Facultad de Matemática y Computación
Universidad de La Habana

Metaheurísticas de Memoria Adaptativa para el Problema de Agrupamiento.

Autor: Arnaldo Pérez Castaño
Tutor: Dr. Ricardo Beausoleil
Cotutor: Msc. Yasser Valcárcel

Trabajo de Diploma presentado en opción al título de Licenciado en Ciencia de la Computación

La Habana, junio 2013

Resumen

El Agrupamiento es un problema clásico de la computación que encuentra aplicaciones en los más diversos campos. Pertenece al área del aprendizaje no supervisado que se engloba dentro del aprendizaje de máquinas una disciplina de la inteligencia artificial. La Zonificación por otra parte es un problema de estudios urbanos, una disciplina asociada a la arquitectura. Las Metaheurísticas de Memoria Adaptativa representan estrategias favorables para encontrar soluciones a estos problema debido a que desarrollan una búsqueda inteligente, basada en el empleo de estructuras de memoria. En este documento se describirán Metaheurísticas para resolver los problemas de Agrupamiento clásico y de Zonificación, mostrando los resultados de dichos algoritmos en conocidos conjuntos de datos, teniendo en cuenta medidas externas para evaluar el agrupamiento y realizando comparaciones con los resultados obtenidos por otros algoritmos.

Agradecimientos

A mi familia por todo el apoyo brindado, a mi tutor y cotutor por toda la paciencia y la comprensión mostrada.

Opinión del tutor

En nuestra opinión el autor del trabajo: *Metaheurísticas de Memoria Adaptativa para el Problema de Agrupamiento*, del estudiante de 5to año de Ciencias de la Computación Arnaldo Pérez Castaño que opta por el título de Licenciado en Ciencias de la Computación, es un excelente alumno, dedicado y con alto grado de independencia. La tesis elaborada por el aspirante consta de una introducción, un primer capítulo de Preliminares que cuenta con dos epígrafes, un segundo capítulo de Algoritmos de agrupamiento, un tercer capítulo de Búsqueda Tabú y un último en el que se exponen los Resultados Computacionales obtenidos y finaliza con las Conclusiones y trabajo futuro.

El aspirante ha trabajado arduamente en esta nueva rama de las matemáticas y ciencias computacionales desde sus Prácticas Investigativas de 4to año, realizando una revisión bibliográfica, así como un amplio estudio de técnicas de memoria adaptativa y conceptos que condujeron al aspirante al desarrollo de adecuaciones de un algoritmo Tabú para la solución de problemas de Optimización Multiobjetivo.

El trabajo realizado se refleja en la participación en la Jornada Científica ICIMAF 2013 culminando con una memoria del evento ISBN: 978-959-7056-33-1. El aspirante demuestra tener dominio de la materia, las adecuaciones al algoritmo son novedosas, la tesis en tres capítulos es coherente y la literatura científica utilizada es actual.

Por las razones antes expuestas, consideramos se le otorgue al aspirante Arnaldo Pérez Castaño el grado de Licenciado en Ciencias de la Computación.

Tutores.

Dr. Ricardo P. Beausoleil

MSc. Yasser Valcárcel Miró

Índice general

Introducción

En la presente tesis se describen *Metaheurísticas* de Memoria Adaptativa, cuya finalidad es encontrar soluciones al Problema de Agrupamiento. Las Metaheurísticas son métodos de optimización, más específicamente algoritmos de aproximación que se apoyan en heurísticas para dirigir la búsqueda en el espacio solución. En este caso las Metaheurísticas estarán embebidas en un marco multiobjetivo, de modo que se optimizan varias funciones al mismo tiempo y la salida del algoritmo es la imagen del conjunto Pareto, conocido como Frente Pareto.

El Problema de Agrupamiento (en inglés *clustering*) consiste en particionar un conjunto de n objetos en k grupos de modo tal que objetos de un mismo grupo tengan la mayor similitud posible y objetos en grupos distintos posean la mayor disimilitud posible. La medida de similitud puede definirse de varias formas, usualmente se piensa en la distancia euclidiana, pero en general cualquier medida puede ser válida y depender del contexto del problema.

Como problema forma parte de la clase de complejidad NP-completo y puede verse dentro del campo de la optimización combinatoria. Los algoritmos de agrupamiento forman parte de la inteligencia artificial, más específicamente de una de sus ramas conocida como aprendizaje de máquinas que se divide en dos grupos: los algoritmos de aprendizaje no

supervisado y los de aprendizaje supervisado. El Agrupamiento es una técnica del aprendizaje no supervisado, esto se traduce en que no requieren de un conjunto de datos inicial para entrenarse o aprender, de modo que predicen la clase de un nuevo dato sin requerir una primera fase de entrenamiento.

El Agrupamiento es un problema clásico de la computación que encuentra aplicaciones en las más diversas áreas de la ciencia como pueden ser la Minería de Datos, el procesamiento de lenguajes naturales, la teoría de señales, la recuperación de información o el análisis de la formación de galaxias.

Su gran utilidad en diversos campos de la ciencia y la importancia de las Metaheurísticas como métodos para brindar soluciones a problemas complejos definen la motivación principal de esta tesis.

Las Metaheurísticas de Memoria Adaptativa brindan soluciones satisfactorias al Problema de Agrupamiento. Esta es la hipótesis que se plantea en el presente trabajo.

De esta forma los objetivos fundamentales de esta tesis serían, primero, **desarrollar algoritmos de Memoria Adaptativa que resuelvan satisfactoriamente Problemas de Optimización Multiobjetivo, en particular el Problema de Agrupamiento. Además se propone como objetivo la comparación de los resultados obtenidos con algoritmos del estado del arte, aunque NO constituya un objetivo de la presente tesis la obtención de resultados competitivos con los del estado del arte.**

Primeramente se dedicará un primer capítulo a presentar definiciones que resultan indispensables para comprender el contenido de esta tesis. En el segundo capítulo se examinan un grupo de algoritmos tradicionales

que brindan soluciones al Problema de Agrupamiento. En tanto el tercer capítulo estará dedicado a describir una importante Metaheurística de Memoria Adaptativa: la Búsqueda Tabú. Además se presentarán dos algoritmos de este tipo, uno que brinda soluciones al Problema de Agrupamiento clásico y el otro al problema de Zonificación. El cuarto capítulo presentará resultados computacionales de los algoritmos descritos en el capítulo anterior, realizando comparaciones. Finalmente, se darán las conclusiones y las recomendaciones para trabajo futuro.

Capítulo 1

Preliminares

1.1. Teoría y definiciones

A continuación se brindan algunas definiciones que resultan imprescindibles para comprender el funcionamiento de los algoritmos que se describen en este documento.

Definición 1.1.1 (Problema de optimización multiobjetivo) *Un problema de optimización multiobjetivo se puede formular como:*

$$min\ F(x) = (f_1(x), f_2(x), ..., f_q(x))$$

$$sujeto\ a\ X \in S$$

donde S representa el espacio factible.

Definición 1.1.2 (Dominado) *Se dice que un vector $v = (v_1, v_2, ..., v_n)$ domina a otro vector $u = (u_1, u_2, ..., u_n)$ si y solo si se cumple $u_i \leq v_i$ y $\exists i$ tal que $u_i < v_i$.*

Definición 1.1.3 (Pareto óptimo) *Una solución factible x se dice que es Pareto óptima si y solo si $\nexists y$ tal que $F(y)$ domine a $F(x)$.*

Definición 1.1.4 (Conjunto Pareto) *Es el conjunto de soluciones Pareto óptimo, que se denominará* P.

