores se describen a continuación:

1. **Elitismo:** Se define un porcentaje de la población actual para que continúe en la siguiente generación. Estos individuos son aquellos con la mejor función objetivo.
2. **Selección:** Para el proceso de cruce se debe realizar una selección previa de los padres, en la cual se garantice que sean los individuos más adaptados de la población los que se reproduzcan.
3. **Cruce:** Una vez se eligen los padres, se lleva a cabo el cruce, cuyo objetivo es crear dos nuevos individuos (hijos) de dos individuos existentes (padres). El objetivo es que los hijos tengan información combinada de los padres. En el caso en el que se tenga codificación de las soluciones mediante vectores, se pueden generar al azar uno o dos puntos de cruce mediante los cuales se intercambian algunos genes de los cromosomas de los padres.
4. **Mutación:** Una vez se ha generado la nueva población el último paso es la mutación. Este proceso consiste en mutar al azar algunos de los individuos de la población, con una probabilidad de mutación predeterminada. Así se genera un nuevo individuo a partir de un solo padre y se refuerza la diversificación del conjunto de soluciones.

3.4 Colonia de Hormigas

Esta metaheurística, desarrollada en los 90 por Marco Dorigo, pretende simular el comportamiento de las hormigas reales en la búsqueda de alimento, las cuales encuentran el mejor camino

desde el hormiguero hasta la fuente de comida sin tener información visual del recorrido, usando una hormona llamada *feromona* [11]. Al ir en busca de alimento, cada hormiga deja un rastro de feromona que le sirve para volver al hormiguero y para guiar al resto de hormigas. Con el paso del tiempo, los caminos con mayor feromona son aquellos por los que más hormigas han pasado, es decir, en los que se encuentran las mejores fuentes de alimento.

Este mecanismo de búsqueda de comida es el que se imita para resolver problemas de optimización. En Colonia de Hormigas, al igual que en algoritmos genéticos, se tiene una población de soluciones, conocida como colonia, en la cual cada hormiga representa una solución y tiene asociada una función objetivo. Cada hormiga construye un camino de feromona asociado a la bondad de la solución, búsqueda que realiza cada hormiga en cada iteración. Poco a poco, aquellos caminos (soluciones) que ofrecen una mejor función objetivo tienen más feromona.

El algoritmo de la metaheurística se presenta a continuación [11]:

1. Se crea una colonia inicial y se inicializa la feromona de la colonia, τ .
2. Cada hormiga realiza una búsqueda.
3. Se hace una actualización de la feromona, la cual, según la bondad de cada solución, se intensifica o evapora.
4. Se vuelven a ubicar las hormigas, teniendo en cuenta la feromona de la colonia.
5. Se repiten los pasos 2-4 hasta cumplir con un criterio de convergencia.

CAPÍTULO IV

Métodos Metaheurísticos de clasificación con Solución Matriz de Pertenencia

Estos son métodos determinísticos de clasificación en los cuales, para representar las soluciones, se emplea la matriz de pertenencia de los individuos a las clases (U) de dimensión $c \times N$, en la cual el elemento u_{ij} indica la pertenencia del j -ésimo individuo a la clase i , y está dado por:

$$u_{ij} = \begin{cases} 1, & \text{si el } j\text{-ésimo individuo } \in \text{ a la clase } i \\ 0, & \text{de lo contrario} \end{cases} \quad (6)$$

Para cada una de las clases que se generen se tiene un centro de la clase (centroide), el cual se calcula como la media sobre todos los atributos de los individuos que pertenecen a dicha clase, es decir, los centroides o prototipos son vectores de promedios por atributo para cada una de las c clases. La función de objetivo para los algoritmos determinísticos de clasificación metaheurística, está dada por la suma sobre todas las clases de las distancias entre los centroides y los individuos de cada clase, ésta es la misma función objetivo de KM y está dada por:

$$\sum_{i=1}^c J_i = \sum_{i=1}^c \sum_{j=1}^N u_{ij} d_{ij}^2 \quad (7)$$

donde d_{ij} es la distancia euclidiana entre el individuo x_j y el centroide i , y u_{ij} es la pertenencia del j -ésimo individuo a la clase i .

4.1 Generación de la solución inicial

Para inicializar los métodos en una solución avanzada, la cual tenga una función objetivo mejor que una solución iniciada completamente al azar, se usa el siguiente algoritmo:

1. Se hace una partición del espacio de decisión, dividiendo cada atributo en tantas partes como número de clases se quieran generar, obteniendo $numClases^n$ posibles regiones del espacio de decisión.

2. La ubicación de los centroides iniciales se elige aleatoriamente de las regiones del espacio, sin permitir que haya centroides en la misma región.
3. Con los centroides iniciales se calcula la matriz de pertenencia inicial, asignando cada individuo a la clase más cercana.

4.2 Número variable de clases

Los métodos metaheurísticos generados permiten fijar un número máximo de clases (c_{max}) y un número mínimo de individuos por clase ($minIndi$), facilitando de esta forma la obtención del número óptimo de clases, pues no es necesario conocerlas a priori. Para esto, cada vez que se genera un vecino se verifica que en cada clase activa haya el número mínimo de individuos por clase, a través del siguiente algoritmo:

i=0

while i< c_{max}

- if ω_i activa
 - Calcular número individuos de clase i ($numIndi_i$).
 - if $numIndi_i < minIndi \Rightarrow$ desactivar clase y poner individuos en la clase más cercana
- $i = i + 1$

end while

4.3 Generación de Vecinos

Los vecinos se generan moviendo individuos de una clase a otra. Los individuos a cambiar se eligen al azar, y el procedimiento que se lleva a cabo para cada uno de estos individuos es:

1. Se busca la clase en la que el individuo que se va a cambiar presenta el mejor valor de la función objetivo : ω_i
2. if $\omega_i \neq \omega_{actual} \Rightarrow \omega_{actual} = \omega_i$
 - else
 - p = Probabilidad de diversificación para clases
 - Generar $u = unif(0, 1)$
 - if $u \leq p \Rightarrow \omega_{actual} = \omega_{rand}(\omega_{rand} \neq \omega_{actual})$

4.4 Generación de la vecindad

Las vecindades se generan definiendo un porcentaje de vecinos que tienen un número grande de individuos que cambian de clase; y en el porcentaje restante se generan vecinos en lo que se cambian de clase pocos individuos.

4.5 Actualización de función objetivo

La función objetivo de estos métodos se puede reescribir como:

$$\sum_{i=1}^c \sum_{j=1}^{N_i} d_{ij}^2 \quad (8)$$

donde d_{ij} es la distancia euclidiana entre el individuo $x_j : x_j \in \omega_i$, y el centroide i .

La función objetivo de la ecuación 8 se reescribe nuevamente, teniendo en cuenta que el individuo j y el centroide de ω_i son vectores con n elementos:

$$\sum_{i=1}^c \sum_{j=1}^{N_i} \sum_{k=1}^n (x_j^{(k)} - c_i^{(k)})^2 = \sum_{i=1}^c \sum_{k=1}^n \sum_{j=1}^{N_i} (x_j^{(k)} - c_i^{(k)})^2 \quad (9)$$

donde $x_j^{(k)}$ es el atributo k del individuo j que pertenece a ω_i y $c_i^{(k)}$ es el atributo k del centroide de la clase i .

Dado que el centroide de las clases es el vector de medias por atributo de los individuos que pertenecen a dicha clase, se obtiene una expresión final para la función objetivo, dada por:

$$\sum_{i=1}^c \sum_{k=1}^n \sum_{j=1}^{N_i} (x_j^{(k)} - \bar{x}_i^{(k)})^2 \quad (10)$$

Teniendo en cuenta que para un conjunto de N observaciones x_1, x_2, \dots, x_N , la varianza estimada está dada por la ecuación 11 [6], la función objetivo se puede expresar en términos de esta varianza, tal como se muestra en la ecuación 12.

$$S^2 = \frac{\sum_{i=1}^N (x_i - \bar{x})^2}{N - 1} \quad (11)$$

donde x_i es cada una de las observaciones, \bar{x} es la media de las observaciones y N es el número de observaciones.

$$\sum_{i=1}^c \sum_{k=1}^n (N_i - 1) S_i^{2(k)} \quad (12)$$

donde $S_i^{2(k)}$ es la varianza estimada del atributo k en la clase i .

Al tener expresada la función objetivo en términos de la varianza, para hacer actualizaciones de la función objetivo solo se necesita modificar la varianza. En este tipo de métodos, los cambios de solución implican cambios en la clase del individuo j ; este cambio en la solución modificará la varianza de dos clases: la clase a la que pertenecía el individuo y la nueva clase a la que pertenece. Para calcular dicha modificación se tiene en cuenta otra de las definiciones de varianza estimada [6], dada por la ecuación:

$$S^2 = \overline{x^2} - \bar{x}^2 \quad (13)$$

Teniendo en cuenta esta expresión, para la actualización de la función objetivo solo se necesita actualizar las medias y las medias al cuadrado de las clases que se ven afectadas con el cambio de clase del individuo.

$$\bar{x}_{\text{nueva}} = \frac{N\bar{x}_{\text{anterior}} \pm x_{\text{cambia}}}{N \pm 1} \quad (14)$$

donde \bar{x}_{nueva} es la media modificada, $\bar{x}_{\text{anterior}}$ es la media antes del cambio, x_{cambia} es el individuo que cambia de clase y N es el número de individuos de la clase antes del cambio. El signo $+$ es para la clase a la que cambia el individuo, y el $-$ para la clase de la que sale.

4.6 Convergencia

Para garantizar la convergencia de estos métodos, cada cierto número de iteraciones todos los individuos se asignan a la clase del centroide más cercano, mediante el siguiente procedimiento:

$i=0$

while $i < \text{número individuos}$

- Asignar individuo x_i a la clase más cercana
- Actualizar medias, medias al cuadrado y función objetivo
- $i = i + 1$

4.7 Descripción de Algoritmos

4.7.1 Búsqueda Tabú - TS-U [15][37][35]

Además de las características mencionadas anteriormente sobre la función objetivo, la actualización de dicha función y la generación de vecindades, para búsqueda Tabú se definió la lista Tabú. En la lista Tabú se guardan los elementos que han cambiado de clase, teniendo en cuenta que los individuos están cambiando a la mejor clase posible. Para que una solución sea Tabú se define un porcentaje de movimientos contenidos en las soluciones de la lista Tabú.

4.7.2 Enfriamiento Simulado - SA-U [12][18]

Con base en la descripción de la vecindad y la actualización de la función objetivo, se implementó el algoritmo SA-U.

