Revista de Matemática: Teoría y Aplicaciones 1999 **6**(1) : 51–66
cimpa – ucr – ccss issn: 1409-2433

NUEVAS TÉCNICAS DE PARTICIONAMIENTO EN CLASIFICACIÓN AUTOMÁTICA

Eduardo Piza*– Alex Murillo†– Javier Trejos‡

Recibido: 19 Noviembre 1998

Resumen

En este artículo se exponen algunas ténicas novedosas para la búsqueda de óptimos globales en el problema de la Clasificación Automática por medio de particiones, con las cuales se mejoran sensiblemente los resultados obtenidos con los métodos tradicionales. Los métodos aquí desarrollados son bien conocidos en el campo de la Optimización Combinatoria: i) *el sobrecalentamiento simulado*; ii) *la búsqueda tabú*; iii) *los algoritmos genéticos*. Se utilizan estos tres enfoques aplicados al problema del particionamiento de objetos en Clasificación Automática, siguiendo un esquema de búsqueda análogo al planteado en el tradicional algoritmo de transferencias de Régnier.

Palabras-clave: Clasificación, optimización estocástica, optimización combinatoria, heurística.

Abstract

In this article there are exposed some new techniques for the search of global optima in the partitioning problem in Cluster Analysis. With these techniques the results are sensibly improved with respect to the traditional methods. The methods developed here are well known in Combinatorial Optimization: i) *simulated annealing*; ii) *tabu search*; iii) *genetic algorithms*. We use these three approaches in the partitioning problem for clustering, following a search scheme similar to that of Regnier's algorithm of transfers.

Keywords: clustering, stochastic optimization, combinatorial optimization, heuristics.

AMS Subject Classification: 92G30, 62H30, 90C15.

* CIMPA, Escuela de Matemática, Universidad de Costa Rica, 2060 San José, Costa Rica; Tel.: +(506) 207 4400, Fax: +(506) 207 4397; E-Mail: epiza@cariari.ucr.ac.cr

† Misma dirección; E-Mail: murillof@cariari.ucr.ac.cr

‡ Misma dirección; Tel.: +(506) 207 5574; E-Mail: jtrejos@cariari.ucr.ac.cr

1 Introducción

La Clasificación Automática puede definirse como el campo de la matemática aplicada que pretende resolver, mediante ideas, algoritmos y métodos el problema general de la clasificación: *dada una colección de objetos, deseamos clasificarlos en clases o grupos bien diferenciados, de acuerdo con las disimilitudes entre los mismos, de forma tal que las clases sean homogéneas internamente*. Podemos distinguir entre dos grandes familias de métodos:

- **Los métodos jerárquicos**, en los cuales trabajamos con jerarquías indexadas, árboles taxonómicos, ultramétricas, pirámides, disimilitudes, agregaciones, etc. La teoría sobre la clasificación jerárquica fue desarrollada en su mayoría en las décadas de los años 60 y 70, principalmente por Ward, Johnson, Jardine, Sibson, Benzécri, Diday, Lance, Williams y Jambu, entre otros [Piz87, Cel89].

- **Los métodos de particionamiento**, en los cuales se busca una única partición de los objetos en estudio en k clases disjuntas. La teoría tradicional de los métodos de particionamiento son fundamentalmente: "k-means" de Forgy, "Nubes Dinámicas" de Diday, "algoritmo de transferencias" de Régnier o "k-means de McQueen, "Particiones Principales", entre otros [Cel89].

En este artículo estamos interesados en la Clasificación Automática mediante particionamiento y presentamos un nuevo enfoque al problema, mediante el empleo de las más modernas técnicas de optimización combinatoria.

Los métodos tradicionales de Clasificación Automática, para encontrar una partición de un conjunto de objetos en un número prefijado de clases, obtienen tan solo soluciones parciales al problema, localizando apenas algunos *óptimos locales* [Did80, Did82, Ler81].

De ahí surge nuestro interés por aplicar algunas de las modernas técnicas de la Optimización Combinatoria para la búsqueda de *óptimos globales*, entre ellas el *sobrecalentamiento simulado*, la *búsqueda tabú* y los *algoritmos genéticos* [Tre98]. En particular hemos obtenido excelentes resultados con las técnicas de sobrecalentamiento simulado y la búsqueda tabú, como se describe en el presente artículo.

Los problemas de los que trata la Optimización Combinatoria se pueden formular como sigue: *dado un conjunto finito de configuraciones y una función de costo que evala cada configuración, se trata de encontrar "la mejor" configuración del conjunto*. A pesar de que esta manera de presentar la disciplina hace que ésta pueda parecer casi trivial, en la práctica existen serias dificultades de índole matemática y computacional que han hecho que la Optimización Combinatoria haya sido objeto de intensos estudios en los últimos años, obteniéndose un apreciable desarrollo teórico con la introducción de las llamadas técnicas heurísticas y estocásticas.

La sección 2 presenta los detalles del problema de particionamiento y en la sección 3 se muestran algunos resultados teóricos que justifican el uso de tcnicas de optimización combinatoria. Los nuevos mtodos propuestos son presentados en la sección 4, junto con una breve descripción de las tres tcnicas de optimización combinatoria usados: sobreca-

lentamiento simulado, bsqueda tab y algoritmos genticos. Los resultados obtenidos sobre algunas tablas de datos se presentan en la

sección 5, terminándose el artículo con una sección de conclusiones.

2 Clasificación Automática por Particionamiento

Sea $\Omega = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ el conjunto finito de n objetos que deseamos clasificar y sea $k < n$ el número de clases en los cuales deseamos clasificar a los objetos. Una partición $P = (C_1, \ldots, C_k)$ de Ω en k clases C_1, ..., C_k, está caracterizada por las siguientes condiciones:

1. $\Omega = \bigcup_{i=1}^{k} C_i$.
2. $C_i \cap C_j = \emptyset$, para todo $i \neq j$.

En nuestro enfoque se permitir eventualmente que algunas de las clases C_i sea vacía, de manera que en realidad las particiones $P = (C_1, \ldots, C_k)$ que se consideran son particiones de Ω en k *o menos* clases. Sin embargo, se ver (teorema 3) que las particiones óptimas —de acuerdo al criterio de la inercia aquí considerado— contienen exactamente k clases no vacías.