Definición 1.1.5 (Frente Pareto) *Se define como la imagen de* P*, que se denominará* F.

Definición 1.1.6 (PSET) *Es la herramienta encargada de mantener y actualizar* F.

Definición 1.1.7 (Vector Nadir) *Un punto* $y^* = (y_1^*, y_2^*, ..., y_q^*)$ *es Nadir si* $y_i^* = max(f_i(x))$ *con* $x \in P$.

Definición 1.1.8 (Vector Ideal) *Un punto* $y^* = (y_1^*, y_2^*, ..., y_q^*)$ *es Ideal si* $y_i^* = min(f_i(x))$ *con* $x \in P$.

Definición 1.1.9 (Punto de referencia) *Un punto de referencia es un vector* $z = (z_1, z_2, ..., z_q)$ *que define el nivel de aspiración que se desea alcanzar en cada función objetivo* f_i.

El vector ideal es casi imposible de alcanzar en la mayoría de los problemas de la vida real, así que muchas veces se emplea el punto de referencia para determinar cuando una solución Pareto óptima es aceptable. El punto ideal y el Nadir brindan información acerca de los rangos del Frente Pareto.

La siguiente definición está estrechamente vinculada con Metaheurísticas de búsqueda que definen una estructura de vecindad.

Definición 1.1.10 (Pareto óptimo local) *Una solución* x *es Pareto óptima local si y solo si* $\forall y \in N(x)$, $F(x)$ *domina a* $F(y)$, *donde* $N(x)$ *representa la vecindad de* x.

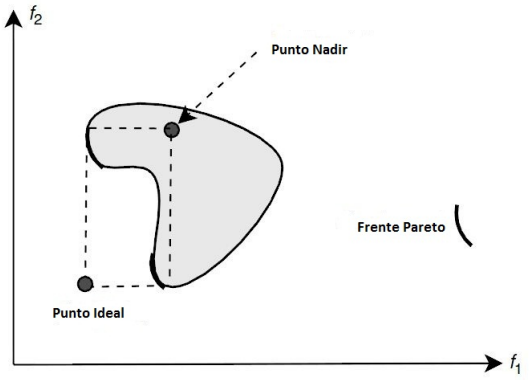

Figura 1.1: Punto ideal y Nadir para un caso bi-objetivo.

1.2. Optimización multiobjetivo

La optimización multiobjetivo tiene sus inicios en el siglo XIX en los trabajos de economía de Edgeworth y Pareto. Comenzó siendo utilizada en este campo y gradualmente ha pasado a ser vital en la ingeniería y en diferentes ciencias. Encuentra diversas aplicaciones ya que se ajusta a problemas de la vida real donde deben tenerse en cuenta varios objetivos que están en conflicto. Por ejemplo, si se quisiera encontrar un terreno para empezar un negocio se debería intentar minimizar el costo y maximizar la calidad del terreno, basar la elección de acuerdo a solo uno de los criterios anteriores puede resultar en un terreno muy barato pero de poca calidad o en un terreno bueno pero extremadamente caro. Aquí aparece el rol del *decisor*, quien es el encargado de decidir, de todas las soluciones, aquella que va de acuerdo a sus necesidades.

Las Metaheurísticas multiobjetivo comenzaron a ser ampliamente investigadas a finales de los 80s del siglo pasado, en un interés por resolver complejos problemas multiobjetivo. La solución de un problema multiobjetivo generalmente no se reduce a una única solución sino a un conjunto

de soluciones Pareto óptimas que definen el Frente Pareto. El objetivo de una Metaheurística multiobjetivo es obtener una aproximación del Frente Pareto que cumpla las siguientes dos propiedades:

- Convergencia: garantiza soluciones que brindan una buena aproximación al Frente Pareto óptimo.
- Uniformidad: garantiza una buena distribución de las soluciones obtenidas a lo largo del Frente Pareto óptimo evitando la pérdida de información valiosa.

Un aspecto final de un PMO resulta la etapa del *decisor* quien es el encargado de decidir finalmente, del conjunto Pareto, cuales serán las soluciones que se ajusten a sus intereses. Los PMOs se dividen en dos categorías fundamentales: los continuos y los combinatorios. Los primeros tratan con soluciones codificadas con variables continuas mientras que los segundos cuentan solo con variables que toman valores discretos. La gran mayoría de los trabajos realizados desde hace más de 40 años pertenecen a los continuos.

Capítulo 2

Algoritmos de agrupamiento

En este capítulo se describen algunos de los más populares algoritmos de agrupamiento. El conocimiento de dichos algoritmos resulta indispensable para abordar el estudio del Problema de Agrupamiento y se dividen en dos grupos que se verán en secciones continuas, estos son: los de particionamiento y los jerárquicos.

2.1. Algoritmos de particionamiento

El objetivo que persiguen los algoritmos de particionamiento, es precisamente obtener una partición de un conjunto de datos en k grupos o clases de manera que se maximice o minimice una determinada función objetivo. Hallar la partición óptima siempre sería posible si se enumeran todas las posibles particiones, pero esto resulta completamente infactible para una computadora, debido a su costo temporal exponencial. Por ejemplo, para un problema de 30 objetos, con 3 particiones, el número de particiones posibles es de aproximadamente $2 * 10^{14}$, haciendo indispensable el uso de algoritmos que empleen heurísticas para encontrar

soluciones aproximadas al óptimo.

2.1.1. K-Medias

El K-Medias es uno de los más populares algoritmos de agrupamiento, considerado por muchos como elemental debido a su fácil implementación. Intenta llegar a una partición óptima minimizando la suma de errores cuadrados, con un procedimiento iterativo, semejante a una búsqueda local. Ofrece buenos resultados cuando los grupos resultantes son compactos y tienen la forma de una hiperesfera, además la complejidad computacional es aproximadamente lineal lo cual lo convierte en una buena opción para agrupar grandes conjuntos de datos. A pesar de las ventajas mencionadas, K-means padece de problemas heredados de la búsqueda local, uno de estos problemas es la necesidad de definir la cantidad de particiones a priori, algo poco común en problemas de la vida real. El algoritmo se puede resumir en los siguientes pasos.

1. Inicializar una k-partición de manera aleatoria, definiendo los k centroides.

2. Asignar cada objeto al grupo cuyo centroide resulta ser el más cercano.

3. Recalcula los centroides basado en:

$$m_i = \frac{1}{N_i} \sum_{x_j \in C_i} x_j$$

   donde C_i es el grupo del centroide m_i.

4. Repetir pasos 2 y 3 hasta que ningún grupo cambie.

La condición de parada no tiene que ser precisamente que ningún grupo cambie, pudiera ser que se ha alcanzado un número prefijado de iteraciones, que ningún centroide ha cambiado o que se ha logrado un valor de la distancia intra-clase que supere un umbral prefijado.

2.2. Algoritmos jerárquicos

Los algoritmos jerárquicos agrupan los objetos como una secuencia de particiones anidadas. Los resultados del agrupamiento jerárquico son usualmente representados por un árbol binario o dendograma. La raíz del dendograma representa todo el conjunto de objetos mientras que cada hoja representa uno de esos objetos. Por tanto, los nodos intermedios describen la proximidad entre objetos y la altura del árbol expresa la distancia entre cada par de objetos, entre grupos o entre un grupo y un objeto. En la figura 2.1 se muestra un dendograma. La dirección del agrupamiento aglomerativo jerárquico es de abajo hacia arriba, mientras que la del divisivo jerárquico es en el sentido contrario.