4.7.3 Algoritmos Genéticos - GA-U [39]

- Elitismo: Se emplea el operador de elitismo, se define un número de individuos que se desea como población élite.
- Selección: Se emplea el método de selección de la ruleta[17], en el cual la probabilidad de que una solución sea escogida como padre es inversamente proporcional a la función objetivo (se debe tener en cuenta que es un problema de minimización).
- Cruce: Se emplea el operador de cruce por un punto, mediante el cual a partir de un padre (U_{padre}) y una madre (U_{madre}) se generan dos hijos, de la siguiente manera:
 1. Se define $U(a, b)$ como la matriz generada por las columnas de a hasta b de la matriz U .
 2. Generar $u = \text{unif}(0, 1)$
 3. Generar punto de cruce: $pCru = N \times u$
 4. El hijo 1 está dado por:

$$U_{\text{hijo1}} = [U_{\text{padre}}(0, pCru - 1) | U_{\text{madre}}(pCru, N - 1)]$$
 5. El hijo 2 está dado por:

$$U_{\text{hijo2}} = [U_{\text{madre}}(0, pCru - 1) | U_{\text{padre}}(pCru, N - 1)]$$
- Mutación: Para la mutación se fija a priori una probabilidad de mutación y se emplea el procedimiento descrito para la generación de vecinos.

4.7.4 Colonia de Hormigas - ACO-U [32][22][38][33]

Cada una de las hormigas se inicializa usando el método mostrado en la sección anterior. Adicionalmente, se tiene una matriz de feromona τ de $c \times N$, en la cual cada elemento τ_{ij} indica que tan bueno es para la función objetivo que el individuo j pertenezca a ω_i .

El algoritmo de clasificación ACO-U es el siguiente:

1. Se inicializa la matriz de feromona asignándole a cada elemento el mismo valor $\tau_{ij} = \frac{1}{c}$.
2. Se genera una colonia inicial de k hormigas, cada una de las cuales tiene asociada una función objetivo F_j .
3. Se actualiza la matriz de feromona, definiendo a priori un factor de evaporación de la feromona (ρ). Para esto se tienen 3 métodos diferentes:
 - τ_{ij} depende de la función objetivo del conjunto de las L mejores hormigas en las que x_j en ω_i (H_L):

$$\tau_{ij}^{(t+1)} = (1 - \rho)\tau_{ij}^{(t)} + \sum_{l=1}^L \frac{1}{F_l}$$
 donde F_l es la función objetivo de la hormiga $l \in H_L$
 - τ_{ij} depende del número de hormigas para las cuales cambiar x_j a una clase $\omega_k \neq \omega_i$ mejora la función objetivo (llamado N_{cambio}):

$$\tau_{ij}^{(t+1)} = (1 - \rho)\tau_{ij}^{(t)} + (\rho)\frac{N_{cambio}}{N}$$
 - τ_{ij} depende de cuantas de las L mejores hormigas (H_L), tienen a x_j en ω_i (esta cantidad se denomina N_{Lij})

$$\tau_{ij}^{(t+1)} = (1 - \rho)\tau_{ij}^{(t)} + (\rho)\frac{N_{Lij}}{N}$$
4. Se genera una nueva colonia con base en la información de la feromona, para lo cual se fija una probabilidad de intensificación (q_0):
 - Generar $u = \text{unif}(0, 1)$
 - if $u < q_0$, en la nueva hormiga los individuos pertenecen a la clase con mayor feromona (ω_{mayor}).
 - else: El nuevo individuo se asigna aleatoriamente a $\omega_{rand} \neq \omega_{mayor}$.
5. Mutación de las hormigas: Para la mutación se fija a priori una probabilidad de mutación y se emplea el procedimiento descrito para la generación de vecinos. Sin embargo, si la

hormiga mutada presenta una función objetivo peor que la original, se descarta la hormiga mutada.

6. Se repiten los pasos 3-5 hasta cumplir con un criterio de convergencia.

CAPÍTULO V

Métodos Metaheurísticos de clasificación con Solución Centros de Clase

Estos son métodos de clasificación, tanto determinísticos como difusos, en los cuales, para representar las soluciones, se emplean los centros de las clases. Cada una de estas soluciones tiene asociada una matriz de pertenencia, ya sea de tipo difuso o determinístico, dependiendo del método.

5.1 Función objetivo

1. Métodos determinísticos:

La función objetivo de estos métodos, es la misma que la empleada en los métodos con solución matriz de pertenencia, presentada en la ecuación 7.

2. Métodos difusos:

La función objetivo de estos métodos está dada por:

$$\sum_{i=1}^c \sum_{j=1}^N u_{ij}^m d_{ij} \quad (15)$$

donde d_{ij} es la distancia euclidiana entre el individuo x_j y el centroide de la clase i y m es el nivel de difusión entre clases.

5.2 Actualización de matriz de pertenencia y centroides

1. Métodos determinísticos:

(a) Matriz de pertenencia:

$j=0$

while $j < N$

- Para todas las clases, buscar el centroide más cercano al individuo x_j : ω_k
- $u_{kj} = 1$
- $j = j + 1$

(b) Centros de clase:

Una vez se actualiza la matriz de pertenencia los centroides se mueven a las medias de los datos que pertenecen a cada clase.

2. Métodos difusos:

(a) Matriz de pertenencia [4]:

$$u_{ij} = \left(\sum_{k=1}^c \left(\frac{d_{ij}}{d_{ik}} \right)^{\frac{2}{m-1}} \right)^{-1} \quad (16)$$

donde d_{ij} es la distancia entre x_j y c_i , y d_{ik} es la distancia entre x_k y c_i .

(b) Centros de clase [4]:

Una vez se actualiza la matriz de pertenencia se mueven los centros de clase a:

$$c_i = \frac{\sum_{j=1}^N u_{ij}^m x_j}{\sum_{j=1}^N u_{ij}^m}, \quad \forall i = 1, \dots, c \quad (17)$$

5.3 Generación de la solución inicial

La solución inicial se determina con el procedimiento descrito para los métodos con solución Matriz de Pertenencia.

5.4 Generación de Vecinos

Los vecinos se generan moviendo cada uno de los atributos de los centroides mediante una distribución normal. La media de dicha distribución es 0 y la desviación estándar es p veces el rango de cada atributo. Para los vecinos con un cambio grande $p = 2$, y para los que tienen cambio pequeño $p = 1$. Esto teniendo en cuenta que, en una normal en el rango $(\mu - \sigma, \mu + \sigma)$ se acumula una probabilidad aproximada de 66% y en $(\mu - 2\sigma, \mu + 2\sigma)$ se acumula una probabilidad aproximada de 95%. El procedimiento que se lleva a cabo para cada centroide es:

k=0

while $k < n$

- Se calcula la desviación estándar: $\sigma = p * rango_k$
- Se genera un número aleatorio $w = N(0, \sigma^2)$ empleando el método polar para la generación de números normales. [23].
- $x_{nuevo}^{(k)} = x^{(k)} + w$
- Se revisa que el nuevo atributo del centroide se encuentre dentro del rango del atributo
- $k = k + 1$

end while

5.5 Generación de la vecindad

Las vecindades se generan definiendo un porcentaje de los vecinos que tienen un cambio grande en los centroides ($\sigma = 2$ rango atributo), y el porcentaje restante de vecinos se generan cambiando poco los centroides ($\sigma =$ rango atributo).

5.6 Descripción de Algoritmos

5.6.1 Búsqueda Tabú [10][3]

La lista tabú contiene las soluciones, es decir, los centroides visitados anteriormente. Para que una nueva solución sea Tabú, se define un área alrededor de cada centroide tabú y se revisa si los centroides de la nueva solución se encuentran en esta área; la solución es tabú si un porcentaje predefinido de los centroides es tabú.

El método determinístico se denomina TS-Cent y el difuso TS-Fuzzy.

5.6.2 Enfriamiento Simulado

Con base en la descripción de la vecindad y la actualización de la función objetivo se implementaron los algoritmos SA-Cent y SA-Fuzzy

5.6.3 Algoritmos Genéticos [7]

- Elitismo: se emplea el operador de elitismo, se define un número de individuos que será la población élite.
- Selección: se emplea el método de selección de ruleta, descrito anteriormente.

- Cruce: se emplea el operador de cruce por promedio ponderado, mediante el cual a partir de un padre con centros de clase (c_{padre}), y una madre con centros de clase (c_{madre}), se generan dos hijos, así:

Para cada atributo:

- Generar $u = \text{unif}(0, 1)$

- El centroide k del hijo 1 está dado por:

$$c_{\text{hijo1}}^{(k)} = u * c_{\text{padre}}^{(k)} + (1 - u) * c_{\text{madre}}^{(k)}$$

- El centroide k del hijo 2 está dado por:

$$c_{\text{hijo2}}^{(k)} = u * c_{\text{madre}}^{(k)} + (1 - u) * c_{\text{padre}}^{(k)}$$

- Mutación: Para la mutación, se fija a priori una probabilidad de mutación y se emplea el procedimiento descrito para la generación de vecinos.

Los algoritmos que se implementaron son GA-Cent y GA-Fuzzy.

CAPÍTULO VI

Implementación de los algoritmos

Los algoritmos desarrollados fueron implementados en Java 1.5, como complemento a un Toolbox de clasificación desarrollado en el Centro de Investigación de la Facultad de Ingeniería de la Universidad de los Andes, MITRA 1.3 [25]. Para el desarrollo complementario se hizo hincapié en la orientación a objetos del software, que permite una fácil extensión del Toolbox a través de las clases desarrolladas, en particular, las relacionadas con la representación de soluciones, vecindades y poblaciones, así como los métodos de utilidades.

