



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA
FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

TESIS DE LICENCIATURA:

**IMPLEMENTACIÓN DE UN MODELO
MATEMÁTICO PARA RESOLVER
AGRUPAMIENTO SOBRE
HOMOGENEIDAD PARA OBJETOS DE
TIPO GEOGRÁFICO**

Una Tesis por Marco Antonio Rodríguez Flores para obtener el
grado de Licenciado

Asesorado por:
Dra. María Beatriz Bernábe Loranca

2012

Agradecimientos

A Dios Por guiarme toda vida hasta este momento fundamental, por bendirme con los mejores padres que uno puede desear, por darme salud, lo necesario y más. Por poner a toda la gente adecuada en mi vida para convertirme en lo que soy ahora: mis padres, mis hermanos, todos mis amigos y amigas, mi asesora y mis profesores, todos los que han sido una fuerte influencia en mi vida. Y por ser mi fortaleza en los malos momentos y cuando estoy solo.

A mis padres Por procurar darme la mejor educación posible, por apoyar el rumbo que decidí tomar; económicamente y emocionalmente. Por su sabio consejo desde que era pequeño y por estar conmigo en las buenas y en las malas.

A mi hermano y a mi hermana A mi hermano porque crecimos juntos y hemos compartido mucho y a mi hermana que aunque no crecimos juntos, ella y su familia nos apoyan siempre que pueden y en especial cuando más se necesita y sin duda ustedes han influenciado mi forma de ser.

A Adriana Más que una amiga, una hermana, gracias por todo tu apoyo y todos tus consejos en especial este último par de años, me has cambiado mucho para bien, te he aprendido tanto y te admiro por tu fortaleza y tu buena energía a pesar de todo.

A Alix y Tona Gracias por ser mis grandes amigos por ya casi una década, por todas las alegrías que hemos compartido y por estar presentes en mi graduación, han llenado mi vida de mucha felicidad, de sabios consejos y me han cargado de energía.

A mi asesora A la Doctora Beatríz por todas sus enseñanzas y el grandísimo apoyo para poder concluir esta tesis y por ser una gran mentora y amiga. Gracias por guiarme por el rumbo de la investigación, por despertar en mí ese interés.

A mis amigos tenistas Por todas las risas este año de entrenamiento y por su amistad y por todo lo que me han enseñado tanto del deporte blanco como de la vida, cada uno de ustedes aporta algo en mi vida.

To my friends around the world A very special group of people from all over the world; since i met all of you last year my life has been changed so much. I've learned so much from you, from your countries, from your way of seeing life. Thanks to all of you now i'm a more joyful person and i really appreciate your genuine friendship.

To Maria A. Thank you, because since i met you, you've filled my heart with so much happiness. Thanks for all the things and moments we've shared so far, especially for the laughs. I look forward for what's ahead for us...

Resumen

El agrupamiento de unidades geográficas, zonas o áreas ha sido utilizado para resolver problemas relacionados con el Diseño Territorial ya que su propia descripción demanda el procesamiento de datos espaciales bajo esquemas de clasificación con requerimientos topológicos de las zonas.

Este problema ha sido resuelto por métodos de particionamiento clásicos como PAM, el cual tiene un tiempo de respuesta aceptable para instancias pequeñas pero al ser un método de búsqueda exhaustiva; instancias muy grandes del problema son intratables. Por esta razón, cuando el tamaño del problema aumenta, el uso de métodos metaheurísticos se hace necesario. Los algoritmos evolutivos, híbridos que combinan estas técnicas y técnicas de búsqueda en espacios de soluciones muy grandes prometen soluciones en cuanto a optimalidad a un costo de computo razonable.

Un caso en particular de este tipo de problemas es el agrupamiento de AGEBS (Áreas Geo-estadísticas Básicas), que son unidades geográficas definidas por el INEGI (Instituto Nacional de Estadística y Geografía) en México para dividir el territorio nacional. Se han realizado muchos trabajos relacionados con la formación de grupos compactos de AGEBS, pero la inclusión de restricciones adicionales ha sido poco tratada, de este modo, se ha identificado que una extensión necesaria para agrupamiento compacto es construir grupos bajo cierto criterio de homogeneidad y/o balanceo en el número de objetos que componen los grupos.

Por lo tanto en esta tesis, bajo el apoyo de un modelo matemático, el objetivo principal es el desarrollo e implementación de un algoritmo de particionamiento para unidades geográficas que permita encontrar aproximadas soluciones homogéneas (conforme al número de elementos en cada grupo) respetando las restricciones de compacidad usuales. La metaheurística búsqueda

tabú (TS por sus siglas en inglés) ha sido elegida para el manejo del costo computacional y para alcanzar soluciones satisfactorias para instancias grandes.

Introducción

El agrupamiento geográfico pertenece a la clase de problemas de Diseño Territorial (DT) debido a que DT exige para su resolución, como principio fundamental, la creación de grupos de zonas espacialmente compactas, contiguas y/o conexas, esto implica incorporar un método de agrupamiento para datos geográficos. Las aplicaciones de DT incluyen distribución política, comercio, censos, clasificación para muestreo, etc. El área de investigación de este problema es amplio y da lugar a revisar una gran cantidad de trabajos al respecto. En el capítulo 1 se han recogido de la literatura disponible algunos trabajos importantes y relevantes para nuestro caso de estudio.

Primero desglosemos este problema: existen dos grandes categorías de técnicas de agrupamiento: Las jerárquicas y las de particionamiento, en la primera clase se desconoce el número de grupos a formar o bien se desea encontrar el agrupamiento “natural” de los datos, siendo el caso contrario para las del segundo tipo; en nuestro caso se conoce el número de grupos a formar de antemano (el usuario es quien decide cuántos grupos formar) por lo tanto se utiliza una técnica de particionamiento.

Un aspecto importante del agrupamiento es la complejidad ya que está clasificado como un problema NP-difícil, y muchos han sido los esfuerzos para resolver este problema. Las técnicas de aproximación más útiles son las metaheurísticas, algoritmos evolutivos y algoritmos de búsquedas en espacios de solución muy grandes, entre otros. Todas estas técnicas buscan encontrar buenas soluciones o bien la óptima en un tiempo razonable de cómputo. Otro elemento importante de este problema es el tipo de solución, ya sea compacta (minimización de distancias) o en base a cierta homogeneidad que puede ser intragrupo, es decir, que los elementos de determinado grupo comparten una o más características en específico que los distinguen como a un

grupo de objetos similares, mientras que la homogeneidad intergrupo busca que los grupos formados sean homogéneos entre sí, como por ejemplo, que tengan la misma cantidad de elementos, la misma área, la misma población, etc. En este último tipo de homogeneidad se centra este trabajo, donde se ha propuesto un algoritmo de agrupamiento sobre medoides para conseguir grupos compactos y homogéneos en la cardinalidad de los grupos y dado su alto costo computacional el uso de metaheurísticas se hace necesario, siendo búsqueda tabú el método de aproximación utilizado.