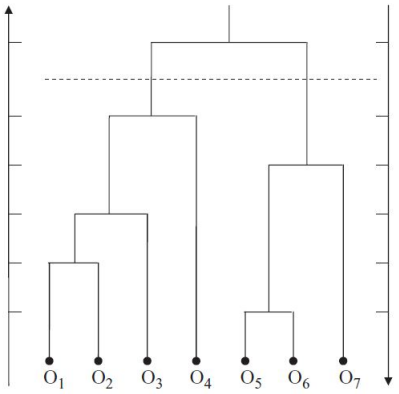

Figura 2.1: Ejemplo de dendograma.

Existen varias formas de lograr un agrupamiento, una estrategia sería comenzar con tantos grupos como objetos (un objeto por grupo) hasta llegar a obtener un grupo con todos los objetos, otra estrategia, sería comenzar con todos los objetos en un grupo hasta llegar a tener un grupo por cada objeto. Estas estrategias reciben el nombre de aglomerativas y divisivas respectivamente.

La crítica común a los algoritmos de agrupamiento jerárquico es su costo temporal, nunca inferior a $O(N^2)$, que representa una limitante para aplicaciones de gran escala. Los métodos divisivos requieren considerar $2^{N-1} - 1$ posibles divisiones de dos subconjuntos para un grupo con N objetos lo cual es muy costoso computacionalmente, por este motivo los métodos aglomerativos son más utilizados, ellos son analizados a continuación.

2.2.1. Aglomerativo general

El agrupamiento aglomerativo comienza con N grupos, uno por objeto, luego se realiza una serie de operaciones de mezcla hasta llegar a un grupo incluyendo todos los objetos. El procedimiento se resume a continuación.

Para decidir cuales son los dos grupos más cercanos existen diferentes medidas. Seguidamente se describen algunas de ellas.

2.2.2. Medidas de distancia entre grupos

La mezcla de grupos depende definitivamente de la medida utilizada para determinar la distancia entre grupos. Un gran número de estas medidas se ven generalizadas en la fórmula de recurrencia propuesta por

Algoritmo 1: Aglomerativo general

1. Comienza con N grupos simples (de un solo elemento). Calcula la distancia entre todo par de grupos.
2. Encuentra los dos grupos C_i ,C_j más cercanos y combinálos en un nuevo grupo $C_{i,j}$.
3. Repite el paso anterior hasta que solo quede un grupo.

Lance, Williams (1967) y que posee la siguiente forma:

$$D(C_l,(C_i,C_j)) = \alpha_i D(C_l,C_i) + \alpha_j D(C_l,C_j) + \beta D(C_i,C_j)$$

$$+\gamma|D(C_l,C_i) - D(C_l,C_j)|$$

donde D es la función de distancia y α, β, γ son parámetros que toman valores en dependencia del esquema utilizado. A continuación se mostrarán algunos esquemas:

- Enlace simple: la distancia entre dos grupos esta dada por la distancia entre los dos objetos más cercanos de cada grupo. Cambien llamado método del vecino más cercano.

  $$D(C_l,(C_i,C_j)) = min(D(C_l,C_i),D(C_l,C_j))$$

  En este caso los parámetros de la fórmula general toman valores: $\alpha = 1/2, \beta = 0, \gamma = -1/2$. Efectivo si los grupos están separados entre sí.

- Enlace completo: utiliza la distancia máxima entre un par de objetos de grupos diferentes para definir la distancia entre grupos.

  $$D(C_l,(C_i,C_j)) = max(D(C_l,C_i),D(C_l,C_j))$$

Efectivo en descubrir pequeños y compactos grupos.

- Enlace promedio de grupo: la distancia entre grupos se define como el promedio de la distancia entre cada par de objetos donde cada uno pertenece a un grupo distinto.

$$D(C_l,(C_i,C_j)) = \frac{1}{2}(D(C_l,C_i),D(C_l,C_j))$$

- Enlace de centroides: dos grupos son mezclados dependiendo de la distancia entre sus centroides definida como:

$$m_i = \frac{1}{n_i}\sum_{x\in C_i} x$$

donde n_i es el número de objetos en el grupo i. La fórmula general para este caso adopta la forma:

$$D(C_l,(C_i,C_j)) = \frac{n_i}{n_i+n_j}D(C_l,C_i)+$$

$$\frac{n_j}{n_i+n_j}D(C_l,C_j) - \frac{n_i n_j}{(n_i+n_j)^2}D(C_i,C_j)$$

Lo cual es equivalente a la distancia Euclidiana entre los centroides de dos grupos:

$$D(C_l,(C_i,C_j)) = ||m_l - m_{i,j}||^2$$

2.2.3. Divisivo jerárquico

El agrupamiento divisivo procede de forma opuesta al aglomerativo. En un principio todos los objetos pertenecen a un mismo grupo y sucesivamente se van dividiendo hasta que existan tantos grupos como objetos, o sea, un grupo por objeto.

Para un conjunto de N objetos, un algoritmo de este tipo comenzaría considerando las $2^{N-1} - 1$ posibles divisiones de los objetos en dos subconjuntos no vacíos, lo cual posee un alto costo computacional, por lo cual el agrupamiento divisivo es realmente poco utilizado en la práctica. Aún así brinda una clara visión de la estructura de los objetos, dado que los grupos más grandes son generados en fases iniciales del proceso de agrupamiento y son menos propensos a sufrir de decisiones erróneas acumuladas, que una vez cometidas son arrastradas a lo largo del proceso.

Uno de los métodos para resolver el agrupamiento divisivo es un algoritmo heurístico conocido como DIANA (DIvisive ANAlysis) que solo considera una parte de todas las posibles divisiones. El grupo con el máximo diámetro es seleccionado para ser dividido. Suponiendo que el grupo C_l será dividido en los grupos C_i y C_j los pasos de DIANA se pueden apreciar en el algoritmo 2.

Entre las críticas que suelen apuntarse de los algoritmos divisivos clásicos está su alta sensibilidad ante el ruido, que se traduce en la presencia de errores en los datos debido a diferentes factores en las etapas de medición, almacenamiento y procesamiento, que no son manejados adecuadamente por los algoritmos divisivos, y que afectan la forma de los grupos, además de distorsionar el algoritmo. Una vez que un objeto es asignado a un grupo no se le vuelve a considerar, lo cual significa que no es posible corregir una mala clasificación.

Algoritmo 2: DIANA

1. $C_i = C_l$ y $C_j = \emptyset$.

2. Para cada objeto $x_m \in C_i$

   a) Para la primera iteración calcula su distancia promedio hacia todos los objetos.

   $$d(x_m, C_i \setminus \{x_m\}) = \frac{1}{N_{c_i}-1} \sum_{x_p \in C_i, p \neq m} d(x_m, x_p)$$

   b) Para las iteraciones restantes calcula la diferencia entre la distancia promedio a C_i y la distancia promedio a C_j:

   $$d(x_m, C_i \setminus \{x_m\}) - d(x_m, C_j) =$$
   $$\frac{1}{N_{c_i}-1} \sum_{x_p \in C_i, p \neq m} d(x_m, x_p) - \frac{1}{N_{c_j}} \sum_{x_q \in C_j} d(x_m, x_q)$$

3. **a)** Para la primera iteración mueve el objeto con máximo valor según **2** hacia C_j.

   b) Para las iteraciones restantes, si el máximo valor de $2b$) es mayor que 0, mueve el objeto con la máxima diferencia a C_j. Repite $2b$) y $3b$). En caso contrario para.