Una aproximación al diagrama de clases se presenta en la fig 2. En esta figura se puede ver que en el paquete se tienen 10 clases, las cuales se pueden dividir en dos grupos:

- Clases con los métodos de clasificación
 1. Estadísticos: Métodos que implementan los métodos Fisher y K-Vecinos.
 2. árboles: Método que implementa CART.
 3. Determinísticos: Métodos que implementan K-means y el método de la montaña.
 4. Difusos: Tiene los métodos que implementan C-means, GK-means y LAMDA.
 5. MetaheurísticosU: Métodos que implementan los algoritmos metaheurísticos de clasificación con representación de la solución tipo matriz de pertenencia: TS-U, SA-U, GA-U y ACO-U. Adicionalmente, tiene unos métodos auxiliares para manejar las soluciones con matriz de pertenencia (solU), los vecinos (vecinoU), y las poblaciones (poblacionU).
En la figura 3 se presenta el diagrama de clases para los métodos metaheurísticos con solución matriz de pertenencia.
 6. MetaheurísticosCent: Métodos que implementan las metaheurísticas de clasificación con representación de la solución con centroides y de tipo determinístico: TS-Cent,

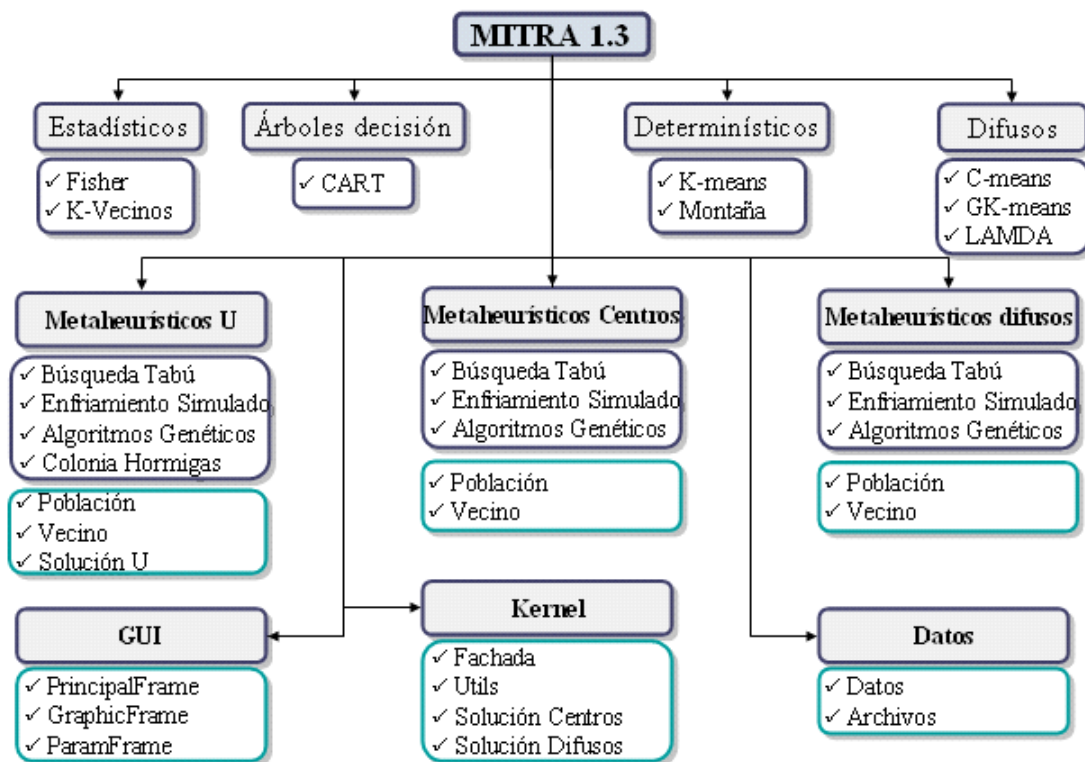


Figure 2: Clases de Mitra

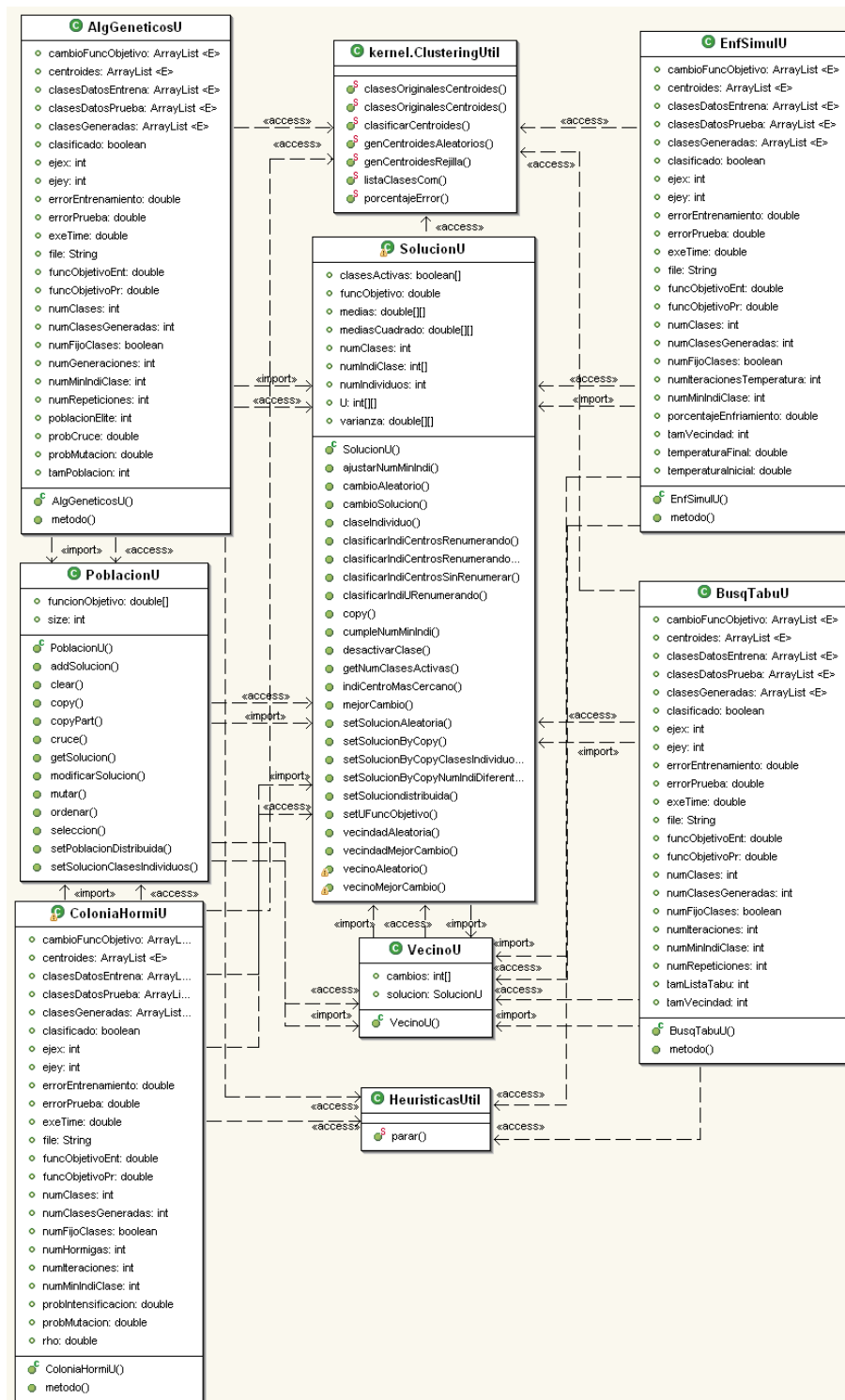


Figure 3: Diagrama de clases de métodos metaheurísticos solución matriz de pertenencia

SA-Cent y GA-Cent. Adicionalmente, tiene unos métodos auxiliares para manejar los vecinos (vecinoCent) y las poblaciones (poblacionCent).

En la figura 4 se presenta el diagrama de clases para los métodos metaheurísticos con solución centroide.

7. Metaheurísticos Dif: Métodos que implementan los algoritmos metaheurísticos de clasificación con representación de la solución centroides y de tipo difuso: TS-Fuzzy, SA-Fuzzy y GA-Fuzzy. Adicionalmente, tiene unos métodos auxiliares para manejar los vecinos (vecinoDif) y las poblaciones (poblacionDif).

En la figura 5 se presenta el diagrama de clases para los métodos metaheurísticos difusos.

- Clases auxiliares

1. GUI: Es la interfaz gráfica del programa, en la cual se tiene: PrincipalFrame, que es la interfaz principal del programa; ParamFrame, interfaz para modificar los parámetros de todos los algoritmos; y dos frames para gráficas: uno para gráficas en 2D y otro para las gráficas de individuos contra clases. En la figura 6 se presenta la interfaz gráfica del software.
2. Kernel: En esta clase se encuentran los métodos: Fachada, la cual maneja todo el programa; diferentes clases de utilidades, como ArrayUtil, NumberUtil, MatrixUtil y ClusterUtil; y por último, los métodos para manejar las soluciones con representación centros de clase, tanto determinísticas (solCent) como difusas (solDif).
3. Datos: Contiene métodos para cargar datos y generar archivos.