Los objetivos específicos son:

- Elegir un modelo matemático adecuado para el problema bajo estudio y realizar modificaciones si es necesario.
- Generar prototipos de algoritmos que ayuden a entender mejor el problema en cuestión y todo lo que implica.
- Crear un prototipo basado en PAM que utilice la función objetivo del modelo elegido.
- Realizar pruebas con el prototipo de PAM para corregir/modificar el modelo o bien el algoritmo.
- Realizar un prototipo de búsqueda tabú básica con base en el modelo revisado.
- Experimentar con los diversos elementos y estrategias que ofrece la búsqueda tabú para elegir los que den mejor resultado en nuestro problema en particular.
- Realizar pruebas con las diversas versiones de búsqueda tabú generadas para hacer las correcciones o modificaciones pertinentes hasta llegar a la versión final.
- Realizar pruebas comparativas con el prototipo de PAM y las versiones de búsqueda tabú que se comporten mejor.
- Graficar los resultados obtenidos.
- Generar mapas de algunas de las instancias resueltas con los algoritmos implementados.

- Hacer pruebas preliminares con el algoritmo final de búsqueda tabú sobre el mapa más grande que se tiene disponible (El valle de México con 5087 Unidades Geográficas).

Justificación: Como se expone en el capítulo 1, sección 1, en los antecedentes del problema, la homogeneidad inter clase con respecto a la cantidad de elementos en cada grupo formado en el problema del agrupamiento geográfico ha sido poco tratado y se continúan haciendo esfuerzos, en particular sobre los mapas de AGEBS para lograr la compacidad y a la vez una homogeneidad estricta (una tolerancia máxima de 10% de diferencia). En dicha sección se exponen algunos trabajos que han tratado la homogeneidad y en particular un algoritmo híbrido de recocido simulado con VNS que no logra muy buenos resultados en este problema. Por lo tanto en esta tesis se propone un nuevo algoritmo de búsqueda tabú basado en un modelo de optimización con el fin de hacer un aporte a esta línea de investigación que requiere atención.

Alcance: En el Capítulo 1, sección 2, se expone las aplicaciones de los problemas de Diseño Territorial, categoría a la cual pertenece nuestro caso de estudio. En particular la homogeneidad interclase con respecto a la cardinalidad de los grupos formados aporta beneficios a la distribución política con fines electorales para una distribución homogénea de las casillas electorales, colocación de escuelas, plantas eléctricas, censos, etc. O bien para el comercio y la industria para explotar la colocación de los puntos de venta y llegar a más clientes de manera homogénea.

Limitantes: Se podría considerar como limitante la cantidad de mapas de AGEBS que se tienen disponibles para poner a prueba el algoritmo desarrollado en esta tesis; solamente se cuenta con los mapas de Toluca y del valle de México, con 469 y 5087 AGEBS respectivamente. El algoritmo recibe como entrada la matriz de disimilitud generada a partir de cada mapa.

De lo anterior se conforma el capítulo 1, aspectos teóricos, se plantean los antecedentes del problema, el estado del arte en Diseño Territorial, Particionamiento, Metaheurísticas, y Agrupamiento homogéneo. El capítulo 2 abarca la descripción del problema y nuestra propuesta de modelo matemático para resolverlo. En el siguiente capítulo se muestra el diseño del algoritmo asociado al modelo matemático y su evolución a lo largo del desarrollo de esta tesis,

empezando por las primeras propuestas hasta la última versión que incorpora la metaheurística Búsqueda Tabú. En el capítulo 4 se presenta y justifica el uso de un método metaheurístico para tratar el problema, su descripción y cada uno de los elementos y estrategias que conforman a la metaheurística elegida que es la Búsqueda Tabú. Finalmente los resultados y conclusiones son presentados en el capítulo 5.

Índice general

Resumen	III
Introducción	V
Lista de Figuras	IX
Lista de Tablas	XI
1. ASPECTOS TEÓRICOS	1
1.1. Antecedentes	1
1.2. Estado del Arte	5
1.2.1. Diseño Territorial	5
1.2.2. Particionamiento	10
1.2.3. Metaheurísticas	16
2. EL PROBLEMA	25
2.1. Definición del Problema	25
2.2. Modelo matemático implementado	27
3. DISEÑO DE ALGORITMO	31
3.1. Algoritmo Versión 1	31
3.2. Algoritmo Versión 2	33
3.3. Algoritmo Versión 3	34
3.4. Algoritmo de Particionamiento de Grafos	39
3.5. Versión 4. Modelo Matemático	46
3.5.1. Modelo Matemático Primera Propuesta	46
3.6. Modelo Matemático Versión Final	49
4. METAHEURÍSTICA BÚSQUEDA TABÚ	50
4.1. Estructuras de Datos	50

<i>ÍNDICE GENERAL</i>	X
4.2. Algoritmo de Búsqueda Tabú	52
4.3. Elementos y Estrategias Tabú	55
4.4. Modificación Swap Method	59
5. Resultados y Conclusiones	60
5.1. Pruebas de las Implementaciones	60
5.1.1. Pruebas preliminares para una instancia grande	65
5.1.2. Conclusiones de las Pruebas	66
5.2. Trabajo a Futuro	68
5.3. Conclusiones Finales	69
Apéndices	71
A. Póster en NaBIC 2012	72
Bibliografía	74