2.3. Medidas de similitud

En el agrupamiento, la función objetivo resulta fundamental para obtener una solución óptima. Es esta función la encargada de evaluar la homogeneidad de los elementos dentro de cada grupo o de evaluar lo diferentes que son los grupos entre sí.

2.3.1. Intra-clase

Una de las funciones más utilizadas es la *suma de los errores cuadrados* en ocasiones llamada distancia *intra-clase*. La idea detrás de esta función es minimizar la diferencia o error que existe en similitud entre cada elemento de un grupo y el centroide o representante de ese grupo. El centroide se calcula como el vector promedio:

$$m_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$$

La distancia intra-clase se puede formular como:

$$Intra = \sum_{i=1}^{k} \sum_{j=1}^{n} \alpha_{i,j} ||x_j - m_i||^2$$

donde m_i representa el centroide del i-ésimo grupo y $\alpha_{i,j}$ es una variable binaria que toma valor 1 solo si x_j se encuentra en el i-ésimo grupo, además $\sum_{i=1}^{k} \alpha_{i,j} = 1, \forall j$. La partición que minimice la suma de los errores cuadrados se considera óptima.

2.3.2. Inter-clase

Otra función de similitud ampliamente difundida es la *inter-clase*, que tiene como premisa fundamental garantizar que la distancia entre los grupos sea la mayor posible, con el objetivo de que objetos en grupos distintos tengan la mayor disimilitud. La idea es maximizar su valor y puede formularse como:

$$Inter = \sum_{i=1}^{k} \sum_{j=i+1}^{k} d(m_i, m_j)$$

donde d es una función que mide la distancia entre m_i y m_j, ambos centroides.

2.3.3. Diámetro

El diámetro se define sobre un grupo C y mide la distancia máxima entre dos elementos del mismo. Se puede formular como:

$$D = max\ d(x_i, x_j)\ \ x_i \in C, x_j \in C$$

2.3.4. Radio

Se define para un grupo C, mide la distancia mínima de las máximas distancias de un elemento a los restantes. Se puede formular como:

$$Radio = min\ (max_{x_j \in C}\ d(x_j, x_k))\ \ \forall x_k \in C$$

2.4. Medidas para evaluar el agrupamiento

Una cuestión de vital importancia es la calidad del agrupamiento brindado por un algoritmo. Para alcanzar una evaluación del mismo se han creado diferentes medidas que determinan la calidad del agrupamiento. En esta sección, X se refiere al conjunto de datos y C al agrupamiento ofrecido por un determinado algoritmo. A continuación se describen algunas de estas medidas.

2.4.1. Medidas internas

Las medidas o criterios internos evalúan la estructura del agrupamiento exclusivamente desde X, sin ninguna información externa. Para llegar a una evaluación utilizarían, por ejemplo, la matriz de similitud para evaluar la validez de C.

Las medidas internas al igual que las externas (que serán introducidas en la siguiente sección) están fuertemente relacionadas con métodos estadísticos, pruebas de hipótesis.

El Coeficiente de Correlación Copenético (CCC) es un índice empleado para validar estructuras de agrupamiento jerárquico. Dada la matriz de similitud S de X. El CCC mide el grado de similitud entre S y la matriz copenética Q, cuyos elementos almacenan el nivel de proximidad donde pares de objetos son agrupados en el mismo grupo por primera vez. Sean μ_s y μ_q las medias de S y Q respectivamente.

$$\mu_s = \frac{1}{M} \sum_{i=1}^{N-1} \sum_{i=j+1}^{N} s_{i,j}$$

$$\mu_q = \frac{1}{M} \sum_{i=1}^{N-1} \sum_{i=j+1}^{N} q_{i,j}$$

donde $M = N(N-1)/2$, CCC se define como:

$$CCC = \frac{\frac{1}{M} \sum_{i=1}^{N-1} \sum_{i=j+1}^{N} s_{i,j} q_{i,j} - \mu_s \mu_q}{\sqrt{(\frac{1}{M} \sum_{i=1}^{N-1} \sum_{i=j+1}^{N} s_{i,j}^2 - \mu_s^2)(\frac{1}{M} \sum_{i=1}^{N-1} \sum_{i=j+1}^{N} q_{i,j}^2 - \mu_q^2)}}$$

El valor de CCC se encuentra en el intervalo $[-1, 1]$ y un valor cercano a 1 determina una alta similitud entre S y Q.

2.4.2. Medidas externas

Si P es una partición especificada a priori para X tal que $|X| = N$, entonces la evaluación externa de C se consigue comparando C con P. Considerando un par de puntos $x_i, x_j \in X$, existen 4 casos de como pudieran estar ubicados en C y en P.

1. x_i, x_j pertenecen al mismo grupo en C y a la misma clase en P.

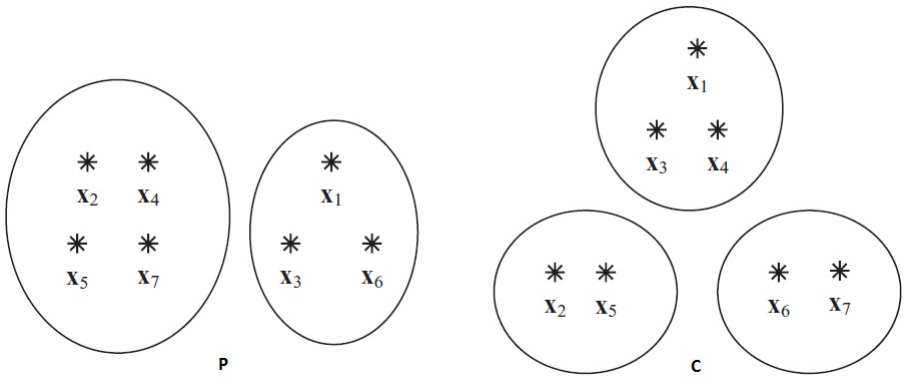

Figura 2.2: Ejemplo de una partición especificada a priori P y un agrupamiento C, brindado por el algoritmo.

2. x_i, x_j pertenecen al mismo grupo en C y a distinta clase en P.

3. x_i, x_j pertenecen a distinto grupo en C y a la misma clase en P.

4. x_i, x_j pertenecen a distinto grupo en C y a distinta clase en P.

En la figura 2.2 se puede apreciar un ejemplo donde se ponen en evidencia estos 4 casos. La Tabla 2.1 analiza dichos casos.

Casos	Pares de puntos	Total
1	$(x_1, x_3); (x_2, x_5)$	2
2	$(x_1, x_4); (x_3, x_4); (x_6, x_7)$	3
3	$(x_1, x_6); (x_2, x_4); (x_2, x_7); (x_3, x_6)$	7
	$(x_4, x_5); (x_4, x_7); (x_5, x_7)$	
4	$(x_1, x_2); (x_1, x_5); (x_1, x_7); (x_2, x_3); (x_2, x_6)$	9
	$(x_3, x_5); (x_3, x_7); (x_4, x_6); (x_5, x_6)$	

Tabla 2.1: Análisis del ejemplo 3.1.

El número de pares de puntos para los cuatro casos se denotan como a, b, c, d. Como el número total de puntos es $M = N(N-1)/2$, se tiene $M = a + b + c + d$.

Algunos índices externos que frecuentemente son utilizados para medir la similitud entre C y P se describen a continuación.