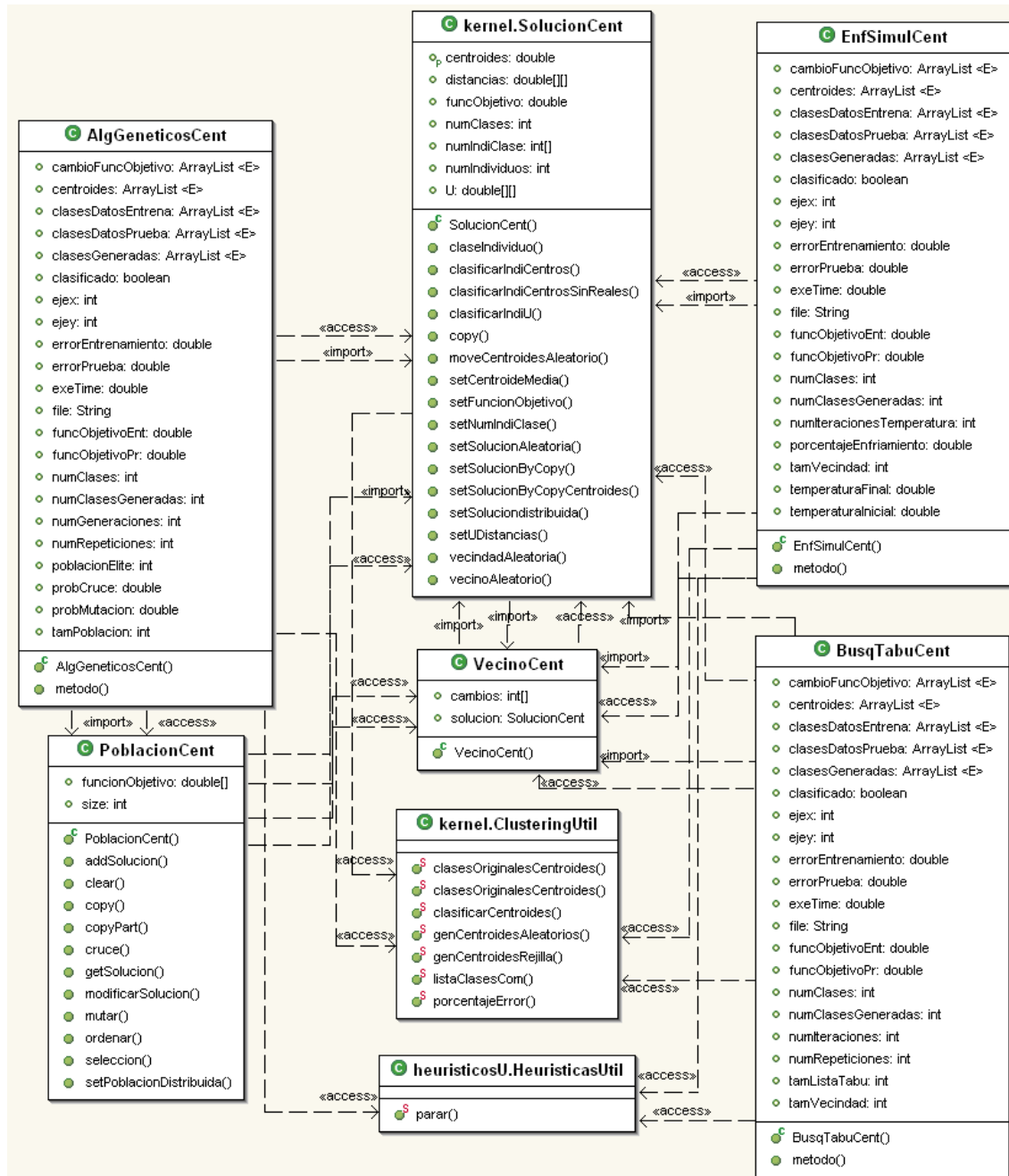


Figure 4: Diagrama de clases de métodos metaheurísticos solución centroe

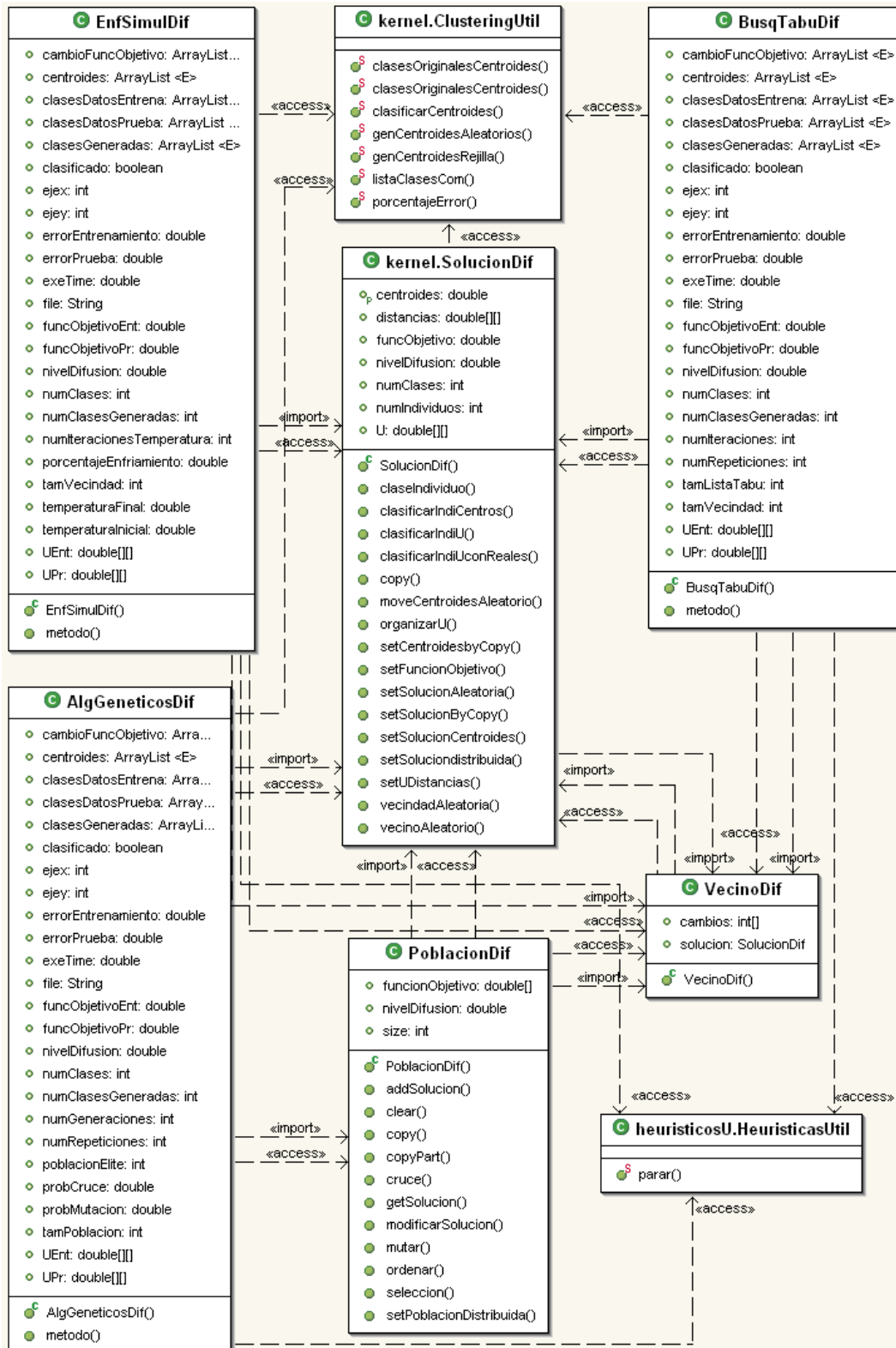


Figure 5: Diagrama de clases de métodos metaheurísticos difusos

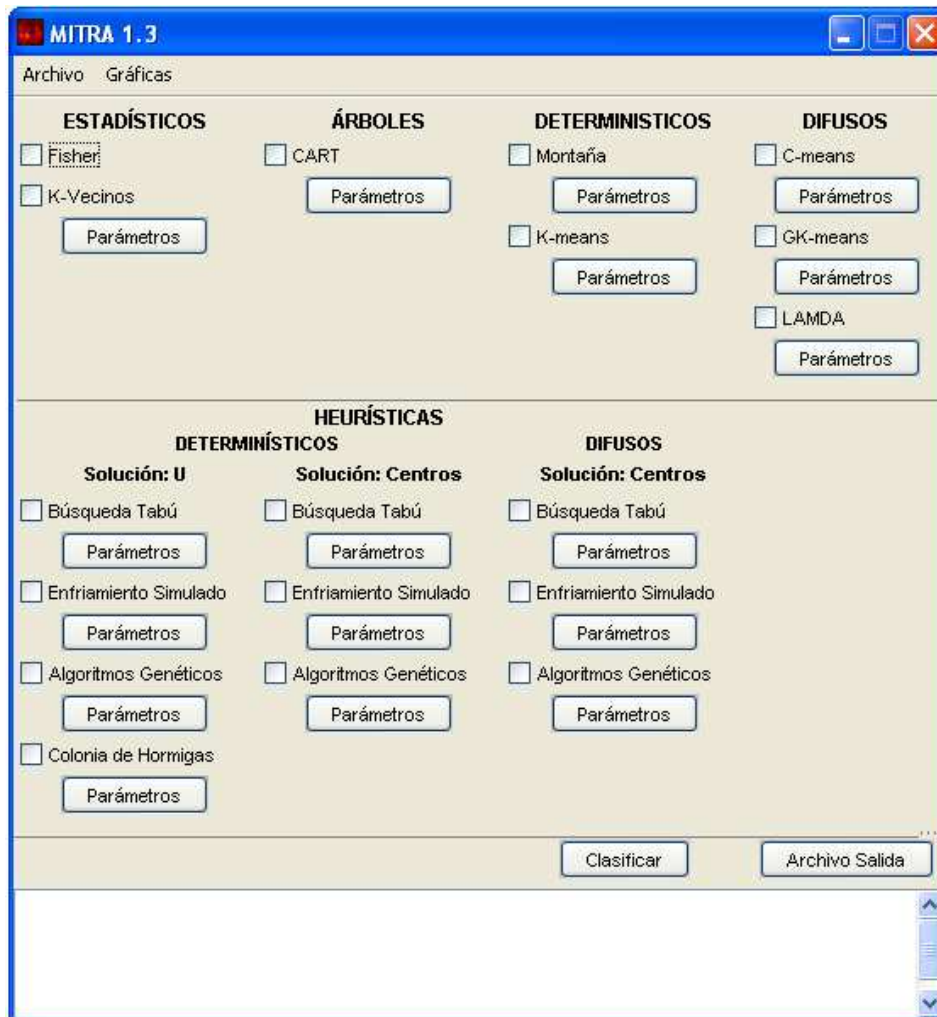


Figure 6: Interfaz Gráfica de Mitra 1.3

CAPÍTULO VII

Evaluación de desempeño de las técnicas metaheurísticas

La evaluación de los algoritmos se realizó en un computador con procesador Pentium IV de 2.8 GHz, mononucleo y con 512 MB de RAM.

Para la evaluación de desempeño de los métodos de clasificación se utilizó una base de datos llamada Geométrica, la cual tiene individuos con dos atributos y tres clases con formas geométricas diferentes: círculo, cuadrado y elipse, cada una de ellas con 2000 individuos. En la fig 7 se presenta la gráfica de los datos y se puede ver que las clases son separables entre sí.

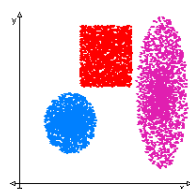


Figure 7: Base de datos Geométrica

Con estos datos se realizó un análisis de la función objetivo encontrada, los porcentajes de error en las clases obtenidas y los tiempos de convergencia. Para la evaluación, se realiza una comparación con los métodos KM y CM, pues en los métodos metaheurísticos se están empleando las mismas funciones objetivo que utilizan estos dos métodos.

7.1 Función objetivo y error de clasificación

Para esta prueba se separaron los datos en un conjunto de entrenamiento (60% de los datos) y uno de prueba (40% de los datos). En aras de lograr resultados comparables para los métodos implementados, se requiere utilizar la misma solución como solución inicial.

En la tabla 1 se presentan los resultados obtenidos. En primer lugar, se puede apreciar que los porcentajes de error y las funciones objetivo encontradas son similares tanto para los algoritmos

determinísticos como para los difusos. En cuanto a los primeros métodos, se puede ver que todos los algoritmos metaheurísticos generan errores y funciones objetivo menores o iguales que las obtenidas con KM, y aunque los tiempos de ejecución son mayores que los de KM, ninguno supera los 13s. En cuanto a los métodos difusos, los errores de entrenamiento y prueba encontrados para los metaheurísticos son mayores que los de CM, aunque es de resaltar que, si bien los errores son mayores, la función objetivo encontrada es menor.

Por último, se puede ver que el error de prueba de GKM es menor que todos los errores encontrados, debido a la posibilidad que tiene de encontrar clases con formas diferentes a hipersferas.