Índice de figuras

1.1. Grafo de adyacencia preciso generado con Triangulación de Delaunay	4
1.2. AGEBS de formas extrañas dificultan la obtención de un grafo de adyacencia preciso	5
3.1. Versión 1. Homogeneidad y Compacidad para 5 grupos	33
3.2. Versión 2. PAM modificado para 5 grupos	34
3.3. PAM modificado con versión 3 para 5 grupos.	37
3.4. PAM modificado con versión 3 para 15 grupos, AGEBS dispuestos.	38
3.5. PAM modificado con versión 3 para 20 grupos, AGEBS dispuestos.	38
3.6. Técnica de Matching - los vértices se emparejan con otros para formar uno nuevo.	40
3.7. Esquema de particionamiento multinivel	41
3.8. Solución Inicial con Particionamiento de Grafos	43
3.9. Grafo Uniforme generado con Triangulación de Delaunay. (Romero, 2002)	45
3.10. Grafo Irregular generado con Triangulación de Delaunay	45
3.11. Estructura de Cubetas. (Benlic, 2010)	46
4.1. Estructura de Centroides	51
4.2. Estructura de AGEBS	51
4.3. Matriz de Agrupamiento	52
5.1. Comparación de costo de compacidad	63
5.2. Comparación de la penalidad de homogeneidad	63
5.3. Comparación del tiempo de cómputo	64

5.4. Comportamiento de Búsqueda Tabú SM 68

Índice de cuadros

1.1. Criterios para el diseño de Distritos Electorales en Artículos Selectos.	7
1.2. Clasificación de Metaheurísticas.	24
5.1. Corridas de PAM Extendido y Búsqueda Tabú para la ciudad de Toluca.	61
5.2. Corridas de Búsqueda Tabú SM para el mapa del valle de México.	65

Capítulo 1

ASPECTOS TEÓRICOS

1.1. Antecedentes

El problema del agrupamiento geográfico consiste en agrupar unidades geográficas pequeñas en clusters geográficos más grandes llamados territorios, bajo algún criterio, como la minimización de distancias para formar grupos compactos y contiguos. Esto se logra a través de una función objetivo que considera la distancia total de las unidades geográficas hacia el representante del grupo en el que fueron asignadas.

La definición anterior muy simplificada del problema hace ver al problema como algo trivial y sencillo, sin embargo, la solución del problema es de carácter combinatorio (NP-difícil) y exige que su solución sea generada a partir de métodos no exactos de optimización como las metaheurísticas [10].

Por esta razón es un problema en el que se ha trabajado desde hace mucho tiempo y en diferentes partes del mundo. Los primeros trabajos sobre agrupamiento geográfico se remontan a los años 70 con Garfinkel y Nemhauser que se han enfocado en la distribución política [21]. En México, ha sido Romero en el año 2002 quien inició esta línea de investigación [52].

A continuación se presentan una serie de trabajos de los cuales esta tesis extiende el problema hacia restricciones de homogeneidad. Los trabajos mencionados a continuación tienen el fin de agrupar unidades geográficas conocidas como AGEBS (Áreas Geoestadísticas Básicas) y abordan el pro-

blema con técnicas de optimización combinatoria. Un AGEB está definido por el INEGI de la siguiente manera:

Es la extensión territorial que corresponde a la subdivisión de las Áreas Geoestadísticas Municipales. Constituye la unidad básica del Marco Geoestadístico Nacional y, dependiendo de sus características, se clasifican en dos tipos; Áreas Geoestadísticas Básicas Urbanas y Áreas Geoestadísticas Básicas Rurales .

[31]

El Marco Geoestadístico está definido como:

El sistema único, de carácter nacional, diseñado por el INEGI para referir geográficamente la información estadística de los censos y las encuestas. Es la división del país en Áreas Geoestadísticas con tres niveles de desagregación: Estatal (AGEE), Municipal (AGEM) y Básica (AGEB).

[31]

Los primeros intentos de particionamiento para los AGEBS, se apoyaron de las herramientas de clasificación que ofrecen algunos paquetes estadísticos, sin embargo, los resultados obtenidos de agrupamiento distan mucho de los principios básicos de agregación territorial [8, 9]. Estas agrupaciones cumplen con criterios predeterminados por el paquete estadístico, sin embargo, en los grupos resultantes se observó que las unidades geográficas están dispersas (no contigüidad en los grupos), violando la restricción de compacidad. Además, las soluciones que ofrecen son sólo óptimos locales [9].

En [10] Bernábe y otros trabajan en el problema de regionalización de carácter censal sobre AGEBS, de tal manera que se implementa un algoritmo de recocido simulado de particionamiento teniendo como principal objetivo la minimización de distancias entre objetos y centroides.

La metaheurística recocido simulado (RS) muestra un rendimiento muy superior en cuanto a tiempo de cómputo con respecto a la técnica clásica PAM (Partitioning Around Medoids) [54]. Por otro lado, RS logra el objetivo de escapar de los óptimos locales; uno de los mayores inconvenientes de trabajar

con paquetes estadísticos y se logra el cumplimiento de las restricciones de compacidad y contigüidad.

En los trabajos [6, 7] Bernábe y otros resuelven el problema del agrupamiento geográfico con Búsqueda por Entorno Variable(VNS) y Recocido Simulado (RS) respectivamente. Donde el modelo matemático propuesto utiliza una función objetivo de minimización de distancias con el fin de encontrar una solución compacta y contigua óptima. La compacidad y contigüidad son requisitos obligados para la mayoría de los problemas de diseño territorial, sin embargo varios de estos problemas tienen otras especificaciones que obligan a encontrar soluciones óptimas para varios criterios u objetivos, por ejemplo, para la distribución electoral y los censos surge la necesidad de formar grupos homogéneos en cuanto al número de unidades geográficas, extensión territorial o concentración de la población aunado a la compacidad y contigüidad de las zonas formadas, es decir, se debe de encontrar una solución óptima para homogeneidad y compacidad. En [5] Bernábe, Rodríguez y otros proponen y prueban varias técnicas o algoritmos para lograr el tipo de homogeneidad antes mencionado, sin embargo los resultados obtenidos, en la mayoría de los casos presentan dispersión en los grupos (soluciones infactibles para compacidad y contigüidad) además que el tiempo de cómputo parece incrementarse considerablemente a partir de 15 grupos para el algoritmo recursivo que extiende a PAM y que construye soluciones homogéneas propuesto por los autores .

En [53] Romero, Burguete y otros abordan la parcelación de territorio utilizando un método de optimización combinatoria como apoyo para generar un marco muestral. El problema consiste en determinar agrupamientos de manzanas para áreas urbanas y de localidades para zonas rurales, sujetos a características operativas y geométricas que faciliten los procesos de selección de muestras y recorridos en campo. Como primera instancia se genera un grafo de adyacencia entre manzanas utilizando la Triangulación de Delaunay, en donde el centro de cada manzana son los nodos y las aristas representan la contigüidad de las manzanas. Las manzanas poseen una forma regular (cuadriláteros) y todas son contiguas, es decir, no hay manzanas dispersas, por lo tanto el grafo generado es una representación fidedigna de la realidad como se puede ver en la figura 1.1.