Sea $\mathcal{P}_k$ el conjunto de todas las particiones $P = (C_1, \ldots, C_k)$ de Ω en k o menos clases. Se quieren encontrar "buenas particiones", esto es, aquellas particiones que reflejen las relaciones de similitud existentes entre los objetos $\mathbf{x}_i \in \Omega$. Cada objeto $\mathbf{x}_i \in \Omega$ estará caracterizado por p distintos atributos o variables medidos en una escala numérica, de donde cada objeto $\mathbf{x}_i$ será visto como un vector del espacio euclídeo $\mathbb{R}^p$. En este espacio de representación se cuenta con una métrica euclídea M (matriz simétrica y definida positiva), que sirve para definir el producto interno $\langle \mathbf{x}_i \,|\, \mathbf{x}_j \rangle_M = \mathbf{x}_i^t M \mathbf{x}_j$ entre objetos y la norma $\|\mathbf{x}\|_M^2 = \mathbf{x}^t M \mathbf{x}$. En la programación de los algoritmos se ha supuesto, sin pérdida de generalidad para efectos de convergencia, que $M = \mathrm{Id}$ (métrica euclídea clásica). En efecto, en el caso general M se descompone como $U^t U$, y la transformación $\mathbf{z}_i = U\mathbf{x}_i$ nos lleva a la métrica euclídea clásica, con los nuevos datos $\mathbf{z}_i$.

Asociado con cada objeto $\mathbf{x}_i \in \Omega$ tendremos el peso de $\mathbf{x}_i$, denotado por ω_i, que refleja la importancia relativa del objeto $\mathbf{x}_i$ en el estudio. Los pesos ω_i son todos positivos y su suma es la unidad: $\sum_{i=1}^{n} \omega_i = 1$.

La calidad de una partición $P = (C_1, \ldots, C_k)$ se mide a través de la *inercia inter-clases* $B(P)$, índice que refleja la intensidad de la separación entre los centros de gravedad de las diversas clases C_j:

$$B(P) := \sum_{j=1}^{k} \omega(C_j) \cdot \|\mathbf{g}(\Omega) - \mathbf{g}(C_i)\|^2,$$

donde $\omega(C_j)$ es el peso relativo de la clase C_j mientras que $\mathbf{g}(\Omega)$ y $\mathbf{g}(C_j)$ son los vectores centros de gravedad de Ω y C_j respectivamente, calculados mediante las fórmulas

siguientes:

$$w(C_j) = \sum_{\mathbf{x}_i \in C_j} w_i, \qquad \mathbf{g}(\Omega) = \sum_{i=1}^{n} \omega_i \, \mathbf{x}_i, \qquad \mathbf{g}(C_j) = \frac{1}{\omega(C_j)} \sum_{\mathbf{x}_i \in C_j} \omega_i \, \mathbf{x}_i.$$

El problema de la Clasificación Automática por medio de particionamiento, consiste entonces en hallar la partición $P \in \mathcal{P}_k$ que maximiza la inercia inter-clases $B(P)$; esto es, maximizar la intensidad de la separación entre los centros de gravedad. La complejidad computacional de este problema es del tipo *NP-hard*, pues se trata de una generalización del conocido problema *NPP*, el cual originó la terminología de este grado de complejidad para problemas análogos [Laa88]. Para tener una idea del tamaño combinatorio de este problema, si denotamos por $S(n,k)$ y B_n el número de particiones de Ω en k clases no vacías y el número total de particiones de Ω respectivamente[1], entonces por ejemplo $S(60,2) \approx 0.58 \times 10^{18}$, $S(60,5) \approx 0.72 \times 10^{40}$, $S(100,5) \approx 0.66 \times 10^{68}$, mientras que $B_{10} = 115975$, $B_{15} \approx 0.14 \times 10^{10}$ y $B_{40} \approx 0.16 \times 10^{36}$. En un problema de Clasificación Automática de tamaño "regular", en el cual Ω tenga 100 objetos y el número de clases sea $k = 5$, si existiera sobre la Tierra un computador tan veloz que fuese capaz de calcular $B(P)$ para cada una de las particiones P de Ω en un tiempo de 10^{-10} segundos en busca de un máximo global, ¡a este veloz computador le tomaría algo más de 2×10^{48} siglos en completar el análisis de todas las particiones del problema!

El otro criterio de calidad de una partición $P = (C_1, \ldots, C_k) \in \mathcal{P}_k$ es el criterio de la *inercia intra-clases* $W(P)$, definido mediante

$$\begin{aligned} W(P) &= \sum_{j=1}^{k} I(C_j), \\ \text{con} \quad I(C_j) &= \sum_{\mathbf{x}_i \in C_j} \omega_i \, \|\mathbf{x}_i - \mathbf{g}(C_j)\|_M^2. \end{aligned}$$

El término $I(C_j)$ es llamado la *inercia de la clase* C_j. Este criterio brinda información acerca de la homogeneidad interna de las clases C_j de la partición considerada. Entre menor sea el valor de $W(P)$, mejor será la partición P, pues la "dispersión" de los objetos dentro de cada una de las clases es más pequeña: se trata de clases homogéneas.

En los algoritmos que desarrollamos utilizamos algunas veces el criterio de la inercia inter-clases $B(P)$ y en otras el criterio de la inercia intra-clases $W(P)$, motivados por el resultado del teorema 1 y el corolario 1.