Algoritmo	Error datos Ent (%)	Error datos prueba (%)	Tiempo ejecución (s)	Función Objetivo Ent	Función Objetivo Prueba
TS-U	0.889	1	3.603	65433.561	44322.061
SA-U	0.889	1	6.005	65433.561	44322.061
GA-U	0.861	1	4.790	65433.431	44321.524
ACO-U	0.833	1	12.502	65433.730	44322.069
KM	0.889	1	0.266	65432.518	44322.063
TS-Cent	0.889	1	4.807	65432.518	44322.063
SA-Cent	0.889	1	7.302	65432.639	44320.968
GA-Cent	0.889	1	4.650	65432.518	44322.063
CM	1.167	1.458	0.375	48109.781	32195.119
TS-Fuzzy	1.222	1.5	19.034	48102.492	32188.282
SA-Fuzzy	1.222	1.5	9.168	48102.492	32188.288
GA-Fuzzy	1.222	1.5	13.551	48102.569	32187.771
GKM	1.056	0.708	0.345		

Tabla 1: Resultados con Base de datos Geométrica

En la figura 8 se presentan las gráficas de los individuos en cada clase para cada método. Se puede ver que las gráficas son similares entre si, y que los errores se presentan en los límites de las clases, especialmente en la identificación de la clase elíptica. Si se comparan estos resultados con los obtenidos con GKM (fig. 8(m)) se ve que efectivamente el menor error de clasificación se debe a la capacidad de encontrar diferentes formas de clases.

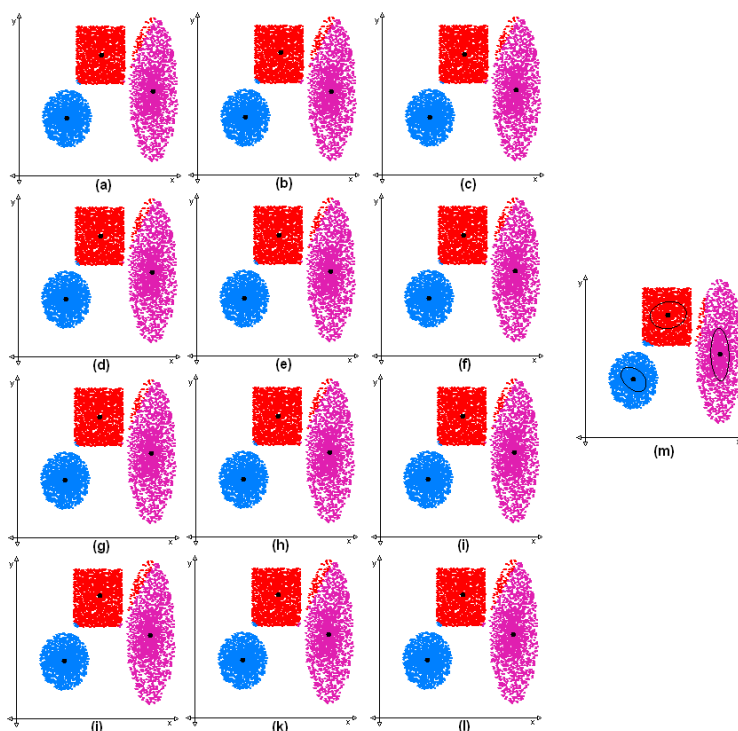


Figure 8: Resultados Clasificación Datos Geométricos. (a)KM, (b)CM, (c)TS-U, (d)SA-U, (e)GA-U, (f)ACO-U, (g)TS-Cent, (h)SA-Cent, (i)GA-Cent, (j)TS-dif, (k)SA-dif, (l)GA-dif, (m)GKM

7.2 Tiempos de convergencia

Para la anterior prueba, en la que se separaron los datos en entrenamiento y prueba, la figura 9 presenta el tiempo de convergencia para cada uno de los métodos. En esta gráfica se puede ver que ACO es el algoritmo que tarda más en comenzar a converger, y se aprecia que todos los algoritmos empiezan a estabilizarse, en tiempos similares, alrededor de la solución óptima. También es de notar que la operación en la que se asignan los individuos a la clase más cercana genera mejoras considerables en la función objetivo, los cuales se aprecian como cambios bruscos de la evolución.

En cuanto a los métodos metaheurísticos que usan solución centroide, se puede notar que tienen mayor tiempo de convergencia que los de solución matriz de pertenencia, y que todos tienen una evolución similar en su funciones objetivo (Ver fig. 10 y 11).

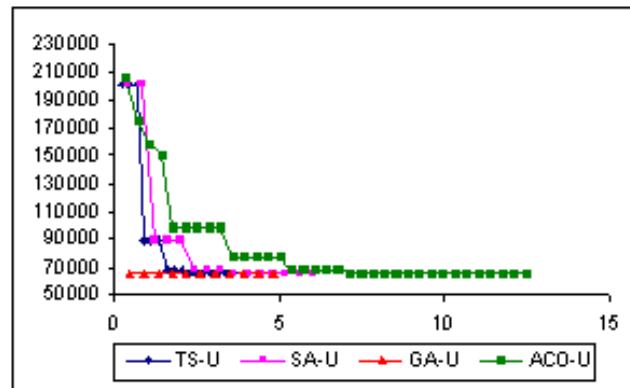


Figure 9: Convergencia de F.O. con Matriz de Pertenencia para Geométrica

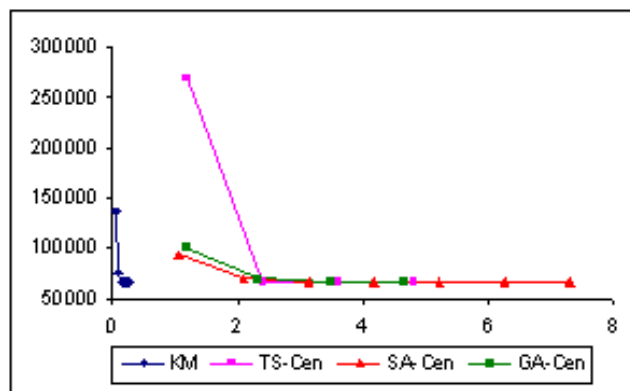


Figure 10: Convergencia de F.O. con Centros de Clase para Geométrica

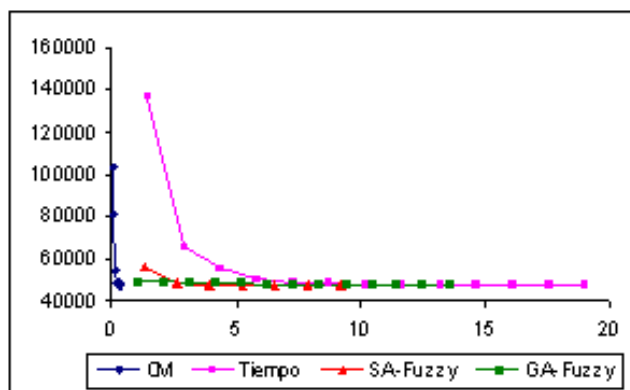


Figure 11: Convergencia de F.O. con solución Difusa para Geométrica

7.3 Comportamiento del tiempo de ejecución en función del número de individuos a clasificar

Con la base de datos geométrica, se usaron diferentes porcentajes de datos de entrenamiento, para probar los algoritmos variando el número de individuos. Las pruebas se realizaron para: 100, 200, 500, 1000, 2000, 3000, 4000 y 5000 individuos. En las figuras 12, 13 y 14 se presenta, para los diferentes métodos, el cambio en el tiempo de ejecución al variar el número de individuos. Se puede notar que existe una tendencia de crecimiento del tiempo de convergencia al aumentar el número de individuos, especialmente para los métodos de clasificación metaheurísticos con solución centros de clase. Por otro lado, se aprecian comportamientos particulares de ACO-U y GA-Cen, lo cual muestra la sensibilidad de estos métodos a las poblaciones iniciales elegidas. Aquellos métodos que no requieren de una población inicial, sino de una única solución, se iniciaron en la misma solución para obtener resultados comparables.

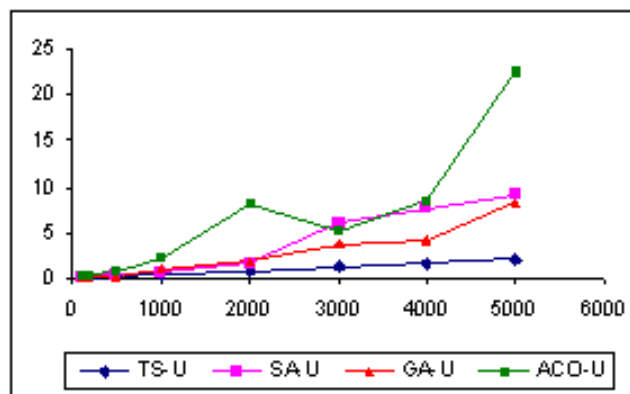


Figure 12: Tiempo de Ejecución vs. Individuos con Matriz de Pertenencia

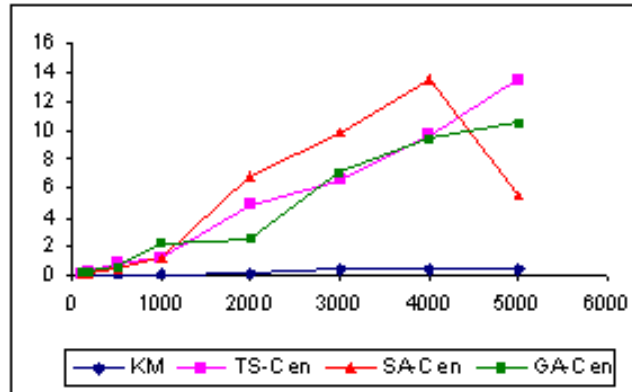


Figure 13: Tiempo de Ejecución vs. Individuos con Centros de Clase

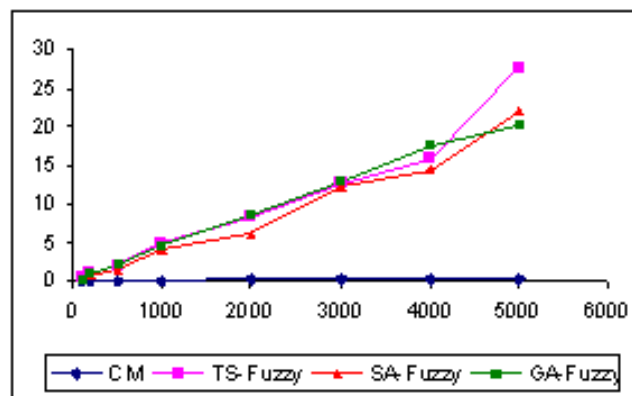


Figure 14: Tiempo de Ejecución vs. Individuos con solución Difusa

CAPÍTULO VIII

Aplicaciones de los algoritmos

8.1 Aplicación a Clasificación de Flores Iris

Las flores Iris son un Benchmark disponible en el repositorio de la Universidad de California[31] con características de diferentes tipos de flores iris: Iris Setosa, Iris Versicolor e Iris Virgínica. Tiene 150 individuos, 33.3% de cada una de las clases, descritos con 4 atributos cuantitativos: Longitud del sépalo en centímetros, ancho del sépalo en centímetros, longitud del pétalo en centímetros y ancho del pétalo en centímetros. Una de las clases es linealmente separable de las otras dos, en tanto que éstas dos no son separables entre sí, como se puede ver en la fig. 15.