1. Rand Index:

$$R = \frac{a+d}{M}$$

2. Coeficiente de Jaccard:

$$J = \frac{a}{(a+b+c)}$$

Como puede apreciarse en la definición, los valores de estas medidas están en el intervalo $[0, 1]$, para valores cercanos a 1 mayor será la similitud entre P y C.

Capítulo 3

Búsqueda Tabú

La Búsqueda Tabú (BT) es una Metaheurística presentada por Fred Glover, que utiliza Memoria Adaptativa y exploración sensitiva y que además hereda de Búsqueda Local (BL), una de las más antiguas y simples Metaheurísticas. Es posible considerar que Búsqueda Tabú es básicamente una Búsqueda Local pero con numerosas y significativas mejoras o evoluciones. El núcleo de BT funciona de manera similar al de BL; comienza en una determinada solución (usualmente generada aleatoriamente), realiza iteraciones hasta que se logra una condición de parada y cada iteración reemplaza la solución actual por otra que mejora la función objetivo, y es hallada en la vecindad de la solución actual. La regla de parada para BL se alcanza cuando ningún vecino de la solución actual logra una mejora en el valor de la función objetivo lo cual se traduce en un mínimo local. Esta es la principal desventaja para BL; lo mejor que puede encontrar es un óptimo local, una desventaja que Búsqueda Tabú no comparte dado que incluye mecanismos de diversificación lo cual prevee que quede estancado en un óptimo local. El término diversificación se refiere a la exploración del espacio de búsqueda, intentando visitar regiones no

exploradas del espacio solución, la intensificación por otro lado intenta favorecer la explotación de las mejores soluciones encontradas hasta ahora, en este caso las regiones más prometedoras son exploradas más a fondo.

Es usual ver la BT aplicada a la optimización combinatoria en problemas como el del viajante TSP (Travelling Salesman Problem) o el de asignación cuadrática QAP (Quadratic Assignment Problem). El uso de Metaheurísticas para resolver el problema de Zonificación, así como el del viajante (TSP) es obligatorio, por pertenecer ambos a la clase de problemas NP Completo. De hecho, por lo general, no se hallan soluciones óptimas sino aproximaciones a estas soluciones óptimas y algunas veces estas aproximaciones pueden igualar a alguna de estas soluciones óptimas.

3.1. Memoria Adaptativa

Probablemente la característica más importante de la Búsqueda Tabú resulta su capacidad de recordar la evolución de la búsqueda, lograda a través del uso de estructuras de datos que almacenan informaciones de la trayectoria de la búsqueda. La *Lista Tabú*, representa una de estas estructuras de datos, utilizada para guardar aquellos movimientos realizados o aquellas soluciones que ya fueron visitadas durante la búsqueda y que por tanto no deben ser tomados durante un determinado tiempo, evitando así la posibilidad de repetir soluciones previamente visitadas, al menos durante un tiempo. En BT estos mecanismos están expresados a través de la memoria de término medio y por la memoria de término largo o memoria de frecuencia.

3.2. Vecindad y Codificación de la solución

Como ya se mencionó la BT es una Metaheurística de búsqueda, lo cual implica que deba definirse una codificación para sus soluciones y una estructura de vecindad. La codificación puede verse como el tránsito de una solución a otra. Para el problema del viajante, por ejemplo, se pudiera definir como solución o codificación de una solución un arreglo de n elementos donde n es el número de ciudades que debe recorrer el viajante y el orden en que aparezcan esas ciudades en el arreglo será el orden en que se deban transitar las ciudades. La vecindad de una determinada solución se puede definir como el conjunto de soluciones que resultan de intercambiar el orden de cualesquiera dos ciudades en y solo en la solución de la cual nos interesa hallar la vecindad, para una solución de n ciudades resultaría una vecindad de $(n-1)n/2$ soluciones vecinas.

3.3. Programación objetivo

Sea S un conjunto de soluciones. Los niveles de aspiración son valores que se establecen y a los que se desea llegar a un determinado criterio. Un umbral de aspiración Z^* es utilizado para obtener soluciones de la siguiente manera, sin pérdida de generalidad se asume que se minimiza cada objetivo, entonces sea $\Delta f(s^{'}) = (\Delta f(s_1^{'}), \Delta f(s_2^{'}), ..., \Delta f(s_r^{'}))$ donde $\Delta f_k(s^{'}) = z_k^* - f_k(s^{'})\ \forall k \in \{1, ..., r\}$.

Entonces se considera que un objetivo se satisface, admitiendo la entrada de s' a S si $\exists \Delta f_k(s^{'}) \geq 0$ o $\forall k \in \{1, ..., r\}\ \Delta f_k(s^{'}) = 0$, de lo contrario se rechaza esa solución. El punto de referencia se actualiza como, $z^* = min\ f_k(s'), \forall k \in \{1, ...r\}, s' \in S$

Para medir la calidad de una solución se propone una función aditiva afv con coeficientes $\lambda_k \geq 0$ que representan la importancia brindada a cada objetivo por el decisor. Se quiere escoger los pesos λ_k de modo que la solución seleccionada se encuentre más cerca del umbral de aspiración. Por tanto cada componente de este vector de pesos obtiene su valor en dependencia del valor del objetivo correspondiente. Se concede más importancia a esos objetivos lo más alejados del punto de referencia posible. La influencia entonces se encuentra dada por la función exponencial e^{-s_k} donde $s_k = (z_k^* - f_k(s'))/z_k^*$, con $z_k^* \neq 0$, $\lambda_k = 2 - e^{-s_k}$, $k \in \{1, ..., r\}$, entonces el valor de una solución queda definida por $afv(s') = \sum_{k=1,...,r} \lambda_k (z_k^* - f_k(s'))$.

3.4. PSET

PSET (Pareto Set) es el componente del algoritmo encargado de construir el conjunto Pareto. Una vez que se ha encontrado una solución PSET verificará si la solución es “admisible” para ser considerada en el conjunto de soluciones. Admisible significa que no es duplicada (que ya se encuentra en PSET) y no dominada. En caso que la solución sea no dominada entonces quizás sea necesario eliminar soluciones antiguas, soluciones que ya no clasifiquen como Pareto por encontrarse dominadas. PSET posee un rango de tolerancia (en 0 por defecto) como un parámetro que pudiera permitir la inclusión de soluciones en PSET que realmente se encuentren dominadas, pero cercanas a una solución no dominada (su diferencia es menor o igual a un parámetro de tolerancia).

PSET será además el encargado de decidir el próximo punto para realizar una iteración de BT. Esto lo hará basándose en el valor de la función

aditiva.

3.5. Regla de parada

La regla de parada, como el nombre sugiere, define el momento o la iteración en la cual el algoritmo debería detenerse. En este caso se ha prefijado un número de iteraciones y una vez alcanzado dicho número el algoritmo debe detenerse. Sin embargo existe otra condición de parada para el algoritmo, que toma en cuenta el hecho de no encontrar nuevas soluciones no dominadas, dado este caso el algoritmo también se detendrá, luego de un cierto número, prefijado, de iteraciones sin mejora.

3.6. Problema de Agrupamiento clásico

Esta sección y la posterior estarán dedicadas a presentar dos algoritmos de Búsqueda Tabú que demuestran como Metaheurísticas de Memoria Adaptativa pueden ser utilizadas para brindar soluciones a diferentes problemas de la vida real. Se empleará el marco general de la Metaheurística BT adaptado a cada problema en cuestión. La diversificación se lleva a cabo apoyado en la memoria de frecuencia mostrando cuantas veces un elemento ha sido incluido en una solución como centro de un grupo.