Con base en este grafo, el problema es tratado como uno de particionamiento de grafos y la función objetivo busca la homogeneidad en cuanto al

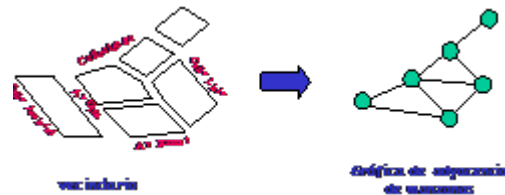


Figura 1.1: Grafo de adyacencia preciso generado con Triangulación de Delaunay

número de viviendas por sobre la compacidad. El particionamiento de grafos es un problema NP-completo, por lo cual se emplea la metaheurística de recocido simulado para obtener soluciones factibles en un tiempo razonable de cómputo. Básicamente un problema NP-difícil (como el agrupamiento geográfico) es como uno NP-completo pero no necesariamente en NP, mientras que un problema NP-completo (como el particionamiento de grafos) es un problema maximalmente difícil en NP.

En conclusión, el modelo logra su objetivo, es flexible, robusto y la implementación lograda constituye un aporte para el INEGI, sin embargo para nuestro caso de estudio este esquema de particionamiento no es eficiente, principalmente porque los mapas de AGEBS con los que se trabajan son irregulares: la forma de los AGEBS varía mucho y algunos están dispersos (no se encuentran junto a los demás) por lo tanto el grafo de adyacencias generado con Triangulación de Delaunay no es una representación exacta de las adyacencias reales entre los AGEBS en el mapa, como se puede ver en la figura 1.2, existen AGEBS con forma de “H”, otros semicirculares y muchos otros tienen otras formas no geométricas que hacen imposible una representación exacta por medio de grafos de adyacencia obtenidos por métodos de la computación geométrica como la Triangulación de Delaunay.

En [11] Bernábe, Pinto, Olivares y otros, desarrollaron una metaheurística híbrida compuesta por Búsqueda por Entorno Variable (VNS) y Recocido Simulado (RS) para resolver homogeneidad y compacidad en el problema del agrupamiento geográfico, sin embargo, la restricción de balanceo de los grupos (homogeneidad) es moderada ya que logran cumplir con el balance de los grupos sólo para cotas muy holgadas y pocos grupos; una de los mejores resultados es para 11 grupos con cota inferior igual a 30 elementos y cota



Figura 1.2: AGEBS de formas extrañas dificultan la obtención de un grafo de adyacencia preciso

superior igual a 55 elementos. Para una cantidad mayor de grupos es necesario que las cotas sean menos estrictas.

Esta tesis sigue la misma línea de investigación que los trabajos presentados, proponiendo un modelo matemático de optimización multiobjetivo para encontrar soluciones óptimas para 2 criterios: la compacidad y la homogeneidad intergrupos en cuanto al número de elementos (AGEBS). Así mismo se implementa una técnica metaheurística, la búsqueda tabú, adaptada el modelo propuesto para lidiar con la complejidad del problema.

1.2. Estado del Arte

La línea de investigación de esta tesis abarca diversos temas: Diseño Territorial, Particionamiento y Metaheurísticas; en especial la Búsqueda Tabú. A continuación se expone a grandes rasgos el panorama de la investigación alrededor de estos temas.

1.2.1. Diseño Territorial

El diseño territorial puede ser visto como el problema de agrupar pequeñas unidades geográficas llamadas áreas básicas en grupos (clusters) geográficos más grandes llamados territorios de tal manera que estos últimos son aceptables de acuerdo a cierto criterio de planificación relevante. Dependiendo del contexto, estos criterios pueden estar motivados ya sea económicamen-

te (promedio de ventas potenciales, carga de trabajo o número de clientes) o tener un fondo demográfico (número de habitantes, población votante). Además restricciones espaciales (contigüidad, compacidad) son demandadas con regularidad [46].

Algunas de las aplicaciones del problema de diseño territorial son: el diseño de distritos electorales, territorios escolares, albergues, recolección de basura, ventas y servicios, etc. Sin embargo los más importantes son el diseño de territorios electorales y las ventas [46].

Para el caso del diseño de distritos electorales se pueden destacar tres criterios de agrupamiento: demográficos, geográficos y políticos.

El criterio demográfico implica la homogeneidad en cuanto al número de habitantes y recientemente considerar a las minorías para que tengan las mismas facilidades de participar en los procesos electorales que el resto de la población.

En los criterios geográficos se destacan la compacidad, que implica la formación de distritos de forma “circular” y sin distorsiones, sin embargo, no hay una definición concreta de compacidad como lo menciona Nickel en [46].

Por último en los criterios políticos se implica el uso de datos con fines políticos para lograr una homogeneidad socio-económica a través de los distritos para lograr una mejor representación de habitantes que comparten puntos de vista y/o preocupaciones.

En [18] Duque y otros proponen una nueva metodología para el diseño territorial homogéneo con base en características estadísticas de las unidades geográficas y sus relaciones. La homogeneidad que se busca es intra grupos, es decir, que los elementos de cada grupo sean similares entre sí, con respecto a alguna característica socio-económica, como por ejemplo obtener zonas geográficas de acuerdo a clases sociales, género o edad. Cabe destacar en este artículo la utilización de la Triangulación de Delaunay para generar un grafo de las unidades geográficas y así tener a disposición en todo momento las unidades contiguas, por lo tanto el problema es tratado como particionamiento de grafos. Como se mencionó anteriormente, en una de las propuestas de esta tesis se intentó realizar lo mismo, sin embargo no resultó factible debido a

la forma irregular de los AGEBS y debido a que en el mapa del valle de Toluca no todos los AGEBS son contiguos y como resultado la Triangulación de Delaunay genera un grafo poco preciso de dicho mapa.

Los resultados en [18] muestran que la metodología propuesta es suficientemente buena y competitiva por lo cual es una opción de alta calidad para implementar en software comercial, sin embargo uno de sus defectos es que aunque la contigüidad de las unidades geográficas se cumple, la forma que pueden adoptar las zonas es indeterminada por la manera en que se realiza el particionamiento en grafos, a diferencia de algoritmos como PAM y k-medias que tienden a formar zonas de forma circular.