[1] Los números B_n son conocidos en la literatura como los números de Bell, mientras que los números $S(n,k)$ son conocidos como los números de Stirling de segunda especie. Una fórmula para estos números es la siguiente:

$$S(n,k) = \frac{1}{k!} \sum_{j=0}^{k} (-1)^{k-j} \binom{k}{j} j^n \qquad , \qquad B_n = e^{-1} \sum_{j=0}^{\infty} \frac{(j+1)^n}{j!}.$$

3 Algunos resultados teóricos

Teorema 1 (Huygens) *Para cada partición $P \in \mathcal{P}_k$, se tiene que la suma $B(P)+W(P)$ es siempre la constante I_Ω, llamada* inercia total de Ω*, cuya fórmula es*

$$I_\Omega := \sum_{i=1}^{n} \omega_i \, \|\mathbf{x}_i - \mathbf{g}(\Omega)\|_M^2.$$

Demostración: Puede consultarse en las referencias [Pag76, Did82, Cel89]. ■

Corolario 1 *Los siguientes dos problemas de optimización combinatoria son equivalentes:*

$$(1) \left\{ \begin{array}{c} \text{Maximizar } B(P) \\ \text{sujeto a } P \in \mathcal{P}_k \end{array} \right. \qquad (2) \left\{ \begin{array}{c} \text{Minimizar } W(P) \\ \text{sujeto a } P \in \mathcal{P}_k \end{array} \right.$$

Demostración: Consecuencia inmediata del teorema anterior. ■

Para una implementación eficiente de los algoritmos de sobrecalentamiento simulado y búsqueda tabú, es de fundamental importancia el poder calcular en forma rápida la variación en la inercia intra-clases ΔW, luego de transferir un objeto de una clase a otra, aprovechando los cálculos realizados para la partición previa. Para tal fin, obtuvimos el siguiente resultado.

Teorema 2 *Cuando se genera una nueva partición $\tilde{P}$ a partir de P, formada al transferir el objeto $\mathbf{x}_i$ de la clase no unitaria C_j a la clase C_ℓ, se puede calcular recursivamente las nuevas inercias y centros de gravedad a partir de los previamente calculados, como sigue:*

- *Cambio total de la inercia:*

$$\begin{aligned} \Delta W &= W(\tilde{P}) - W(P) \\ &= \frac{\omega(C_\ell)\,\omega_i}{\omega(C_\ell)+\omega_i} \, \|\mathbf{g}(C_\ell) - \mathbf{x}_i\|_M^2 - \frac{\omega(C_j)\,\omega_i}{\omega(C_j)-\omega_i} \, \|\mathbf{g}(C_j) - \mathbf{x}_i\|_M^2 \end{aligned}$$

- *Inercia de las nuevas clases:*

$$\begin{aligned} I(C_\ell \cup \{\mathbf{x}_i\}) &= I(C_\ell) + \frac{\omega(C_\ell)\,\omega_i}{\omega(C_\ell)+\omega_i} \, \|\mathbf{g}(C_\ell) - \mathbf{x}_i\|_M^2 \\ I(C_j \setminus \{\mathbf{x}_i\}) &= I(C_j) - \frac{\omega(C_j)\,\omega_i}{\omega(C_j)-\omega_i} \, \|\mathbf{g}(C_j) - \mathbf{x}_i\|_M^2 \end{aligned}$$

- *Nuevos centros de gravedad:*

$$\begin{aligned} \mathbf{g}(C_\ell \cup \{\mathbf{x}_i\}) &= \frac{1}{\omega(C_\ell)+\omega_i} \, (\omega(C_\ell)\,\mathbf{g}(C_\ell) + \omega_i\,\mathbf{x}_i) \\ \mathbf{g}(C_j \setminus \{\mathbf{x}_i\}) &= \frac{1}{\omega(C_j)-\omega_i} \, (\omega(C_j)\,\mathbf{g}(C_j) - \omega_i\,\mathbf{x}_i) \end{aligned}$$

Demostración: Primeramente se calculan los nuevos centros de gravedad:

$$\begin{aligned}
\mathbf{g}(C_\ell \cup \{\mathbf{x}_i\}) &= \frac{1}{\omega(C_\ell \cup \{\mathbf{x}_i\})} \left(\sum_{\mathbf{x}_s \in C_\ell} \omega_s \, \mathbf{x}_s + \omega_i \, \mathbf{x}_i \right) \\
&= \frac{\omega(C_\ell)}{\omega(C_\ell) + \omega_i} \, \mathbf{g}(C_\ell) + \frac{\omega_i}{\omega(C_\ell) + \omega_i} \, \mathbf{x}_i. \\
\mathbf{g}(C_j \setminus \{\mathbf{x}_i\}) &= \frac{1}{\omega(C_j \setminus \{\mathbf{x}_i\})} \sum_{\substack{\mathbf{x}_s \in C_j \\ \mathbf{x}_s \neq \mathbf{x}_i}} \omega_s \, \mathbf{x}_s = \frac{\omega(C_j)}{(\omega(C_j) - \omega_i)} \frac{1}{\omega(C_j)} \Big(\sum_{\mathbf{x}_s \in C_j} \omega_s \, \mathbf{x}_s - \omega_i \, \mathbf{x}_i \Big) \\
&= \frac{\omega(C_j)}{\omega(C_j) - \omega_i} \, \mathbf{g}(C_j) - \frac{\omega_i}{\omega(C_j) - \omega_i} \, \mathbf{x}_i,
\end{aligned}$$

obteniéndose entonces las fórmulas para los nuevos centros de gravedad. Para demostrar las fórmulas de las inercias de las nuevas clases, se utiliza el siguiente resultado bien conocido (ver por ejemplo [Did82]), válido para dos clases disjuntas h y h' de Ω:

$$I(h \cup h') = I(h) + I(h') + \frac{\omega(h) \cdot \omega(h')}{\omega(h) + \omega(h')} \, \|\mathbf{g}(h) - \mathbf{g}(h')\|_M^2.$$

Aplicando este resultado en el cálculo de las nuevas inercias de los grupos, se obtiene inmediatamente,

$$I(C_\ell \cup \{\mathbf{x}_i\}) = I(C_\ell) + \frac{\omega(C_\ell) \cdot \omega_i}{\omega(C_\ell) + \omega_i} \, \|\mathbf{g}(C_\ell) - \mathbf{x}_i\|_M^2,$$

mientras que para el cálculo de la inercia de $I(C_j \setminus \{\mathbf{x}_i\})$ se procede como sigue:

$$\begin{aligned}
I(C_j) &= I(\{\mathbf{x}_i\} \cup (C_j \setminus \{\mathbf{x}_i\})) \\
&= I(C_j \setminus \{\mathbf{x}_i\}) + \frac{\omega(C_j \setminus \{\mathbf{x}_i\}) \cdot \omega_i}{\omega(C_j \setminus \{\mathbf{x}_i\}) + \omega_i} \, \|\mathbf{g}(C_j \setminus \{\mathbf{x}_i\}) - \mathbf{x}_i\|_M^2 \\
&= I(C_j \setminus \{\mathbf{x}_i\}) + \frac{(\omega(C_j) - \omega_i) \cdot \omega_i}{\omega(C_j)} \, \|\mathbf{g}(C_j \setminus \{\mathbf{x}_i\}) - \mathbf{x}_i\|_M^2. \qquad (3)
\end{aligned}$$