La primera aplicación que se presentará de Búsqueda Tabú será la resolución del Problema clásico de Agrupamiento. Para ello BT debe apoyarse en funciones, descritas en capítulos anteriores, que sirven de medida para evaluar la calidad de una solución basado en un determinado criterio. El método realizará una búsqueda por objetivos para identificar la nueva solución a la que debe moverse en el espacio solución. Las funcio-

nes a optimizar serán combinaciones de diámetro, intra-clase, inter-clase y promedio intra-clase.

3.7. Problema de Zonificación

La Zonificación es un problema que pertenece al área de estudios urbanos; apareció por primera vez para separar las áreas residenciales de las industriales. La idea principal con este problema, el más popular en urbanismo, es producir una partición de regiones homogéneas de acuerdo a un criterio. En nuestro caso cada variable representa un criterio y cada área geoestática básica posee una colección de valores cada uno representando el valor de alguna variable. Estas variables pudieran ser demográficas por ejemplo personas cuya edad supera los veinte años o personas entre los 7 y los 8 años. Un área geoestática básica (AGB) es la manera en que se designa a una región básica o primitiva que será agrupada. Cualquier AGB consiste en un par (posición, variables) donde la posición marca la ubicación del área en espacio (usualmente dos coordenadas) y variables representa una lista de valores para cada variable en el problema.

3.7.1. Codificación y vecindad

Una solución se codifica como un par (*centros*,*elementos*), primero un arreglo de longitud $n-k$ donde n es el número de objetos y k es el número de grupos a formar y un valor x_i indica que el objeto i se encuentra en el grupo x_i. El otro arreglo *centros*, de longitud k contiene cada centro. La vecindad de una solución x denotada $N(x)$ se obtiene intercambiando cada par de elementos (i, j) donde i es un centro y j es un elemento,

de esta forma tener $s = ((c_1, ..., c_k), (e_1, ..., e_{n-k}))$ como solución implica $((e_1, ..., c_k), (c_1, ..., e_{n-k})) \in N(s)$.

3.7.2. Problema de Zonificación óptima

La Búsqueda Tabú se encuentra embebida en un marco multiobjetivo, así que para resolver el problema se deben tener en cuenta varias funciones. La primera de estas funciones minimiza la distancia intra-clase, o sea, la distancia de cada objeto al centro de su clase o grupo.

$$min \sum_{i=1}^{k} d(c_i, e) \ \ \forall e \in e(c_i)$$

En esta fórmula $d(x, y)$ representa la distancia Euclidiana, c_i un centro y $e(c_i)$ todos los elementos del grupo con centro c_i.

El segundo objetivo considera la homogeneidad de la solución, lo cual significa que elementos que pertenecen a un mismo grupo comparten algunas características, en este caso semejanza en los valores de sus variables. Cuando se particiona bajo el criterio de homogeneidad la idea es encontrar un balance o equilibrio en cada grupo para cada variable, así que la función a ser optimizada es el balance de homogeneidad. Para hallar el balance de un grupo con respecto a alguna variable, se suma el valor de esa variable en cada uno de los integrantes del grupo y luego se divide esa suma entre la cardinalidad del grupo, o sea se halla el promedio, que representará el valor ideal de ese grupo. Luego, por cada miembro del grupo se encuentra la diferencia entre el valor ideal y el valor actual de su variable, la suma de estas diferencias será el valor a minimizar y representa el balance de homogeneidad. Para cada variable que se desee homogeneizar una función de balance de homogeneidad debe ser añadida al modelo de optimización. De este modo, si se quiere homogeneizar

tres variables, el modelo contar con cuatro objetivos, la función intra-clase y una de balance de homogeneidad por cada variable. De igual forma cada una de las funciones que se identifican con cada variable pueden ser agregadas en una sola, para simplificar el modelo.

$$min \sum_{i=1}^{k} |V(c_i) - k * V^*(c_i)|$$

Donde $V(c_i)$ representa la suma de los valores de la variable para cada elemento en el grupo c_i, $V^*(c_i)$ representa el valor ideal para ese grupo y $k = |c_i|$.

Una vez descrito el modelo de optimización es el momento para describir la estrategia de BT para encontrar soluciones no dominadas y construir el frente Pareto.

La estrategia se divide en tres etapas diferentes.

1. Cálculo de la distancia intra-clase.
2. Cálculo de la homogeneidad.
3. Chequeo de dominación de la solución: en caso se ser no dominada se añade utilizando PSET.

Capítulo 4

Resultados Computacionales

Con el objetivo de validar el correcto funcionamiento y de reconocer las capacidades de los algoritmos que se han descrito en este documento, se han realizado pruebas experimentales con diferentes conjuntos de datos. A continuación se presentarán los resultados obtenidos.

4.1. Problema de Agrupamiento clásico

Para el Problema de Agrupamiento clásico se emplearon 3 conjuntos de datos extraídos de casos de la vida real. Cada uno se encuentra representado por un conjunto de vectores.

El conjunto de datos **Iris**, es uno de los más conocidos conjuntos de datos, contiene 3 clases de 50 instancias cada una, donde cada planta se refiere a una planta iris.

El conjunto de datos **Corazón**, describe diagnósticos cardíacos. Cada paciente es clasificado en dos clases: normal y anormal. El conjunto cuenta con 267 pacientes y 44 características continuas por paciente.

El conjunto de datos **Vino**, describe el resultado de análisis químicos

Diámetro	Prom.Intra-clase	Rand	Jaccard
840.10	424.09	0.72	0.45
835.11	424.80	0.72	0.45
835.04	425.19	0.72	0.45
1269.13	345.52	0.34	0.33

Tabla 4.1: Resultados obtenidos para el conjunto de datos Vinos en 15 iteraciones y 3 clases.

Diámetro	Prom.Intra-clase	Rand	Jaccard
3.91	3.58	0.61	0.0
4.83	2.16	0.45	0.0

Tabla 4.2: Resultados obtenidos para el conjunto de datos Iris en 15 iteraciones y 3 clases.

realizados a vinos que crecen en una misma región de Italia pero que provienen de 3 diferentes cultivos. El análisis determina las cantidades de 13 constituyentes encontrados en los 3 vinos. El conjunto contiene 187 objetos. A continuación se presentan los resultados para cada conjunto de datos, optimizando las funciones diámetro y promedio intra-clase.

Note que la distancia inter-clase es negativa porque se quiere hallar su valor máximo y se trata de un modelo de minimización. El algoritmo BT arroja resultados positivos para los conjuntos Vinos y Corazón logrando en todos los casos soluciones con índices de Rand que superan el 0.7 en tan solo 15 iteraciones.

Diámetro	Prom.Intra-clase	Rand	Jaccard
236.46	89.84	0.83	0.83
230.33	134.94	0.84	0.84
201.71	309.40	0.70	0.65
192.15	309.47	0.70	0.65
191.73	310.70	0.70	0.65

Tabla 4.3: Resultados obtenidos para el conjunto de datos Corazón en 15 iteraciones y 3 clases.

Intra-clase	Inter-Clase	Rand	Jaccard
278.85	-13.84	0.05	0.0
278.95	-14.17	0.05	0.0
551.75	-15.51	0.07	0.0
163.54	-13.08	0.46	0.0
131.81	-13.06	0.46	0.0
127.38	-12.98	0.46	0.0
123.15	-12.25	0.48	0.0
124.39	-12.75	0.48	0.0

Tabla 4.4: Resultados obtenidos para el conjunto de datos Iris en 15 iteraciones y 3 clases.