El cuadro 1.1 proporciona información de algunos artículos sobre diseño de distritos electorales, indicando qué criterios de los antes mencionados se utilizan en cada uno.

Referencia	País	Contigüidad	Fronteras	Compacidad	Datos Politicos
Hess et al. [28]	EUA	+	-	+	-
Garfinkel and Nemh. [22]	EUA	+	-	+	-
Helbig et al. [26]	EUA	+	+	+	-
Bodin [13]	EUA	+	-	-	-
Bourjolly et al. [14]	Canadá	+	-	+	+
Nygreen [47]	Gales	+	+	+	-
Hojati [29]	Canadá	-	-	+	-
George et al. [23]	Nueva Zelanda	+	+	+	-
Ricca and Simeone [50]	Italia	+	+	+	-
Mehrotra et al. [43]	EUA	+	+	+	-
Cirincione et al. [17]	EUA	+	+	+	+
Bozkaya et al. [15]	Canadá	+	+	+	+
Forman and Yue [20]	EUA	+	+	+	-

Cuadro 1.1: Criterios para el diseño de Distritos Electorales en Artículos Selectos.

En estos artículos todos los autores toman en cuenta la equidad poblacional, por tal motivo no aparece en la tabla como criterio. Esta homogeneidad se maneja como una restricción en el modelo matemático en estos trabajos.

En [28] los autores desarrollaron una medida de compacidad y una heurística de “localización de almacenes” para dibujar distritos constitucionales

políticos imparciales. La heurística asegura la formación de distritos compactos y contiguos de población equitativa.

Como es usual, el criterio de minimización es la suma de los cuadrados de las distancias de cada población al centro del distrito.

En [22] se persigue el mismo objetivo pero con una técnica diferente: generan primero todos los distritos posibles factibles con base en compacidad, contigüidad y población, una segunda fase encuentra el conjunto de M distritos factibles que cumple con las restricciones del problema. Se reporta que para el hardware disponible de esa era (1970) lograban encontrar soluciones en 10 minutos o poco más para estados con 40 condados en una IBM 7094, sin embargo, para 55 condados no tuvieron éxito debido a las limitaciones del hardware.

En [26] se discuten los diversos criterios para la redistribución de distritos políticos aceptable como son la equidad poblacional, compacidad, contigüidad y preservación de las fronteras naturales y/o políticas. En este artículo se muestra la viabilidad de tratar este problema por computadoras.

En [13] se reporta un método heurístico para resolver este mismo problema que a pesar de que no logra muy buenas soluciones (es incapaz de elegir la mejor solución encontrada a través de la búsqueda) aporta una construcción de una buena solución inicial que podría ser utilizada para otros métodos.

En [47] trata el mismo problema en la nación de Gales para asegurar la equidad de los votantes. Se comparan tres métodos diferentes de solución: programación entera, particionamiento de conjuntos y enumeración implícita.

En [29] se propone la siguiente metodología para resolver el problema de la distribución de distritos políticos: 1) Utilizar relajación Lagrangeana para determinar los centros de los distritos, 2) Utilizar la técnica de transportación para asignar unidades poblacionales a los centros y 3) Resolver el problema de división por medio de una secuencia de problemas de transportación capacitada.

En [23] se describe un modelo de optimización basado en redes, que se

utiliza en Nueva Zelanda por el gobierno. El modelo trabaja bien para problemas de gran escala, se reporta su aplicación en un caso particular de 35,000 unidades geográficas y 95 distritos, cabe mencionar que las soluciones logran prácticamente la equidad poblacional y generalmente la forma de los distritos es compacta.

En [43] se presenta una heurística basada en optimización que incorpora los criterios antes mencionados. Se modela el problema como uno de particionamiento de grafos y se desarrolla una solución especializada basada en branch-and-price. Se demuestra la factibilidad de la metodología satisfaciendo el principio una-persona, un-voto con distritos compactos y contiguos para el estado de Carolina del Sur, Estados Unidos.

En [17] los autores emplean un método computacional intensivo para evaluar la base fáctica para la afirmación de la raza como factor predominante en contra del plan de distritos políticos de Carolina del Sur, Estados Unidos. Utilizan cuatro algoritmos que ponderan los criterios de distritos tradicionales. Se concluye que: 1) la raza es un factor en el diseño de los distritos políticos de Carolina del Sur, 2) la raza predomina por sobre la preservación de las líneas de los condados y 3) la raza puede predominar sobre la compacidad de los distritos.

En [15] se utiliza la exitosa metaheurística Búsqueda Tabú para lograr la distribución de distritos electorales bajo restricciones como contigüidad, equidad poblacional, compacidad y homogeneidad socio-económica. Se propone una formulación donde se incluyen varias restricciones en una sola función multicriterio para dividir 828 unidades geográfica en 19 distritos.

En [20] se presenta un algoritmo genético para resolver el mismo problema con las restricciones comunes. Este algoritmo utiliza una técnica basada en codificación y operadores genéticos utilizados para resolver el problema del agente viajero (TSP). Esta codificación fuerza la igualdad población de los distritos y utiliza una función de idoneidad para promover la contigüidad y la compacidad. Una etapa de pos-procesamiento refina aún más la equidad poblacional.

Por otro lado el diseño territorial para ventas y servicios consiste en la necesidad de las empresas de dividir el mercado en regiones de responsabi-

dad. En cuanto a los servicios se desea diseñar territorios para atender a los clientes o a instalaciones técnicas.

En general hay varios motivos para alinear territorios existentes o diseñar nuevos. Primero un incremento o decremento en los agentes de ventas obviamente requiere algún ajuste de los territorios. Otras razones son para lograr mejor cobertura con el personal existente o para balancear la carga de trabajo entre ellos. Además de cambios en la ubicación de los clientes o la introducción de un nuevo producto hacen necesario la modificación de los territorios.

Los criterios que se toman en cuenta son los Organizacionales: número de territorios, áreas básicas, asignación exclusiva de áreas básicas, ubicación de los agentes de venta. Geográficos: accesibilidad, compacidad. Criterios relacionados con la actividad: balance y maximización de ganancias [46].

Además de las aplicaciones políticas y de ventas y servicios también tienen aplicación para el diseño de territorios escolares, servicios de emergencias, distritos para plantas eléctricas, etc. Todos estos casos poseen características similares y han sido resueltos por lo tanto por técnicas similares.