La última expresión en norma admite una simplificación:

$$\begin{aligned}
\|\mathbf{g}(C_j \setminus \{\mathbf{x}_i\}) - \mathbf{x}_i\|_M^2 &= \left\| \frac{\omega(C_j)}{\omega(C_j) - \omega_i} \, \mathbf{g}(C_j) - \frac{\omega(C_j \setminus \{\mathbf{x}_i\}) + \omega_i}{\omega(C_j) - \omega_i} \, \mathbf{x}_i \right\|_M^2 \\
&= \left(\frac{\omega(C_j)}{\omega(C_j) - \omega_i} \right)^2 \|\mathbf{g}(C_j) - \mathbf{x}_i\|_M^2.
\end{aligned}$$

Sustituyendo en (3) y simplificando, se llega a la fórmula para la inercia de la clase $C_j \setminus \{\mathbf{x}_i\}$:

$$I(C_j) = I(C_j \setminus \{\mathbf{x}_i\}) + \frac{\omega(C_j) \cdot \omega_i}{\omega(C_j) - \omega_i} \, \|\mathbf{g}(C_j) - \mathbf{x}_i\|_M^2.$$

Finalmente, calculamos el cambio global de la inercia ΔW:

$$\begin{aligned}\Delta W &= I(C_j \setminus \{\mathbf{x}_i\}) - I(C_j) + I(C_\ell \cup \{\mathbf{x}_i\}) - I(C_\ell) \\ &= \frac{\omega(C_\ell)\cdot\omega_i}{\omega(C_\ell)+\omega_i}\,\|\mathbf{g}(C_\ell)-\mathbf{x}_i\|_M^2 - \frac{\omega(C_j)\cdot\omega_i}{\omega(C_j)-\omega_i}\,\|\mathbf{g}(C_j)-\mathbf{x}_i\|_M^2. \quad \blacksquare\end{aligned}$$

En el teorema anterior se ha supuesto que la clase C_j de la cual proviene el objeto $\mathbf{x}_i$ no es unitaria. En el caso en que $C_j = \{\mathbf{x}_i\}$, el transferir el objeto $\mathbf{x}_i$ de la clase C_j a la clase C_ℓ entonces se vacía la clase C_j, modificándose ligeramente las fórmulas anteriores de la siguiente forma, como el lector puede fácilmente comprobar:

- Cambio total de la inercia:

$$\Delta W = W(\tilde{P}) - W(P) = \frac{\omega(C_\ell)\,\omega_i}{\omega(C_\ell)+\omega_i}\,\|\mathbf{g}(C_\ell)-\mathbf{x}_i\|_M^2.$$

- Inercia de las nuevas clases:

$$\begin{aligned}I(C_\ell \cup \{\mathbf{x}_i\}) &= I(C_\ell) + \frac{\omega(C_\ell)\,\omega_i}{\omega(C_\ell)+\omega_i}\,\|\mathbf{g}(C_\ell)-\mathbf{x}_i\|_M^2 \\ I(C_j \setminus \{\mathbf{x}_i\}) &= \text{no se define.}\end{aligned}$$

- Nuevos centros de gravedad:

$$\begin{aligned}\mathbf{g}(C_\ell \cup \{\mathbf{x}_i\}) &= \frac{1}{\omega(C_\ell)+\omega_i}\,(\omega(C_\ell)\,\mathbf{g}(C_\ell) + \omega_i\,\mathbf{x}_i) \\ \mathbf{g}(C_j \setminus \{\mathbf{x}_i\}) &= \text{no se define.}\end{aligned}$$

Aunque durante el desarrollo de los métodos propuestos trabajamos eventualmente con clases vacías (por motivos de orden práctico en la programación de los algoritmos), sin embargo las particiones óptimas contienen exactamente k clases no vacías. En efecto, esto es demostrado en el siguiente teorema.

Teorema 3 *Bajo las hipótesis que $k < n$ y todos los objetos a clasificar son diferentes, la partición óptima $P^\star$ de los problemas de optimización* (1) *y* (2) *del corolario 1 contiene exactamente k clases no vacías, y no menos.*

Demostración: Supóngase que $P^\star$ tiene *menos* de k clases no vacías. Entonces escójase cualquier objeto $\mathbf{x}_i$ de una de las clases de $P^\star$ que no sea unitaria, de forma tal que $\mathbf{x}_i$ no coincida con el centro de gravedad de su clase.[2] Dígase que $\mathbf{x}_i \in C_j$. Se forma una nueva partición $\tilde{P}$ a partir de $P^\star$, transfiriendo el objeto $\mathbf{x}_i$ a una nueva clase unitaria, $\{\mathbf{x}_i\}$. Evidentemente la nueva partición $\tilde{P}$ pertenece a $\mathcal{P}_k$ y se puede calcular su inercia intra-clases usando una fórmula del teorema anterior:

$$W(\tilde{P}) = I(C_j \setminus \{\mathbf{x}_i\}) + I(\{\mathbf{x}_i\}) + \sum_{s\neq j} I(C_s)$$

[2]Esto siempre puede hacerse puesto que $k < n$ y todos los objetos son diferentes, por hipótesis.

$$
\begin{aligned}
&= I(C_j) + \sum_{s \neq j} I(C_s) - \frac{\omega(C_j) \cdot \omega_i}{\omega(C_j) - \omega_i} \, \|\mathbf{g}(C_j) - \mathbf{x}_i\|_M^2 \\
&= W(P^\star) - \frac{\omega(C_j) \cdot \omega_i}{\omega(C_j) - \omega_i} \, \|\mathbf{g}(C_j) - \mathbf{x}_i\|_M^2 < W(P^\star),
\end{aligned}
$$

lo cual contradice el supuesto de que $P^\star$ es óptima. ■

4 Los métodos propuestos

Hemos desarrollado con éxito tres tipos de algoritmos de naturaleza distinta: i) *sobrecalentamiento simulado*; ii) *búsqueda tabú*; iii) *algoritmos genéticos*. A continuación se describen estos métodos.