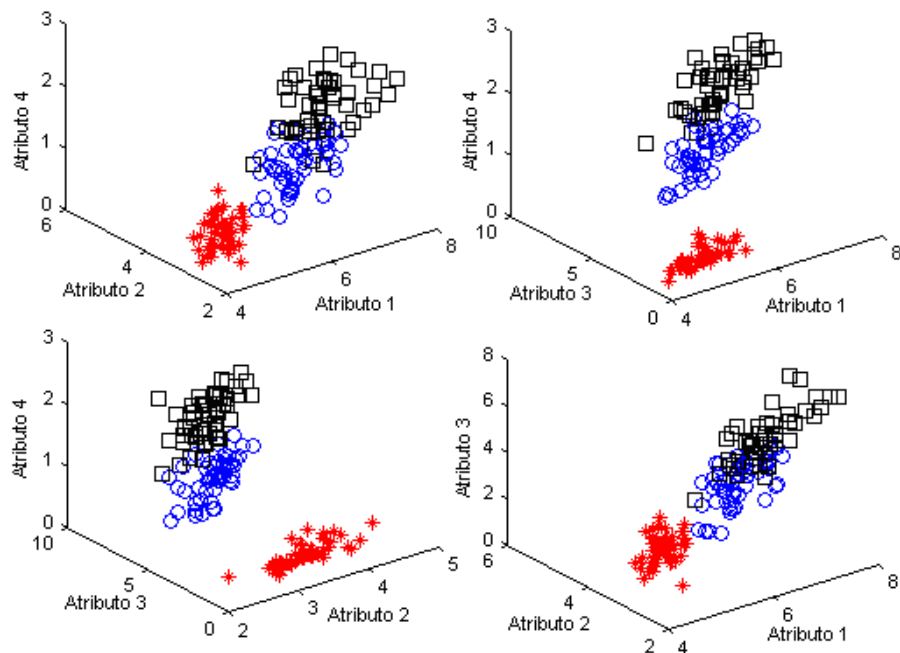


Figure 15: Base de datos Iris

8.1.1 Función objetivo y error de clasificación

En la tabla 2 se presentan los resultados comparativos de los métodos de clasificación. Para estos resultados se separaron los datos en un conjunto de entrenamiento (60% de los datos) y conjunto de prueba (40% de los datos). Por consiguiente, se presentan los porcentajes de error y las funciones objetivo para los dos conjuntos.

Para realizar estas pruebas se usó la misma solución inicial para los métodos que tienen una única solución aleatoria inicial, para tener tiempos de ejecución comparables entre si.

Algoritmo	Error datos Ent (%)	Error datos prueba (%)	Tiempo ejecución (seg)	Función Objetivo Ent	Función Objetivo Prueba
TS-U	8.889	13.333	0.073	46.379	34.457
SA-U	8.889	13.333	0.128	46.379	34.457
GA-U	8.889	13.333	0.104	46.379	34.457
ACO-U	5.556	11.667	0.090	47.671	36.981
KM	10	13.333	0.000	46.448	33.410
TS-Cent	8.889	13.333	0.176	46.379	34.457
SA-Cent	10	13.333	0.146	46.448	33.410
GA-Cent	5.556	10	0.290	46.666	35.629
CM	7.778	13.333	0.018	35.373	25.885
TS-Fuzzy	8.889	11.667	0.637	35.361	25.795
SA-Fuzzy	8.889	11.667	0.607	35.361	25.799
GA-Fuzzy	8.889	11.667	0.461	35.361	25.788

Tabla 2: Resultados con Base de datos Iris

Es de notar que los métodos se comportan de manera bastante similar, aunque se puede apreciar una particularidad en ACO-U, ya que, aunque tiene las funciones objetivo más altas, presenta los menores errores. Esto hace pensar que la función objetivo con clases hiperesféricas pueden presentar inconvenientes en bases de datos en las que no se tienen clases con esa forma. Por otro lado, es AG-Cent el método con el menor error de entrenamiento y prueba, mostrando las ventajas de tener una población de soluciones y no una única solución.

En esta base de datos, en la cual las clases no son muy separables, se puede apreciar que algunos de los métodos metaheurísticos tienen ventajas, al hacer una exploración más amplia del espacio

de solución.

8.1.2 Tiempos de convergencia

En la tabla 8.1.1 se puede ver que los tiempos de convergencia se encuentran en un rango entre 0.063 y 0.637 segundos, esto muestra que aunque los algoritmos metaheurísticos tienen un mayor tiempo de convergencia, las diferencias no son demasiado grandes.

En las figuras 16, 17 y 18 se presenta la evolución de la función objetivo contra el tiempo para cada uno de los métodos de clasificación. Se puede observar que los métodos con un mayor tiempo de convergencia son los difusos, aunque esto puede deberse fundamentalmente al criterio de parada empleado. Por otro lado, ACO-U es el algoritmo que más tarda en acercarse a la solución óptimo. Al igual que para la prueba con la base de datos geométrica se puede ver que todos los algoritmos metaheurísticos de clasificación tienden al óptimo en tiempos semejantes.

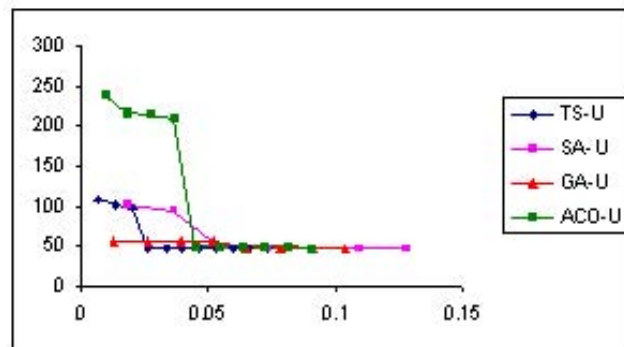


Figure 16: Convergencia de F.O. con Matriz de Pertenencia para Geométrica

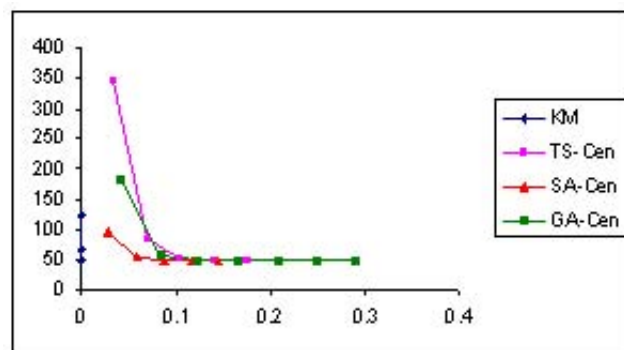


Figure 17: Convergencia de F.O. con Centros de Clase para Geométrica

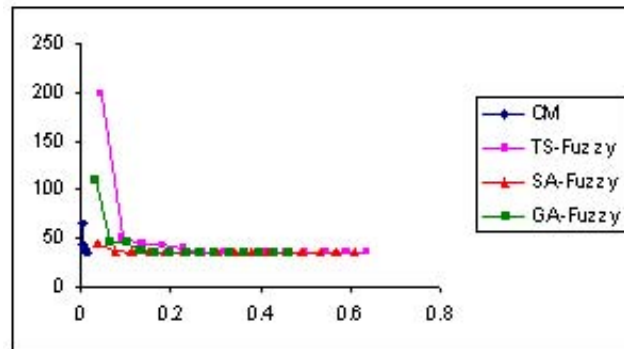


Figure 18: Convergencia de F.O. con solución Difusa para Geométrica

8.2 Aplicación a Diagnóstico de fallas

El diagnóstico de un sistema se asocia a la detección de errores, de malos funcionamientos o de defectos. éste consiste en un bloque de razonamiento al cual entra información descriptiva de una situación, y se obtiene la posible causa del estado del sistema. Cada clase representa un estado del sistema, el cual se puede asociar a un mal funcionamiento o falla específica. Cada individuo corresponde a una medición obtenida al tener el sistema en funcionamiento .

8.2.1 Descripción del sistema

Se empleó un sistema de tanques acoplados que se encuentra en el laboratorio de electrónica de la Universidad de los Andes. La planta consta de una bomba de agua y dos tanques, uno lineal y otro no lineal, comunicados por medio de un tubo (ver fig. 19). La bomba permite controlar el caudal de agua que ingresa al tanque lineal, mediante el cual se controla el nivel de agua sobre el segundo tanque (tanque no-lineal). Se dispone además de dos válvulas ON-OFF que permiten simular fugas inesperadas en el Sistema de Tanques Acoplados [30].



Figure 19: Disposición de los tanques de prueba

8.2.2 Datos Utilizados

Los datos utilizados fueron obtenidos y utilizados por Fandiño [30] en su trabajo con el método LAMDA, los cuales tienen 3219 individuos, obtenidos mediante mediciones al sistema realizadas cada dos segundos. Los atributos empleados son los niveles de agua de los tanques y sus derivadas, tal como se aprecia en la fig. 20, donde se presenta el nivel de agua del tanque lineal y la derivada de dicho nivel. En este escenario de funcionamiento se pueden identificar 5 clases iniciales que corresponden a los siguientes estados de la planta: Estado de transición inicial, Estado Estable en el primer nivel de referencia (20 cm), Estado de transición entre los dos niveles, Estado Estable en el segundo nivel de referencia (25 cm) y Estado de falla.

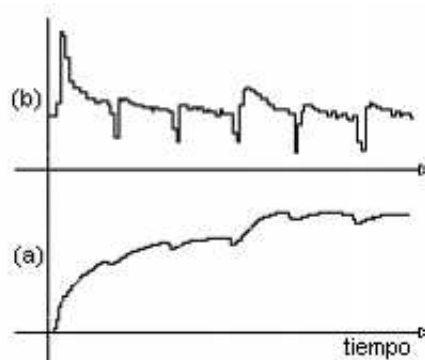


Figure 20: Datos Utilizados para la detección de fallas en Tanques (a) Nivel del tanque (b) Derivada del nivel del tanque 1

8.2.3 Resultados obtenidos

En la fig. 21 se muestran los estados identificados con los métodos determinísticos comparados con el nivel del agua del tanque lineal. En la gráfica se puede ver que los algoritmos reconocen un estado de falla y varios estados de niveles de agua y de cambios de nivel.

En la fig. 22 se muestran los estados identificados con los métodos difusos comparados con el nivel del agua del tanque lineal. En la gráfica se puede ver que, al igual que los métodos determinísticos, los algoritmos reconocen un estado de falla y varios estados de niveles de agua y de cambios de nivel.

Con estos resultados se puede ver el buen funcionamiento de los algoritmos metaheurísticos implementados, pues todos logran reconocer apropiadamente los estados de la planta.