1.2.2. Particionamiento

Las técnicas para resolver al problema del agrupamiento geográfico pueden clasificarse en dos grandes categorías: las jerárquicas y las no-jerárquicas o de particionamiento. La diferencia entre estas técnicas es que en las primeras se desconoce el número de grupos a formar y por lo tanto se busca encontrar el número de grupos ideal o natural de los datos proporcionados, mientras que en las técnicas de particionamiento se conoce el número de grupos a formar o bien el caso de estudio requiere la libertad de formar cualquier cantidad de grupos.

El algoritmo implementado en esta tesis entra en la categoría de particionamiento, en esta categoría se genera una solución inicial de k grupos, esta solución puede ser aleatoria o ser generada por medio de otras técnicas de manera inteligente. Después se hacen intercambios entre los elementos de los grupos o bien se intercambian los representantes de los grupos por otros para

generar nuevas soluciones, llamadas “vecinas”, hasta que se satisface alguna condición de parada.

Los métodos más utilizados para generar soluciones vecinas son los siguientes [38]

- *Swap Method*: Se selecciona un objeto al azar de un grupo y otro objeto de otro grupo y luego se hace un intercambio entre estos. Se describe como sigue: Dada una solución A , selecciona 2 objetos al azar $A(i)$ y $A(j)$ tales que no estén en el mismo grupo, luego se asigna $A(i)$ al grupo de $A(j)$ y viceversa.
- *Single Method*: Este es el modo más sencillo. Un objeto se mueve a la vez de un grupo a otro. Se describe como sigue: Dada una solución actual A , selecciona al azar $A(i)$, asigna X definida como un entero generado al azar en el rango $[1,k]$, ($k =$ número de grupos) y que es distinto del grupo al que pertenece $A(i)$. Entonces sea $A(i) = X$.
- *Threshold Method*: El modo de umbral de probabilidad se utiliza comúnmente para establecer las soluciones vecinas en la búsqueda tabú basado en el método de agrupamiento. Modera la “sacudida” sobre la solución actual. Entre mayor sea el valor del umbral de probabilidad, menor es la sacudida permitida y por lo tanto, las soluciones de prueba son más similares a la actual y viceversa.

En [39] Se encuentra una clasificación detallada de algoritmos de particionamiento, de la cual se ha tomado parte de la siguiente información.

Los algoritmos de particionamiento más populares son: k -medias [41, 1], k -medianas, PAM (Particionamiento Alrededor de los Medoides) y CLARA [35]. Los inconvenientes de estos algoritmos es que dependiendo de la calidad de la solución inicial será la calidad de la solución final o bien cuando se trabaja con soluciones iniciales aleatorias, cada ejecución de los algoritmos obtendrá una solución distinta, la cual podría ser un óptimo local.

K-medias

El algoritmo k -medias es uno de los métodos de partición más difundidos y populares. El origen del algoritmo menciona a MacQueen [41] como su

precursor, a partir del cual, se presentan diversas variaciones tales como aquellas propuestas por Anderberg [1] y Hartigan [25].

El algoritmo funciona como sigue. Dado un número inicial de conglomerados k , el objetivo del algoritmo es minimizar la distancias de los elementos dentro de cada conglomerado, respecto a su centro. La similitud de los objetos dentro de cada conglomerado es medida respecto al vector de promedios, llamado generalmente centroide. El criterio de parada usado es el error cuadrático medio definido por,

$$E = \sum_{l=1}^k \sum_{X \in c_l} |X - M_l|^2, \quad (1.1)$$

donde X es el punto en el espacio que representa a los objetos dados y M_l es el vector de promedios del conglomerado c_l . Existen varias formas de implementar el algoritmo, pero básicamente sigue los pasos expuestos en el algoritmo 1.

Algorithm 1 K-medias

- 1: Seleccionar arbitrariamente los k objetos que serán los centros o centroides iniciales de los conglomerados.
 - 2: Se asigna cada objeto al conglomerado con el centroide más cercano, con base al valor medio de los objetos en el conglomerado.
 - 3: Se recalculan los centros de los conglomerados, es decir se actualiza la media.
 - 4: Se iteran los pasos 2 y 3 hasta que se alcance la convergencia del criterio de parada, o hasta que los centroides se modifiquen levemente.
-

Características relevantes de este algoritmo: A menudo termina en un óptimo local, es escalable y eficiente en procesos que involucran grandes conjuntos de datos. Su complejidad es $O(nkt)$: n es el número de objetos, k el número de grupos y t el número de iteraciones. Es sensible al criterio de parada y a valores extremos ya que distorsionan la media.

Debilidades: Sensibilidad a la selección de los centroides iniciales, es decir, diferentes centroides producen finales diferentes.

K-Medias

Descriptivamente, la mediana es una medida más robusta que la media, puesto que no se ve influenciada por valores extremos; el algoritmo k-medias funciona de forma similar al algoritmo k-medias, sustituyendo el vector de promedios por el correspondiente vector de medianas como centro del conglomerado. Las medidas de disimilitud pueden ser la distancia de Manhattan o la distancia euclidiana al cuadrado. Los vectores de medianas de los objetos en cada conglomerado serán denotados por $M_e = \{M_{e_1}, \dots, M_{e_k}\}$.

Usando la medida de distancia de Manhattan, la función a minimizar está dada por la siguiente expresión,

$$P(W, M_e) = \sum_{l=1}^k d(X, M_{e_l}) = \sum_{l=1}^k \sum_{X \in c_k} W_l |X - M_{e_l}|, \quad (1.2)$$

sujeto a

$$\sum_{l=1}^k w_{i,l} = 1, 1 \leq i \leq n, 1 \leq l \leq k, w_{i,l} \in (0, 1),$$

donde $w_{i,l} \in W_l$ y $W = (W_l)$ es una matriz de pesos de dimensión $n \times k$ con vectores-columna representados por W_l y $M_{e_l} = [me_{l,1}, me_{l,2}, \dots, me_{l,k}] \in M_e$.