Los métodos de sobrecalentamiento simulado y búsqueda tabú hacen uso del concepto de vecindario de una partición. Para cada partición $P = (C_1, \ldots, C_k) \in \mathcal{P}_k$ definimos *el vencindario de* P, denotado por $\mathcal{V}(P)$, como el conjunto de todas las particiones que se obtienen a partir de P al transferir un objeto cualquiera $\mathbf{x}_i$ de un grupo a otro cualquiera. Puede observarse que los vecindarios de cualquier partición $P = (C_1, \ldots, C_k) \in \mathcal{P}_k$ tienen todos la misma cardinalidad: $|\mathcal{V}(P)| = n(k-1)$.

Por otra parte, tanto el algoritmo de sobrecalentamiento simulado como el algoritmo genético hacen uso extensivo de generación de números aleatorios. El éxito de estos algoritmos depende en buena parte de un buen generador de números aleatorios. Nosotros utilizamos el generador de tipo sustractivo propuesto por Knuth [Knu81, Pre90], considerado uno de los mejores generadores rápidos existentes.

4.1 Sobrecalentamiento simulado

El método de sobrecalentamiento simulado es una técnica basada en el algoritmo de Metropolis en la Física Estadística. Fue propuesto por Kirkpatrick, Gelat y Vecchi [Kir83] en 1983 y de manera independiente por Černy [Čer85] en 1985. Los detalles sobre este método pueden ser consultados en las obras de Aarts, Korst y Laarhoven [Aar90, Laa88].

En resumen, el algoritmo de sobrecalentamiento simulado, aplicado al particionamiento de Ω en k clases, intenta resolver el problema (2) de optimización, cual es minimizar la inercia intra-clases $W(P)$ sobre todas las particiones $P \in \mathcal{P}_k$. El algoritmo comienza eligiendo una partición $P = (C_1, \ldots, C_k) \in \mathcal{P}_k$ al azar. A partir de esta partición, se empieza a repetir una sucesión de etapas o cadenas de Markov, utilizando en la etapa m-ésima un valor fijo del parámetro de la "temperatura" t_m, el cual tiende a 0 cuando $m \to \infty$.

Dentro de cada cadena de Markov se van generando nuevas particiones $\tilde{P}$ escogidas *al azar* en el vecindario $\mathcal{V}(P)$ de la partición anterior P, aceptándolas de acuerdo con la siguiente probabilidad: $\min\{1\,,\, e^{-\Delta W/t_m}\}$. Esta regla de aceptación es conocida como la *Regla de Metropolis*. Aquí $\Delta W = W(\tilde{P}) - W(P)$ es el cambio de la inercia producida por la partición candidata $\tilde{P} \in \mathcal{V}(P)$. Las "buenas particiones" $\tilde{P}$ (aquellas que tienen inercia intra-clases $W(\tilde{P}) < W(P)$) son aceptadas siempre. Las "malas particiones" $\tilde{P}$ (aquellas que tienen inercia intra-clase $W(\tilde{P}) \geq W(P)$) son aceptadas con probabilidad

igual a $e^{-\Delta W/t_m}$, probabilidad que cada vez es más pequeña conforme "el sistema se va enfriando", pues cuando $m \to \infty$ el parámetro de la temperatura t_m tiende a 0.

Obsérvese que escoger una partición al azar del vecindario $\mathcal{V}(P)$ es equivalente a realizar al azar una transferencia de un objeto de una clase a otra en la partición P. Este método, propuesto y desarrollado por nosotros desde 1996 [Piz96], es similar al propuesto por Klein y Dubes [Kle90] en 1990.

Para cada temperatura fija t_m, la generación de nuevas particiones a través de este algoritmo establece un proceso de Markov aperiódico e irreducible y que por lo tanto posee —al menos en teoría— una distribución estacionaria límite. Por otra parte, es bien conocido que cuando t_m converge a 0 el algoritmo de sobrecalentamiento simulado converge —también en teoría— con probabilidad 1 a un mínimo global de la inercia intra-clases W. Los detalles acerca de la convergencia del método pueden verse en [Aar90, Laa88], por ejemplo.

En la práctica, la aproximación de estos dos procesos límites continuos por medio de cálculos discretos puede fácilmente destruir la optimalidad global del algoritmo. En particular, se requiere de especial cuidado en la definición precisa de los siguientes 4 aspectos, que son los que definen el "plan de sobrecalentamiento" empleado: i) la temperatura inicial t_0; ii) el "esquema de enfriamiento", esto es, la manera en que t_m tiende a 0; iii) el largo de las cadenas de Markov correspondientes a cada temperatura fija t_m; iv) la temperatura final, o sea, el criterio para detener el algoritmo. Nosotros hemos utilizado el siguiente plan de sobrecalentamiento, en forma exitosa en todas las tablas de datos que hemos analizado:

i) **Temperatura inicial t_0:** usamos uno de los enfoques más tradicionales, el cual consiste en calcular $t_0 := \overline{\Delta W}^+ / \ln(\chi_0^{-1})$, donde $\overline{\Delta W}^+$ es el promedio de los cambios en el criterio W para aquellas particiones que *incrementen* la inercia, o sea, que desmejoran el criterio, mientras χ_0 es un umbral inicial de aceptación de particiones que desmejoren el criterio W. Lo anterior se calcula luego de una serie de "ejecuciones en falso" del algoritmo. De esta manera se puede garantizar que al principio la Regla de Metropolis aceptará en promedio el $100\chi_0\%$ de particiones que incrementen la inercia [Laa88]. Hemos utilizado con éxito el valor de $\chi_0 = 0.8$ (80% de aceptación inicial de peores configuraciones).

ii) **Método de enfriamiento del sistema:** usamos el enfoque exponencial de enfriamiento, el cual consiste en calcular $t_{m+1} = \lambda \cdot t_m$, donde λ se escoje previamente en el intervalo $[0.9, 0.98]$. En todos los ejemplos que hemos analizado hasta el momento el valor de $\lambda = 0.95$ ha producido muy buenos resultados.

iii) **Largo de las cadenas de Markov:** Empleamos en el algoritmo el largo de las cadenas igual a $\min\{20000, 100n(k-1)\}$. Sin embargo, las iteraciones de cada cadena de Markov se detienen si ya se han aceptado $\min\{500, 10n(k-1)\}$ particiones que desmejoran el criterio.

iv) **Criterio de finalización del algoritmo:** Empleamos como máximo 150 iteraciones del valor de la temperatura t_m, aunque el algoritmo se interrumpe si en las últimas r temperaturas las cadenas de Markov no han generado ninguna aceptación

mediante la regla de Metropolis. En la práctica hemos utilizado con éxito el valor $r = 4$.