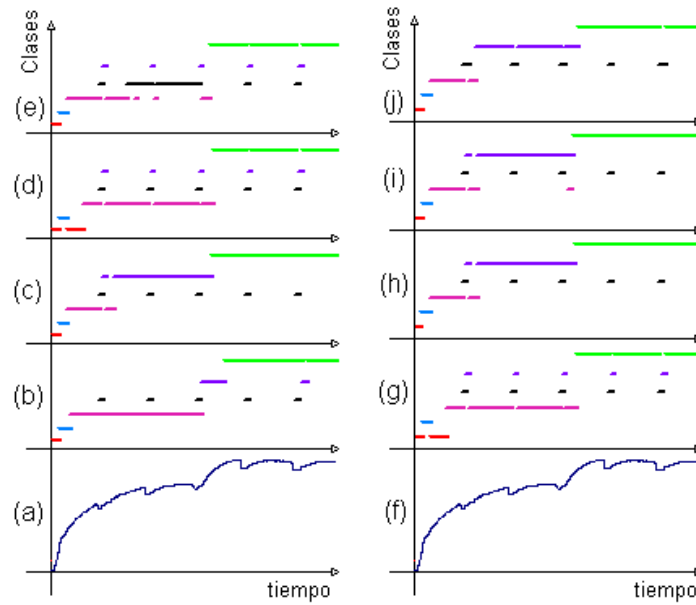


Figure 21: Resultados de métodos determinísticos en detección de fallas. (a)Datos, (b)TS-U, (c)SA-U, (d)GA-U, (e)ACO-U, (f) Datos, (g)KM, (h)TS-Cent, (i)SA-Cent, (j)GA-Cent

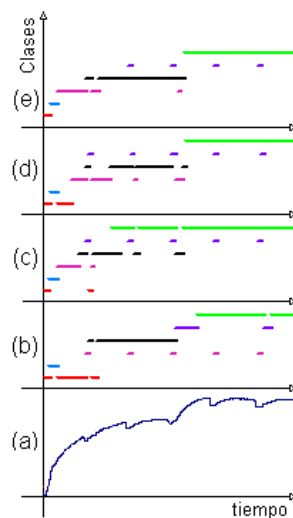


Figure 22: Resultados de métodos difusos en detección de fallas. (a)Datos, (b)CM, (c)TS-Fuzzy, (d)SA-Fuzzy, (e)GA-Fuzzy

8.3 Aplicación a Clasificación de Empresas del sector eléctrico

“La Ley colombiana 143 de 1999, reafirma la posibilidad de participación privada en los negocios eléctricos, creando un mercado mayorista de la electricidad y definiendo los procedimientos y mecanismos para regular las actividades de la industria. En ella la industria eléctrica se dividió en cuatro actividades: generación, transmisión, distribución y comercialización. Los segmentos de generación y comercialización se definieron como competitivos, o potencialmente competitivos, y las actividades de transmisión y distribución son tratadas como monopolios sujetos de regulación.” [40]

Esta regulación la realiza la Comisión de Regulación de Energía y Gas - *CREG*, la cual remunera la inversión y los gastos de administración, operación y mantenimiento de forma separada.

Los segmentos de distribución y transmisión están conformados por un buen número de empresas que difieren en ciertas características propias del negocio y de su entorno. Esto hace que la aplicación de un método para la remuneración de ingresos pueda requerir variaciones en función de dichas características.

En esta aplicación se busca conformar grupos o clusters de empresas de transmisión con características similares que las hagan comparables a la hora de adoptar un esquema de regulación de precios para evitar, de cierta manera, distorsiones en el sistema de incentivos [40][8].

La remuneración de las empresas de transmisión corresponde a un ingreso regulado. Dicho ingreso se calcula valorando el servicio a una tarifa estampilla y descontándole los costos eficientes de inversión y de AOM.

Para la remuneración de costos eficientes de AOM se ha determinado que estos corresponden a un porcentaje de la inversión.

En primer lugar se utiliza el algoritmo de clasificación LAMDA (*Learning algorithm for multivariate data analysis*), el cual da un buen punto de partida al no requerir un conocimiento a priori del número de clases deseadas. Posteriormente se emplean los métodos metaheurísticos de clasificación.

8.3.1 Learning Algorithm for Multivariate Data Analysis - LAMDA

LAMDA es un método de clasificación difusa propuesta en los 80's por Joseph Aguilar con la colaboración de otros autores[2]. Se basa en el análisis del grado de adecuación de cada individuo

a una clase; concepto relacionado con el grado de pertenencia a un conjunto difuso.

Cada individuo a clasificar se analiza de forma independiente: primero se debe determinar la contribución de cada uno de los atributos del individuo al grado de pertenencia marginal MAD (Marginal Adequacy Degree). Para cada una de las clases existirán tantos MADs como número de atributos. Para el cálculo del MAD se puede usar la función binomial (18), la binomial modificada (19) o la normal (20).

$$MADx_{jk} = \rho_{jk} \cdot (1 - \rho_{jk})^{1-x_j} \quad (18)$$

$$MADx_{jk} = \rho_{jk}^{1-d_{jk}} \cdot (1 - \rho_{jk})^{d_{jk}} \quad (19)$$

donde $d_{jk} = |x_j - c_{jk}|$.

$$MADx_{jk} = \frac{1}{\sigma_{jk}} \cdot e^{-\frac{1}{2} \frac{(x_j - \mu_{jk})^2}{\sigma_{jk}^2}} \quad (20)$$

donde en cada caso x_j corresponde al valor presente en el descriptor j para el individuo x , ρ_{jk} corresponde al valor medio de los x_j pertenecientes a la clase k y c_{jk} a la mediana. En el caso de la función normal μ_{kj} es la media de los x_j pertenecientes a k y σ_{kj} a la desviación estándar.

Combinando los valores de los MADs mediante operadores "Difusos Mixtos" (21) se determina la adecuación global del individuo a cada una de las clases, GAD. (Global Adequacy Degree).

$$GADx_k = \alpha T(MADx_{1k}, MADx_{2k}, \dots, MADx_{jk}) + (1 - \alpha) S(MADx_{1k}, MADx_{2k}, \dots, MADx_{jk}) \quad (21)$$

donde α corresponde al parámetro de exigencia y está definido entre 0 y 1 (0 representa la menor exigencia y la máxima exigencia se obtiene con 1). T corresponde a cualquier T-Norma y S a cualquier S-Norma.

El aprendizaje en este método consiste en encontrar los parámetros correspondientes a la función MAD elegida (ρ_{jk} , c_{jk} , μ_{jk} o σ_{jk}).

Hay dos características que diferencian a LAMDA de los demás métodos:

1. No se debe definir a priori el número de clases que se quiere tener.

2. En LAMDA, además de las clases propias del problema, se incluye una clase de no información NIC (Non Informative Class), la cual determina el umbral mínimo de GAD que debe tener un individuo para pertenecer a una clase.

8.3.2 Descriptores utilizados

Para la clasificación de las empresas se utilizó información de las Supercifras del año 2002 de la Superintendencia de Servicios Públicos Domiciliarios [9] y el centro de documentación XM[24]. Las variables empleadas fueron:

1. Longitud total de las líneas (km)
2. Capacidad de transformación (MVA)
3. Nivel de integración: Una característica importante de las empresas del mercado eléctrico colombiano es su grado de integración. Este aspecto puede incidir sobre los análisis de eficiencia que se realicen con base en información contable de las empresas (variable cualitativa).
4. Valor de los activos: Valor de los activos de la empresa como indicador de su tamaño, lo cual es un factor importante a la hora de generar grupos de empresas.

8.3.3 Resultados

El primer paso que se realizó para lograr la clasificación de las empresas fue la normalización de los datos, pues los descriptores elegidos se encuentran en rangos muy diferentes entre si. Si no se hace la normalización, al hacer clasificación mediante la minimización de las distancias, se le estaría dando un mayor peso a unas variables que a otras.

Posteriormente, se realizó una clasificación con el método LAMDA, el cual produjo 4 clases. Con base en esta información se llevó a cabo la clasificación con GA-Fuzzy y SA-Fuzzy, los resultados obtenidos son muy similares y en la tabla 3 se presentan los de GA-Fuzzy. Se escogieron métodos difusos, ya que los niveles de pertenencia de las empresas a las clases brindan información en cuanto a similitudes que tienen las empresas entre si.

En cuanto a los resultados obtenidos es de notar que existen dos clases con una sola empresa: la clase 0 y la 3. Esto indica que hay empresas con características bastante diferentes. De igual forma, es de notar que los niveles de pertenencia de las empresas a las clases son considerables

Empresas Transmisión	Clase	Nivel de pertenencia			
		Clase 0	Clase 1	Clase 2	Clase 3
EPSA	0	1.000	0.000	0.000	0.000
DISTASA	1	0.031	0.949	0.007	0.012
ESA	1	0.025	0.964	0.004	0.007
TRANSELCA	1	0.005	0.992	0.001	0.002
EEB	2	0.012	0.004	0.934	0.050
EPM	2	0.016	0.004	0.856	0.124
ISA	3	0.000	0.000	0.000	1.000

Tabla 3: Resultados de clasificación de empresas de transmisión

aunque hay empresas, como EPM, que tiene cierta similitud con otra clase distinta a la que pertenece.

CAPÍTULO IX

Conclusiones

Se desarrollaron métodos de clasificación utilizando técnicas de optimización metaheurísticas, cuyo funcionamiento se comprobó y se evaluó con diferentes aplicaciones y comparando los resultados con dos algoritmos, KM y CM.

Al evaluar los métodos, se vio la semejanza en el comportamiento de los métodos determinísticos (TS-U, SA-U, GA-U, ACO-U, TS-Cent, SA-Cent y GA-Cent) con KM, y es de resaltar que todos estos algoritmos generan errores y funciones objetivo menores o iguales que las obtenidas con KM. En cuanto a los métodos difusos (TS-Fuzzy, SA-Fuzzy y GA-Fuzzy), los errores de entrenamiento y prueba encontrados para los metaheurísticos son mayores que los de CM, aunque la función objetivo encontrada es menor. Por último, se puede ver que el error de prueba de GKM es menor que todos los errores encontrados, debido a la posibilidad que tiene de generar clases con formas diferentes a hipersferas. Una modificación posible a los algoritmos desarrollados es el uso de la función objetivo con distancia de Mahalanobis.

Otro aspecto importante en la evaluación es el tiempo de convergencia de los métodos, donde se pudo ver que, aunque las metaheurísticas hacen una búsqueda más exhaustiva en el espacio de decisión, los tiempos de convergencia son mayores que los de KM y CM, pero las diferencias no son demasiado representativas. Es de notar, sin embargo, que los métodos que usan una solución de tipo centroide son más lentos que los que emplean la solución matriz de pertenencia; esto debido a que en los últimos se utiliza un método más eficiente de actualización de la función objetivo.