Algorithm 2 K-medias

- 1: Seleccionar arbitrariamente los k objetos que serán los centros o centroides iniciales de los conglomerados. El tipo de selección inicial de centroides es análogo a los presentados en el algoritmo K-Medias.
 - 2: Asignar cada punto al conglomerado con el centroide más cercano, a través de la distancia de Manhattan.
 - 3: Calcular el nuevo conjunto de centroides de los grupos, calculando la mediana de los nuevos grupos formados.
 - 4: Se iteran los pasos 2 y 3 hasta que se minimice la función objetivo 1.2, o hasta que los centroides apenas se modifiquen.
-

Este algoritmo posee la misma complejidad computacional que k-medias. Tiene la misma debilidad de ser sensible a la selección inicial de centroides; la ventaja es que la mediana no está influenciada por los valores extremos, por lo que se logra un método, en teoría, más robusto.

K-Medoids

Estos métodos fueron introducidos por Kaufman y Rousseeuw en 1987 [36], [54]. Están basados en el uso de objetos actuales del conjunto de datos para ser los representantes de los grupos, denominados medoids. Estos medoids se definen como los puntos localizados lo más al centro posible de los grupos y son representativos de la estructura de los datos. Cada objeto restante es agrupado con el medoid más cercano, e iterativamente estos algoritmos realizan todos los cambios posibles entre objetos medoids y los que no lo son, hasta que se minimice una medida de disimilitud entre los k-medoids y los vectores de observaciones que forman los grupos.

En este tipo de algoritmos se destacan PAM (Partitioning Around Medoids) y CLARA (Clustering Large Applications).

Algoritmo PAM

El algoritmo “Partitioning Around Medoids” (Particionamiento Alrededor de los Medoides) es un método tipo k-medoids que intenta determinar k particiones de n objetos determinando los objetos representativos de cada conglomerado (Ng and Hand 1994). Para encontrar los k medoides, PAM empieza con una selección arbitraria de k objetos representativos.

En cada iteración hace un intercambio entre un objeto seleccionado, O_i , y uno no seleccionado, O_h , si y solo si, el intercambio mejora la calidad del agrupamiento.

Algorithm 3 PAM

- 1: Seleccionar arbitrariamente k objetos representativos, los cuales serán los k-medoides iniciales.
 - 2: Calcular TC_{ih} para todos los pares de objetos O_i, O_h donde O_i es actualmente un medoide, y O_h no lo es.
 - 3: Seleccionar el par O_i, O_h el cual corresponde al $\min_{O_i, O_h} \{TC_{ih}\}$. Si el mínimo TC_{ih} es negativo, se intercambia O_i con O_h ; regresar al paso 2.
 - 4: Repetir el paso 2 y 3 hasta que no haya cambio.
 - 5: Asignar cada objeto a su medoide más cercano.
-

La principal ventaja del algoritmo PAM es la robustez del método en presencia de ruido, pues el cálculo del medoide está menos influenciado por ellos u otros valores extremos. PAM comienza a ser muy costoso a medida que el tamaño muestral n y el número de grupos k aumentan, siendo una de las principales desventajas de este algoritmo, razón por la cual es eficiente sólo para pocos datos.

Algoritmo CLARA

El algoritmo CLARA, separa múltiples muestras de la base completa y aplica el algoritmo PAM sobre cada una de ellas, luego, encuentra los conjuntos de k -medoids de las muestras. El principal motivo de Kaufman y Rousseeuw para proponer este algoritmo fue debido a la deficiencia del algoritmo PAM para trabajar con bases de datos con grandes volúmenes de información. Si estas muestras son realmente representativas de toda la base de datos, los medoids de las muestras deberían acercarse a aquellos que se hubiesen escogido de la base de datos completa. Según estos autores, los resultados experimentales indica que 5 muestras con $(40 + 2k)$ objetos cada una, producen resultados satisfactorios. La calidad del agrupamiento es medida con la disimilitud media de todos los datos, y no sólo aquellos objetos considerados en las muestras (Ng and Han 1994).

Algorithm 4 CLARA

- 1: Para $i = 1$ a 5 repetir los siguientes pasos.
 - 2: Seleccionar una muestra aleatoria de $s = (40 + 2k)$ objetos de la base completa.
 - 3: Ejecutar PAM sobre la muestra, para encontrar los k medoides de esta muestra.
 - 4: Para cada objeto O_j de la base completa, determinar su medoide más cercano y agruparlos.
 - 5: Calcular la disimilitud media del agrupamiento obtenido. Si este valor es menor al mínimo actual, usar este valor como el mínimo actual y conservar los k medoides obtenidos en el paso 3 como el mejor conjuntos de medoides obtenidos.
 - 6: Retornar al paso 1 y comenzar la próxima iteración.
-

La efectividad de CLARA depende tanto del tamaño de la muestra como

de su calidad. Note que PAM busca los mejores k medoides entre las muestras seleccionadas del conjunto total de datos. CLARA no podría encontrar la mejor agrupación si los mejores k medoides no son seleccionados dentro de las muestras.

El algoritmo CLARA presenta una complejidad mayor en cada iteración, puesto que depende adicionalmente del tamaño de la muestra seleccionada.

En esta tesis, el modelo matemático propuesto fue probado también en PAM antes de implementar la búsqueda tabú, sin embargo, como se expuso anteriormente, PAM es ineficiente respecto a tiempo de cómputo para grandes cantidades de datos agrupados en un gran número de grupos.

1.2.3. Metaheurísticas

El término heurística proviene del griego *heuriskein* que significa encontrar o descubrir, de la cual se deriva *eureka*, la famosa exclamación de Arquímedes al descubrir su principio [51].

Melián concibe una definición acertada:

Se califica de heurístico a un procedimiento para el que se tiene un alto grado de confianza en que encuentra soluciones de alta calidad con un coste computacional razonable, aunque no se garantice su optimalidad o su factibilidad, e incluso, en algunos casos, no se llegue a establecer lo cerca que se está de dicha situación. Se usa el calificativo heurístico en contraposición a exacto... [44]

Ahora bien las metaheurísticas han surgido con el propósito de obtener mejores resultados que los logrados por la heurísticas. El sufijo “meta” significa “más allá”, a un nivel superior. El término metaheurística fue introducido por Fred Glover en 1986 y a partir de entonces han aparecido muchas propuestas de pautas o guías para diseñar mejores procedimientos de solución de problemas combinatorios [51].

Los procedimientos Metaheurísticos son una clase de métodos aproximados que están diseñados para resolver problemas difíciles

de optimización combinatoria, en los que los heurísticos clásicos no son efectivos. Los metaheurísticos proporcionan un marco general para crear nuevos algoritmos híbridos combinando diferentes conceptos derivados de la inteligencia artificial, la evolución biológica y los mecanismos estadísticos [48].