4.2 Búsqueda tabú

La técnica de la búsqueda tabú para encontrar óptimos globales en problemas de Optimización Combinatoria fue propuesta originalmente por Glover et al. en 1993 [Glo93].

Nuestro algoritmo, propuesto y desarrollado a partir de 1996 [Mur96], se inicia con cualquier partición inicial $P_0 \in \mathcal{P}_k$, por ejemplo escogida al azar. A partir de P_0 se escoge en forma iterativa una sucesión de particiones $(P_m)_{m\in\mathbf{N}}$.

En m-ésima etapa se elige la partición P_m del vecindario de P_{m-1} que minimize la inercia intra-clases $W(P)$, de todas las $n(k-1)$ particiones P que se pueden formar al transferir un objeto $\mathbf{x}_i \in \Omega$ a una clase C_ℓ. Sin embargo, se restringen las transferencias de objetos a clases que se encuentren dentro de una *lista tabú* de "transferencias prohibidas". Esta lista tabú tiene un largo t pre-establecido y está constituida en general por un conjunto de t indicatrices, siendo cada indicatriz aqulla la clase ℓ del individuo $\mathbf{x}_i$ al hacer la transferencia $(\mathbf{x}_i, \ell)$; la composición de la lista se renueva durante el transcurso del algoritmo. Puede observarse que —dependiendo del momento en el cual se realiza la transferencia— distintas transferencias $(\mathbf{x}_i, \ell)$ y $(\mathbf{x}_{i'}, \ell')$ podrían eventualmente conducir a una misma partición, así como una sola transferencia específica $(\mathbf{x}_i, \ell)$ de hecho conduce a diferentes particiones, dependiendo de la partición de partida.

Una nueva partición $P \in \mathcal{V}(P_{m-1})$ puede no ser escogida a pesar de ser la mejor disponible, si se forma a partir de una transferencia que se encuentre dentro de la lista tabú de las "tranferencias prohibidas". Sin embargo, en estas condiciones se hace una excepción en el caso que la nueva partición P sea "tan buena" que satisfaga algún *criterio de aspiración*. En nuestro algoritmo, el criterio de aspiración empleado es simple: si la inercia intra-clase $W(P)$ es la mejor encontrada hasta el momento, se admite la partición P, independientemente de si esta partición se formó a partir de una "transferencia prohibida" o no.

Si en el curso del algoritmo, en alguna etapa m se produjera un empate entre varias particiones $P \in \mathcal{V}(P_{m-1})$ que tienen igual inercia intra-clases $W(P)$ mínima, entonces se escoge *al azar* la nueva partición P_m dentro de ellas. Esto tiende a diversificar la búsqueda de particiones óptimas y previene en cierta forma el ciclaje del algoritmo.

La lista tabú de "transferencias prohibidas" previene al algoritmo de estancarse dentro de un posible óptimo local. El tamaño t de esta lista tabú es uno de los aspectos más delicados de escoger en el algoritmo, pues es allí donde radica la posible diferencia entre la convergencia del método a un óptimo local o a un óptimo global. Precisamente una de las dificultades del algoritmo es que el tamaño t de la lista tabú tiene un rápido crecimiento cuando el número n de objetos a clasificar es grande, o cuando el número de clases k es grande. Esto hace que la técnica de la búsqueda tabú sea poco eficiente para trabajar con grandes tablas de datos.

En general, para obtener buenos resultados deben realizarse algunas pruebas y ajustes del tamaño t de la lista tabú de "transferencias prohibidas". Otra dificultad teórica es que no se dispone de un criterio preciso para detener el algoritmo, luego de realizar un

considerable número de iteraciones. Sin embargo, en la práctica hemos obtenido muy buenos resultados con la búsqueda tabú, empleando tiempos de computación inferiores a los empleados con las otras técnicas, ajustando convenientemente el tamaño de t y los tiempos de cómputo.

La búsqueda tabú —por su comportamiento— es una técnica de tipo *pseudo-determinística*: el azar interviene solamante para decidir los eventuales empates entre inercias intra-clases de particiones, que se puedan producir en algún momento. Por otra parte, aunque se sabe por experiencia que la técnica de la búsqueda tabú tiende a encontrar óptimos globales en los problemas de optimización combinatoria, sin embargo no se conoce aún una demostración rigurosa de este hecho, razón por la cual se le sigue considerando como una técnica *heurística*. No obstante, hemos obtenido excelentes resultados en todas las tablas de datos de tamaño pequeño y mediano hasta ahora analizadas.

4.3 Algoritmo genético

Los algoritmos genéticos se fundamentan en algunos aspectos de la evolución natural de las especies. Mediante estos algoritmos se generan "poblaciones" de soluciones de un problema de Optimización Combinatoria, y se combinan estas soluciones mediante los llamados *operadores genéticos*, con el fin de encontrar mejores soluciones y evitar los óptimos locales. Los operadores genéticos clásicos son la *selección*, el *cruzamiento* y las *mutaciones*.

Los algoritmos genéticos fueron propuestos originalmente por Holland en 1976. Los detalles específicos sobre los mismos pueden ser consultados en el libro de Goldberg [Gol89]. La modelación por medio de cadenas de Markov garantiza la convergencia asintótica de estos algoritmos hacia el óptimo global del problema en estudio, según demostró Rudolph en 1994 [Rud94], bajo la condición de mantener de una iteración a la siguiente al mejor elemento de la población.