Intra-clase	Inter-Clase	Rand	Jaccard
43508.85	-2100.30	0.34	0.33
43596.89	-2110.29	0.34	0.33
43277.48	-2060.48	0.34	0.33
43380.84	-2080.24	0.34	0.33
42858.65	-2040.53	0.35	0.33
42858.96	-2041.13	0.35	0.33
42858.95	-2040.70	0.35	0.33
42858.62	-2040.49	0.35	0.33
42860.16	-2042.31	0.35	0.33
42858.70	-2040.67	0.35	0.33
42998.07	-2044.76	0.35	0.33
43091.81	-2051.21	0.35	0.33
43619.22	-2131.50	0.35	0.33

42361.82	-1774.38	0.35	0.33
46086.31	-2809.96	0.35	0.33
45025.79	-2807.87	0.35	0.33
44403.85	-2806.15	0.35	0.33
43632.39	-2804.53	0.35	0.33
43721.30	-2804.79	0.35	0.33
43665.74	-2804.74	0.35	0.33
44063.66	-2805.09	0.35	0.33
44126.72	-2805.33	0.35	0.33
27115.83	-553.72	0.74	0.45
30028.61	-561.33	0.68	0.39
59069.23	-2810.55	0.35	0.33

Tabla 4.5: Resultados obtenidos para el conjunto de datos Vinos en 15 iteraciones y 3 clases.

Intra-clase	Inter-Clase	Rand	Jaccard
12295.71	-565.92	0.81	0.81
12175.55	-461.75	0.83	0.83
12705.25	-581.15	0.81	0.81
13489.46	-598.51	0.81	0.81
22377.08	-614.59	0.81	0.81
13355.47	-586.17	0.81	0.81
13088.64	-585.41	0.81	0.81
12985.79	-584.25	0.81	0.81
12621.37	-577.80	0.81	0.81
14398.41	-603.79	0.81	0.81
12129.85	-459.25	0.80	0.80
12126.30	-428.23	0.81	0.81
12013.88	-412.13	0.78	0.78
11965.89	-388.72	0.77	0.77
11839.78	-364.97	0.76	0.76
11846.99	-387.50	0.77	0.77

Tabla 4.6: Resultados obtenidos para el conjunto de datos Corazón en 15 iteraciones y 3 clases.

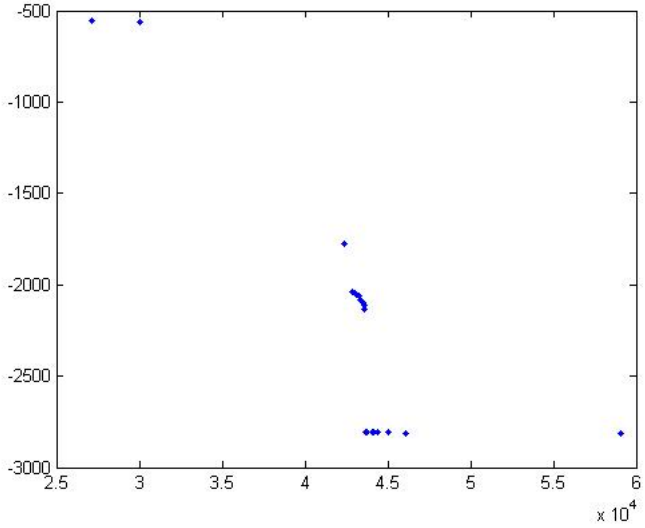

Figura 4.1: Gráfico que ilustra el frente Pareto según los resultados obtenidos para el conjunto Vinos.

4.2. Problema de Zonificación.

Los experimentos para la Zonificación se realizaron utilizando un problema de la vida real. El valle de Toluca en México, requiere ser agrupado en regiones homogéneas y compactas. En estos datos se tienen 2 matrices de disimilitud, una de distancias y la otra de variables demográficas. Cada vector representa una AGB y cada componente una coordenada en el espacio o una variable demográfica según corresponda la matriz de disimilitud. Los resultados obtenidos son los siguientes:

Como se puede apreciar los resultados son extremadamente positivos, teniendo en cuenta que se lograron grupos bastante compactos y que la homogeneidad se toma para 3 variables. En la próxima sección se verá como se comparan estos resultados contra resultados de otros algoritmos.

Hom	Comp
11847.379	36.382
11004.949	46.265
10636.740	46.267
10466.369	46.488
10451.237	47.993
10388.293	49.566
10435.767	49.384
10429.598	49.389
10386.362	49.657
10427.542	49.433
10403.743	49.451
10384.556	50.391
11870.684	32.710

Tabla 4.7: Resultados para la zonificación en 15 iteraciones y $k = 5$.

4.3. Comparaciones

Esta sección estará dedicada a comparar los resultados de las Metaheurísticas de Memoria Adaptativa que aquí se han presentado contra resultados brindados por otros algoritmos.

4.3.1. Problema de Zonificación

A continuación se compararán los resultados de la Búsqueda Tabú para el problema de Zonificación que se describió en esta tesis y que será referida como BT, con los resultados del algoritmo presentado en

[1] que es un VNS (Variable Neighborhood Search). En ambos casos se realizaron 15 iteraciones para $k = 5$, esta es la comparación:

VNS		BT	
Hom	Comp	Hom	Comp
55262	3256.4	**11847.379**	**36.382**
37111	4419.6	**11004.949**	**46.265**
73647	2162.4	**10636.740**	**46.267**
94983	1217.2	**10466.369**	**46.488**
		10451.237	**47.993**
		10388.293	**49.566**
		10435.767	**49.384**
		10429.598	**49.389**
		10386.362	**49.657**
		10427.542	**49.433**
		10403.743	**49.451**
		10384.556	**50.391**
		11870.684	**32.710**

Tabla 4.8: Comparación entre VNS y BT.

Como puede observarse BT obtiene resultados mucho más favorables, teniendo en cuenta que VNS está basando su criterio de homogeneidad en 1 variable y BT en 3 variables al mismo tiempo.

4.3.2. Problema de Agrupamiento clásico

Para comparar los resultados en el Problema de Agrupamiento clásico se tomó en cuenta la media y la desviación estándar del conjunto de solu-

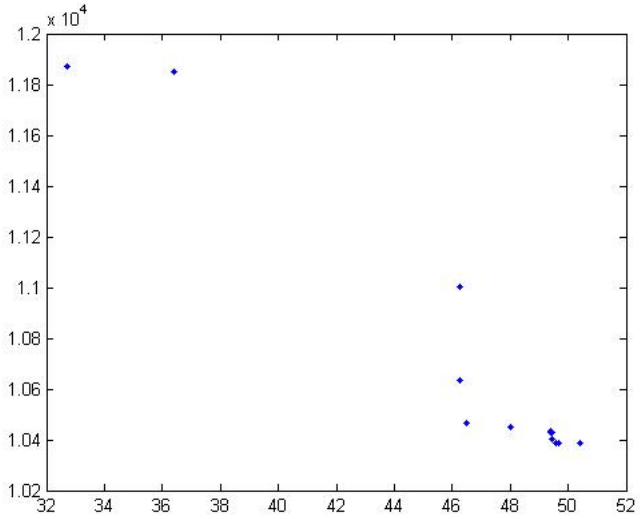

Figura 4.2: Gráfico que ilustra el frente Pareto según los resultados ofrecidos por el algoritmo descrito para el problema de Zonificación.

ciones brindadas por cada algoritmo con respecto a cada función objetivo. En este caso BT se refiere a la Búsqueda Tabú descrita en este trabajo mientras que BTII hace alusión a la Búsqueda Tabú presentada en [2]. En las tablas se indica también la cantidad de soluciones ofrecidas por cada algoritmo.