Otro punto que se evaluó fue el cambio del tiempo de convergencia con el incremento del número de individuos y se observó que existe un aumento, especialmente para los algoritmos

difusos. Para la prueba realizada, los incrementos tienen una tendencia lineal con pendiente aproximada de 0.002 para los métodos con solución U; 0.003 para los métodos determinísticos con solución Centros; y 0.005 para los difusos.

De igual forma, es de resaltar que el tiempo de convergencia está directamente relacionado con la solución inicial. Por esta razón, se propuso un método de inicialización de centroides mediante la discretización del espacio de decisión que reflejó mejores resultados que una inicialización completamente aleatoria.

Un aspecto importante de los métodos con solución matriz de pertenencia es que permite tener un número variable de clases, es decir, se puede fijar un número máximo de clases y un número mínimo de individuos por clase, obteniendo al final de la clasificación particiones con un número “óptimo” de clusters.

De otro lado, se pudo ver que los algoritmos que tienen población inicial de soluciones, y no una única solución (GA-U, ACO-U, GA-Cent y GA-Fuzzy), pueden tener ventajas sobre los otros algoritmos al evaluar de manera “paralela” diferentes regiones del espacio de decisión.

En la base de datos iris, cuyas clases no son completamente separables, se puede apreciar que algunos de los métodos metaheurísticos tienen ventajas sobre KM y CM, al hacer una exploración más amplia del espacio de solución. De igual forma, en el diagnóstico de fallas, los algoritmos desarrollados presentan buenos resultados, puesto que logran reconocer apropiadamente los estados de la planta. Y se puede ver que, en general, los métodos metaheurísticos de clasificación tienen un buen desempeño en diversas aplicaciones.

Por otro lado, este trabajo realiza un aporte en el desarrollo de un marco regulatorio acorde con la realidad de las empresas, al agrupar las empresas de transmisión de energía eléctrica mediante algoritmos con buena fundamentación matemática. Adicionalmente, esta aplicación se encuentra enmarcada en el proyecto de investigación de la CREG, Colciencias y Uniandes para el *Cálculo de costos eficientes de Administración, Operación y Mantenimiento* [8], en el cual se emplearán los clusters obtenidos para hacer comparaciones de eficiencia.

Uno de los aspectos más importantes a la hora de realizar la clasificación es la cantidad y calidad de la información con la que ésta se realiza. Aunque los algoritmos no tienen limitaciones

en cuanto al número de descriptores, en el caso en el que existan restricciones para conseguir información, es pertinente realizar un análisis sobre la significancia de los descriptores, tanto antes como después del proceso de clasificación.

Por último, el Toolbox desarrollado permite la clasificación de cualquier base de datos a través de los diversos métodos presentados. De otra parte, su programación orientada a objetos permite su extensión, usando las clases con las cuales se representan las soluciones, y las que implementan varios tipos de utilidades (para vectores, números, matrices, entre otros).

APÉNDICE A

Análisis de descriptores

A.1 A priori

Antes de iniciar el proceso de clasificación se puede efectuar un análisis de la correlación de los descriptores con el objeto de determinar cuales de ellos son redundantes. De esta manera, se puede identificar el número mínimo de atributos que definiría el agrupamiento de los individuos según sus características. Esto se puede llevar a cabo mediante análisis de componentes principales [19], analizando el peso de cada una de las variables en las primeras componentes. También se puede calcular una matriz de correlaciones y tratar de construir variables a partir de aquellos descriptores que se encuentren altamente correlacionados.

Esto es importante por dos razones:

1. Entrega información sobre las características que realmente influyen en la formación de grupos.
2. En aplicaciones con restricciones de información, como la clasificación de empresas eléctricas para aspectos regulatorios, disminuye los requerimientos de información, esto es, en general, un aspecto crítico para el organismo regulador debido a la enorme asimetría de información existente.

A.2 A posteriori

Una vez se realiza la clasificación se puede determinar qué tan significativo es cada uno de los atributos, calculando la varianza de cada atributo en cada una de las clases, permitiendo de esta manera determinar cuales atributos influyen realmente en la clasificación.

Referencias

- [1] E. Acuna, editor. *Análisis estadístico de datos usando MINITAB para Windows*. Ed. Sanki, 2000. On-line available as <http://math.uprm.edu/~edgar/>.
- [2] J. Aguilar and R. Lopez. The process of classification and learning the meaning of linguistic descriptors of concepts. *Aproximate reasoning in Decision Analysis*, pages 165–175, 1982.
- [3] K.S. Al-Sultan and C.A. Fedjki. A tabu search-based algorithm for the fuzzy clustering problem. *Pattern Recognition*, 30(12):2023–2030, 1997.
- [4] R. Babuska. *Fuzzy Modeling for Control*. Kluwer Academic Publishers, Mass., EEUU, 1998.
- [5] C. Bishop. *Neural networks for pattern recognition*. Oxford University Press, Oxford, UK, 1995.
- [6] G. Canavos. *Probabilidad y Estadística*. Mc. Graw Hill, Méjico, 1998.
- [7] O. Cordon, F. Herrera, F. Hoffmann, and L. Magdalena. *Genetic Fuzzy Systems. Evolutionary Tuning and Learning Knowledge Bases*. World Scientific, Singapore, 2001.
- [8] Grupo de Investigación en Potencia y Energía, Grupo en Informática y Automatización para la Producción, and Centro para la Optimización y Probabilidad Aplicada. *Metodologías para la remuneración de costos eficientes de aom de empresas de transmisión y distribución eléctrica*. Technical report, Universidad de los Andes, Bogotá, Colombia, 2005.
- [9] Superintendencia de Servicios Públicos Domiciliarios. *Supercifras en kilovatios hora*, 2002.
- [10] M. Delgado, A. Gómez, and H. Martínez. A tabu search approach to the fuzzy clustering problem. In *Sixth IEEE International Conference on Fuzzy Systems*, Barcelona, España, 1997.
- [11] M. Dorigo, V. Maniezzo, and A. Colorni. The ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man and Cybernetics*, 26(1):1–13, 1996.
- [12] K.A. Dowsland and B.A. Díaz. Heuristic design and fundamentals of the simulated annealing. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*, (1):93–102, 2003.

- [13] Kathryn A. Dowsland and B. Adenso Díaz. Heuristic design and fundamentals of the simulated annealing. Technical report, Research Group School of Computer Science and Information Technology, University of Nottingham and Escuela Politécnica Superior de Ingeniería de Gijón, Universidad de Oviedo, Nottingham, UK and Gijón, Spain.
- [14] F. Glover. Tabu search - part i. *ORSA Journal on Computing*, 1:190–206, 1989.
- [15] F. Glover. Tabu search: A tutorial. *Interfaces*, 20(4):74–94, 1990.
- [16] F. Glover and M. Laguna. *Tabu Search*. Kluwer Academic Publishers, MA, USA, sixth edition, 2002.
- [17] R.L. Haupt and S.E. Haupt. *Practical Genetic Algorithms*. John Wiley & Sons, NJ, USA, second edition, 2004.
- [18] T. Hofmann and J. Buhmann. Pairwise data clustering by deterministic annealing. *IEEE Transactions on Pattern Analysis and Machine Learning*, 19(1), 1997.
- [19] R. Johnson. *Applied Multivariate Analysis*. Prentice Hall, NJ, USA, fifth edition, 2002.
- [20] R. Johnson and D. Wichern. *Fuzzy classifier Design*. Prentice Hall, New Jersey, USA, 2002.
- [21] L. Kuncheva. *Fuzzy classifier Design*. Physica-Verlag, UK, 2000.
- [22] N. Labroche, N. Monmarché, and G. Venturini. A new clustering algorithm based on the chemical recognition system of ants. Technical report, Laboratoire d'Informatique de l'Université de Tours, Francia.
- [23] A.M. Law and W.D. Kelton. *Simulation, Modeling and Analysis*. Mc. Graw Hill, USA, 2000.
- [24] X M los expertos en mercados. Centro de documentación. On-line available as <http://www.xm.com.co/>.
- [25] A. Marcucci and A. Gauthier. Desarrollo de un toolbox para clasificación de sistemas. Technical report, Centro de Investigación de la Facultad de Ingeniería, Universidad de los Andes, Bogotá, Colombia, 2004.
- [26] A. Marcucci, A. Gauthier, M. Jaramillo, and C. Isaza. Aplicación de algunas técnicas de clasificación en dos casos de diagnóstico. In *VI Congreso de la Asociación Colombiana de Automática*, Ibagué, Colombia, 2004.
- [27] A. Marcucci, M. Jaramillo, and A. Gauthier. Análisis comparativo de métodos de clasificación. Electronical engineering thesis, Universidad de los Andes, Bogotá, Colombia, 2004.
- [28] D. Michie, D.J. Spiegelhalter, and C. Taylor, editors. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, 1994. On-line available as <http://www.amsta.leeds.ac.uk/~charles/statlog/>.

- [29] M. Mitchell. *An introduction to genetic algorithms*. MIT Press, Mass, London, 1999.
- [30] J. Fandi no. Método lamda para clasificación de estados funcionales y plataforma de clasificación de fallas en procesos industriales. Electronical engineering thesis, Universidad de los Andes, Bogotá, Colombia, 2004.
- [31] University of California. Uci machine learning repository. On-line available as <http://www.ics.uci.edu/~mllearn/MLSummary.html>.
- [32] J. Otero and D. Guerrerro. Clasificación automática mediante colonia de hormigas. In *VIII Congreso Internacional de Matemática y Ciencias de la computación, Sociedad Cubana de Matemática y Computación*, Cuba, 2003.
- [33] P.S. Shelokar, V.K. Jayaraman, and B.D. Kulkarni. An ant colony approach for clustering. *Analytica Chimica*, 2004.
- [34] C. Sun, E. Mizutani, and J. Jang. *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. Prentice Hall, NJ, USA, first edition.
- [35] C.S. Sung and H.W. Jin. A tabu-search-based heuristic for clustering. *Pattern Recognition*, (33):849–858, 2000.
- [36] J.F. Torres. Descomposición robusta de horizontes temporales en planificación de la producción. Unpublished.
- [37] J. Trejos, E. Piza, and A. Murillo. A tabu search algorithm for partitioning. Technical report, Universidad de Costa Rica, San José, Costa Rica, 2004.
- [38] Cheng-Fa Tsai, Chun-Wei Tsai, Han-Chang Wu, and Tzer Yang. Acodf: a novel data clustering approach for data mining in large databases. *The Journal of Systems and Software*, 2004.
- [39] L.Y. Tseng and S.B. Yang. A genetic approach to the automatic clustering problem. *Pattern Recognition*, (34):415–424, 2001.
- [40] S. Urbiztondo, J. Rojas, and C. Chahín. Estudio sectorial sobre el sector eléctrico de colombia: “al rescate de la reforma eléctrica de 1994”, informe preliminar. Technical report, Fedesarrollo, Bogotá, Colombia, 2004.