En nuestro problema de particionamiento de Ω en k clases, empleamos una "población" (una muestra) de particiones, descrita por "cromosomas" (particiones) de longitud n y con "genes" (objetos a clasificar) descritos con un alfabeto de k letras o "alelos". Por ejemplo, una partición podría ser $P = (2311332321)$, donde $n = 10$, $k = 3$, y $\mathbf{x}_1$ pertenece a la clase C_2, $\mathbf{x}_2$ pertenece a la clase C_3, etc. El método iterativo empleado [Tre96] se desarrolla de tal manera que en cada iteración son aplicados los siguientes operadores genéticos, de acuerdo con ciertas probabilidades de ocurrencia:

i) Las particiones $P \in \mathcal{P}_k$ son *seleccionadas* al azar, con probabilidad proporcional a su inercia inter-clases $B(P)$. De esta manera, las mejores particiones (aquellas con mayor inercia inter-clases $B(P)$) tienen mayores oportunidades de reproducirse, siguiendo los principios de la selección natural.

ii) Se utiliza un tipo de *mutación* que equivale a realizar una *transferencia* de un objeto de una clase a otra, con probabilidad p_m. Cuando ocurre este tipo de mutación se seleccionan al azar el "gen" u objeto a mutar y el "alelo" o índice de la clase al cual será transferido el objeto.

iii) Se utiliza otro tipo de *mutación* que equivale al *intercambio* entre las clases de dos de

los objetos, con probabilidad p_e. Cuando ocurre este tipo de mutación se seleccionan al azar los objetos que se intercambiarán.

iv) El *cruzamiento* entre dos "cromosomas" (particiones) seleccionados al azar. Se selecciona al azar el punto de corte a partir del cual los "genes" (objetos) de los "cromosomas" se intercambian y se combinan entre sí para formar nuevos "cromosomas". Esta operación genética ocurre con probabilidad p_c.

v) El *cruzamiento forzado* entre dos "cromosomas" (particiones) seleccionados al azar, mediante el cual una clase seleccionada al azar en la partición P_1 es copiada en la partición P_2. Esta operación genética ocurre con probabilidad p_f y genera un nuevo "hijo" P_2' que reemplazará a la partición P_2.

El *cruzamiento forzado* es una operación genética que ha demostrado ser mejor en la práctica que el simple *cruzamiento*, pues mantiene buena información sobre la partición de origen. En efecto, algunos de los objetos que están juntos en P_1, estarán juntos también en P_2', lo cual es una condición deseable cuando P_1 es un "cromosoma" que ha sobrevivido algún tiempo (esto es, algunas iteraciones), aumentando sus posibilidades de tratarse de una buena partición.

Hemos combinado estos operadores genéticos con el método clásico de los "k-means" de Forgy, tambin conocido como nubes dinmicas (con centros de gravedad) o centros móviles: después de cierto número de iteraciones, cada partición en la población converge hacia un óptimo local, si se aplica el método de los "k-means". Con cada solución obtenida, el algoritmo genético continua hasta que algún criterio de parada se satisfaga: i) cuando se alcanza un número máximo de iteraciones, o, ii) cuando la variancia de las inercias inter-clases $B(P)$ de la población es menor que alguna cota establecida de antemano.

5 Resultados numéricos

Hemos aplicado los tres métodos anteriormente descritos en muchas tablas de datos bien conocidas. En la Tabla 1 se muestran los resultados obtenidos en algunas de estas tablas de datos, comparando nuestros algoritmos con dos de los algoritmos tradicionales: "k-means" de Forgy y clasificación jerárquica de Ward (cortando el árbol de clasificación en el número de clases deseado).

Las tablas de datos aquí consideradas son: i) la tabla de *Datos Escolares* [Sch78], que es una tabla pequeña de 9 objetos y 5 variables; ii) los *Peces de Amiard* [Pag76], que es una tabla pequeña de 23 objetos y 16 variables; iii) La *Sociomatriz de Thomas* [Pag76], que es una tabla pequeña de 24 objetos y 24 variables; iv) los *Iris de Fisher* [Eve93], que es una tabla mediana de 150 objetos y 4 variables.

6 Conclusiones

Nuestras técnicas de particionamiento tienden a buscar particiones sub-optimales muy cercanas al óptimo global del problema en estudio, en tiempo y espacio de orden polinomial.

Notas Francesas	**SS** W	%	**BT** W	%	**AG** W	%	*k***M** W	%	**Ward** W
2 clases	28.2	100	28.2	100	28.2	100	28.2	12	28.8
3 clases	16.8	100	16.8	100	16.8	95	16.8	12	17.3
4 clases	10.5	100	10.5	100	10.5	97	10.5	5	10.5
Peces de Amiard	**SS** W	%	**BT** W	%	**AG** W	%	*k***M** W	%	**Ward** W
2 clases	69368	100	69849	96	69368	52	69849	49	—
3 clases	32213	100	32213	100	32213	87	32213	8	33149
4 clases	18281	100	18281	100	22456	90	18281	9	19589
5 clases	14497	100	14497	97	20474	38	14497	1	14497
Sociomatriz de Thomas	**SS** W	%	**BT** W	%	**AG** W	%	*k***M** W	%	**Ward** W
3 clases	271.8	100	271.8	100	271.8	85	271.8	2	279.3
4 clases	235.0	100	235.0	100	235.0	24	235.0	0.15	239.4
5 clases	202.4	100	202.6	98	223.8	4	202.6	0.02	204.7
Iris de Fisher	**SS** W	%	**BT** W	%	**AG** W	%	*k***M** W	%	**Ward** W
2 clases	0.999	100	0.999	100	0.999	100	0.999	100	—
3 clases	0.521	100	0.521	76	0.521	100	0.521	4	—
4 clases	0.378	55	0.378	60	0.378	82	0.378	1	—
5 clases	0.329	100	0.312	32	0.312	6	0.312	0.24	—

Table 1: *Mínima inercia intra-clases W encontrada, utilizando Sobrecalentamiento Simulado (SS), Búsqueda Tabú (BT), Algoritmos Genéticos (AG), el método de "k-means" de MacQueen-Forgy (kM), y el método de Ward. Aquí "%" indica el porcentaje de veces que se obtuvo la mejor solución, al repetir los análisis cientos de veces.*

Hasta el momento no hemos encontrado una colección de datos en donde los métodos tradicionales superen los resultados de las técnicas propuestas. Los tres métodos propuestos en este artículo requieren mayor tiempo de computación que los métodos clásicos, pero producen mejores resultados. Nosotros creemos que el incremento en la calidad de las particiones que se obtienen con nuestros métodos justifica plenamente su uso. Pareciera que los resultados obtenidos utilizando sobrecalentamiento simulado son superiores que los obtenidos con los otros métodos propuestos y con los métodos tradicionales, pues casi en todos los casos encontramos la mejor solución conocida del problema en estudio en el 100% de los experimentos realizados. Sin embargo, sobrecalentamiento simulado falló en encontrar la mejor solución conocida para la tabla del Iris de Fisher con 5 clases, mientras que con 4 clases encontró la mejor solución conocida solamente en el 55% de los experimentos.