Como se puede apreciar BT logra una mejora en relación a la media y la desviación obtenida por BTII para el conjunto de datos Corazón optimi-

	BTII			BT		
	Media	Desv.	Sol.	Media	Desv.	Sol.
Diámetro	531.05	69.54	8	944.84	187.23	4
Prom. Intra-clase	423.59	21.22		**404.9**	34.28	

Tabla 4.9: Comparación entre BTII y BT para el conjunto Vinos.

	BTII			BT		
	Media	Desv.	Sol.	Media	Desv.	Sol.
Diámetro	217.51	157.09	3	**210.47**	**19.14**	5
Prom. Intra-clase	157.09	32.27		230.87	97.78	

Tabla 4.10: Comparación entre BTII y BT para el conjunto Corazón.

zando la función diámetro.

Conclusiones y trabajo futuro

El Agrupamiento es un problema de la computación que encuentra aplicaciones en los más diversos campos. Pertenece a la familia de problemas NP-completo, lo cual hace indispensable el uso de métodos heurísticos para hallar soluciones en un tiempo factible.

En este trabajo se presentaron dos algoritmos de Búsqueda Tabú que fueron adaptados a dos problemas, el Agrupamiento clásico y la Zonificación. Ambos algoritmos fueron desarrollados en un marco multiobjetivo así que se logró, exitosamente conformar algoritmos de Memoria Adaptativa que abordaran la optimización multiobjetivo y de manera particular el problema de Agrupamiento.

La efectividad de cada uno de los algoritmos propuestos fue comprobada mediante un proceso de experimentación con datos de la vida real y los resultados fueron satisfactorios. Para el problema de Zonificación las comparaciones realizadas con otros algoritmos arrojaron resultados superiores para la Búsqueda Tabú que logró de las soluciones obtenidas una media de 46.41 para los valores de la función de compacidad o distancia intra-clase, mientras que VNS obtuvo una media de 2831.4 para la misma función. En cuanto a la homogeneidad el algoritmo presentado en esta tesis consigue una media de 10703.00 lo cual representa una mejora significativa en relación al VNS con el que se comparó que logra una media

de homogeneidad de 62250.75. Por otra parte, para el Problema de Agrupamiento clásico que se comparó con otra propuesta de Búsqueda Tabú los resultados fueron semejantes. Se alcanzó una mejora en las pruebas realizadas para el conjunto de datos Corazón de modo que para la función diámetro se obtuvo una media de 210.47 y una desviación de 19.14 sobre las soluciones obtenidas y para la función promedio intra-clase una media de 230.87 y una desviación de 97.78 con el algoritmo presentado en este documento, en tanto la Búsqueda Tabú descrita en [2] logra medias de 217.51, 157.09 para el diámetro y el promedio intra-clase respectivamente y desviaciones de 157.09, 32.27 para las mismas funciones.

Se puede concluir que los objetivos planteados al comienzo de la presente tesis fueron satisfactoriamente cumplidos. Los resultados alcanzados son exitosos y pueden servir como base para futuras investigaciones y proyectos en el campo de la optimización.

Trabajo Futuro

Como continuación de este trabajo se recomienda:

- Diseñar e implementar otras Metaheurísticas de Memoria Adaptativa como la Búsqueda Dispersa que resuelvan el Problema de Agrupamiento.
- Verificar la calidad de los algoritmos empleando otras medidas de evaluación.
- Mejorar el componente PSET de modo que sea más eficiente y consuma menos recursos computacionales.

Bibliografía

[1] Bernábe Loranca B., Coello Coello A.C., Osorio Lama M., *A multi-objective approach for the heuristic optimization of compactness and homogeneity in the optimal zoning*

[2] Beausoleil P. Ricardo, *Multiobjective clustering using Tabu Search*, Institute of Cybernetics, Mathematics and Physics.

[3] Beausoleil R. P.,(2001), *Multiple Criteria Scatter Search*, Proceedings MIC'2001-4th Metaheuristics International Conference, Porto, Portugal, July 16-20.

[4] Caballero R., Laguna M., Martí R., Molina J., *Multiobjective Clustering with Metaheuristics Optimization Tecnology*.

[5] Talbi El-Ghazali, (2009), *Metaheuristics: from design to implementation*, New Jersey: Wiley and Sons.

[6] Jain A. K. , Dubes R. C., (1988), *Algorithms for Clustering Data*, Prentice-Hall Inc., Upper Saddle River, NJ.

[7] Reeves C. R. (Ed.), (1993), *Modern Heuristics Techniques for Combinatorial Problems*, Blacwell, London.

[8] Xu Rui, Wunsch C. Donald, (2009), *Clustering*, Wiley Inc., Hoboken, New Jersey.

[9] Glover Fred, (1989), *Tabu Search*, ORSA Journal on Computing.

[10] Zhou Aimin, Qu Bo-Yang, Li Hui, Zhao Shi-Zheng, Suganthan Nagatarnam, Zhang Qingfu, (2011), *Multiobjective evolutionary algorithms: A survey of the state of the art*, ElSevier.

[11] Baeza-Yates Ricardo, Frakes B. Williams, *Information Retrieval: Data Structures and Algorithms*.

[12] Fung Glenn, (2001), *A Comprehensive Overview of Basic Clustering Algorithms*.

[13] Fasulo Daniel, (1999), *An analysis of recent work on clustering algorithms*.

[14] Abbas Abu Osama, (2007), *Comparison between data clustering algorithms*, Computer Science Department, Yarmouk University, Jordan.

[15] Branke Jurgen, Deb Kalyanmoy, Miettinen Kaisa, Slowinski Roman, (2008) *Multiobjective Optimization: Interactive and Evolutionary Approaches*, Springer-Verlag.

[16] Barichard Vincent, Ehrgott Matthias, Gandibleux Xavier, T'Kindt Vincent, (2009), *Multiobjective Programming and Goal Programming: Theorical Results and Practical Applications*, Springer.

[17] Casillas A.,González de Lena M. T., Martínez R., *Document Clustering into an unknown number of clusters using a Genetic Algorithm*.

[18] Han Eui-Hong, Karypis George, Kumar Vipin, Mobasher Bamshad, *Clustering In A High-Dimensional Space Using Hypergraph Models*.

[19] Grajales Tevni, *Cluster Analysis*.

[20] Busse M. Ludwig, Orbanz Peter, Buhmann M. Joachim, *Cluster Analysis of Heterogeneous Rank Data*, Institute of Computational Science, ETH Zurich, 8092 Zurich, Switzerland.

[21] A. K. Jain, M. N. Murty and P. J. Flynn, (1999), *Data clustering: A review*, ACM Computing Surveys.

[22] Glover Fred, Laguna Manuel, (1993), *Tabu Search*, Modern Heuristic Techniques for combinatorial problems, Blackwell Scientific Publications, Oxford.

[23] Glover Fred, Laguna Manuel, Martí Rafael, *Principles of Tabu Search*.

[24] Glover Fred, *Tabu Search and adaptive memory programming, advances, applications and challenges*.