La búsqueda tabú es también una excelente técnica para realizar particionamiento con tablas de datos medianas y pequeñas, y es algo más rápida que sobrecalentamiento simulado. Por otra parte, el algoritmo genético produjo mejores resultados que los algoritmos de "k-means" y la técnica jerárquica de Ward, aunque estos resultados son inferiores a los obtenidos con los algoritmos de sobrecalentamiento simulado y búsqueda tabú, aparte que su implementación es muy lenta.

A pesar de los buenos resultados obtenidos, se necesita aún afinar los detalles acerca de la correcta escogencia de parámetros en nuestras técnicas: en sobrecalentamiento simulado, la escogencia adecuada del largo de las cadenas de Markov y el esquema de enfriamiento; en búsqueda tabú, la escogencia adecuada del tamaño de la lista tabú y el criterio de parada del método; en algoritmos genéticos quedan muchos aspectos por investigar, en especial la escogencia adecuada de las probabilidades para los operadores genéticos y el tamaño de la población.

Algunas generalizaciones de estos métodos están siendo actualmente investigadas, como por ejemplo el empleo de técnicas mixtas, tal como el método de sobrecalentamiento simulado incorporándole una lista tabú de transferencias prohibidas, o un nuevo esquema de enfriamiento/calentamiento en el método de sobrecalentamiento simulado, llamado el "método del acordeón", algoritmos que prometen obtener excelentes resultados, especialmente en el particionamiento de tablas de gran tamaño. También estamos investigando la generalización de los métodos propuestos trabajando con distancias diferentes a las distancias euclídeas, así como con objetos caracterizados por variables no-numéricas.

References

[Aar90] Aarts, E.; Korst, J. (1990) *Simulated Annealing and Boltzmann Machines. A Stochastic Approach to Combinatorial Optimization and Neural Computing*. John Wiley & Sons, Chichester.

[Pag76] Cailliez, F.; Pagès, J.P. (1976) *Introduction à l'Analyse des Données*. SMASH, París.

[Cel89] Celeux, G.; Diday, E.; Govaert, G.; Lechevallier, Y.; Ralambondrainy, H. (1989) *Classification Automatique des Données. Environnement Statistique et Informatique*. Dunod-Informatique, París.

[Čer85] Černy, V. (1985) "Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm", *Journal of Optimization Theory and Applications* **45**: 41–51.

[Did80] Diday, E. y colaboradores (1980) *Optimisation en Classification Automatique*, 2 tomos. INRIA, Rocquencourt.

[Did82] Diday, E.; Lemaire, J.; Pouget, J.; Testu, F. (1982) *Eléments d'Analyse de Données*. Dunod, París.

[Eve93] Everitt, B.S. (1993) *Cluster Analysis*. 3a edición. Edward Arnold, Londres.

[Glo93] Glover, F. et al. (1993) "Tabu search: an introduction", *Annals of Operations Research*, **41**(1–4): 1–28.

[Gol89] Goldberg, D. E. (1989) *Genetic Algorithm in Search, Optimization and Machine Learning*. Addison-Wesley, Reading-Mass.

[Kir83] Kirkpatrick, S.; Gelatt, D.; Vecchi, M.P. (1983) "Optimization by simulated annealing", *Science* **220**: 671–680.

[Kle90] Klein, R. W.; Dubes, R. C. (1990) "Experiments in projection and clustering by simulated annealing", *Pattern Recognition* **22**: 213–220.

[Knu81] Knuth, D.E. (1981) *Seminumerical Algorithms*. Segunda edición, volumen 2 del libro *The Art of Computer Programming*. Addison-Wesley, Reading, Mass.

[Laa88] Laarhoven, P.; Aarts, E.; Korst, J. (1988) *Simulated Annealing: Theory and Applications*. Kluwer Academic Publishers, Dordrecht.

[Ler81] Lerman, I.C. (1981) *Classification et Analyse Ordinale des Données*. Dunod, París.

[Mur96] Murillo, A.; Trejos, J. (1996) "Classification tabou basée en transferts",*IV Journ. Soc. Frac. Classif.*, S. Joly & G. Le Calvé (eds.), Vannes: 26.1–26.4.

[Piz87] Piza, E. (1987) "Clasificación Automática Jerárquica Aglomerativa", *Revista de Ciencias Económicas* **7**(1).

[Piz96] Piza, E.; Trejos, J. (1996) "Partitionnement par recuit simulé", *IV Journ. Soc. Frac. Classif.*, S. Joly & G. Le Calvé (eds.), Vannes: 27.1–27.4.

[Pre90] Press, W.H.; Flannery, B.P.; Teulolsky, S.A.; Vetterling, W.T. (1990) *Numerical Recipes (Fortran Version). The Art of Scientific Computing*. Cambridge University Press, New York.

[Rud94] Rudolph, G. (1994) "Convergence of a genetic algorithm", *IEEE Transactions on Neural Networks* **5**(1), 96–101.

[Sch78] Schektman, Y. (1978) "Estadística Descriptiva", I parte, *Memorias I Simposio Métodos Matemáticos Aplicados a las Ciencias*, J. Badia, Y. Schektman y J. Poltronieri (eds.), Universidad de Costa Rica, San Pedro: 9–67.

[Tre96] Trejos, J. (1996) "Un algorithme génétique de partitionnement", *IV Journ. Soc. Frac. Classif.*, S. Joly & G. Le Calvé (eds.), Vannes: 31.1–31.4.

[Tre98] Trejos, J.; Piza, E.; Murillo, A. (1998) "Global stochastic optimization techniques applied to partitioning", *Advances in Data Science and Classification*, M. Rizzi, M. Vichi & H.-H, Bock (eds.), Springer–Verlag, Berlin: 185–190.