En [51] se clasifican las metaheurísticas de la siguiente manera:

Metaheurísticas constructivas Incorporan iterativamente elementos a una estructura inicialmente vacía que representa a la solución. Ejemplo: GRASP.

Metaheurísticas evolutivas Construyen un conjunto de soluciones durante el proceso a diferencia de los otros métodos que pasan de una solución a otra en cada iteración. Ejemplo: Algoritmos Genéticos, Búsqueda Dispersa.

Metaheurísticas de búsqueda Son métodos que presuponen que existe una solución y realizan procedimientos de búsqueda, la diferencia con los métodos analíticos es que no necesariamente se encontrará la solución óptima. Ejemplo: Búsqueda Tabú, Recocido Simulado.

Razones por las que se utilizan las heurísticas y/o metaheurísticas:

- El problema es de una naturaleza tal que no se conoce ningún método exacto para su resolución.
- Aunque existe un método exacto para resolver el problema, su uso es computacionalmente costoso.
- El método heurístico es más flexible que un método exacto, permitiendo, por ejemplo, la incorporación de condiciones de difícil modelación.
- El método heurístico se utiliza como parte de un procedimiento global que garantiza el óptimo de un problema. Existen dos posibilidades:
 - El método heurístico proporciona una buena solución inicial de partida.

- El método heurístico participa en un paso intermedio del procedimiento, como por ejemplo la fase de refinamiento de la partición en el particionamiento de grafos multinivel.

[42]

A la fecha se han desarrollado una gran variedad de algoritmos metaheurísticos como son los bioinspirados: colonia de hormigas, enjambre de abejas. Los evolutivos, recocido simulado, GRASP, Búsqueda Tabú, entre otros. En seguida se presenta una breve descripción de tres metaheurísticas populares y efectivas.

GRASP (Greedy Randomized Adaptive Search Procedures)

Los métodos GRASP fueron desarrollados al final de la década de los 80 con el objetivo de resolver problemas de cubrimientos de conjuntos (Feo y Resende, 1989). El término GRASP fue introducido por Feo y Resende (1995) como una nueva técnica metaheurística de propósito general.

GRASP es un procedimiento de multi-arranque en donde cada paso consiste en una fase de construcción y una de mejora. En la fase de construcción se aplica un procedimiento heurístico constructivo para obtener una buena solución inicial. Esta solución se mejora en la segunda fase mediante un algoritmo de búsqueda local. La mejor de todas las soluciones examinadas se guarda como resultado final. La traducción de Greedy Randomized Adaptive Search Procedures (GRASP) al español puede ser: “Procedimientos de Búsqueda Voraces Aleatorios que se Adaptan”.

En la fase de construcción se construye iterativamente una solución posible, considerando un elemento en cada paso. En cada iteración la elección del próximo elemento para la solución parcial viene determinada por una función greedy (voraz). Esta función mide el beneficio de añadir cada uno de los elementos según la función objetivo y elegir la mejor. Evidentemente no tiene en cuenta qué sucederá en las iteraciones sucesivas al realizar una elección, sino únicamente contempla la interacción actual.

Se dice que el heurístico greedy se adapta porque en cada iteración se actualizan los beneficios obtenidos al añadir un elemento a la solución parcial [42].

El heurístico es aleatorizado porque no selecciona el mejor candidato según la función greedy adaptada sino que, con el objeto de diversificar y no repetir soluciones en dos construcciones diferentes, se construye un lista con los mejores candidatos de entre los que se toma uno al azar.

En la fase de mejora se suele emplear un procedimiento de intercambio simple de tal manera que no se emplee mucho tiempo en esta mejora. Como GRASP se basa en realizar múltiples iteraciones y quedarse con la mejor solución de todas las encontradas no es de beneficio para el método detenerse mucho tiempo en mejorar una solución dada.

Algorithm 5 GRASP

```
while Condición de Parada no Cumplida do  
  Fase Constructiva  
  Seleccionar una lista de elementos candidatos.  
  Considerar una lista restringida de los mejores candidatos.  
  Seleccionar un elemento aleatoriamente de la lista restringida.  
  Fase de Mejora  
  Realizar un proceso de búsqueda local a partir de la solución construida  
  hasta que no se pueda mejorar más.  
  Actualización  
  Si la solución obtenida mejora a la mejor almacenada, actualizarla.  
end while
```

Las implementaciones GRASP generalmente son robustas en el sentido de que es difícil encontrar ejemplos en los cuales el método funcione arbitrariamente mal.

Su sencillez de implementación y su efectividad hacen a este método muy popular y exitoso inclusive en los problemas de diseño territorial [42].

Recocido Simulado

Kirpatrick, Gelatt y Vecchi proponen en 1983 un procedimiento para obtener soluciones aproximadas a problemas de optimización, llamado Simulated Annealing (Recocido Simulado). Este procedimiento es una analogía al comportamiento de un sistema físico al someterlo a un baño de agua caliente. Recocido Simulado ha sido probado con éxito en numerosos problemas de

optimización, mostrando gran “habilidad” para evitar quedar atrapado en óptimos locales. Su auge en los 80’s se debe a su sencillez de implementación aunado a los buenos resultados que ha obtenido.

Le estrategia que sigue el recocido simulado consiste en iniciar con una temperatura inicial muy alta, y con base en un factor de enfriamiento α se reducirá en cada iteración hasta llegar a una temperatura final. De acuerdo a la temperatura actual es la probabilidad de aceptación de soluciones que empeoren o mejoren la solución, es decir, mientras más alta sea la temperatura hay una mayor probabilidad de que la solución por elegir empeore la solución actual, esta probabilidad se reduce conforme la temperatura disminuye, de tal manera que cuando se llega a la temperatura final la probabilidad de elegir “malas” soluciones tiende a cero o bien llega a ser cero. Este es el elemento clave de recocido simulado para evitar quedar atrapado en óptimos locales.

Algorithm 6 Recocido Simulado

Require: Solución Inicial x

Require: Temperatura inicial T

```

while No Congelado do
  Realizar  $L$  veces
  Tomar  $x'$  de  $N(x)$ 
   $d = f(x') - f(x)$ 
  if ( $d < 0$ ) then
     $x = x'$ 
  end if
  if ( $d > 0$ ) then
     $x = x'$  con  $p = d^{-e/T}$ 
  end if
   $T = rT$ 
end while

```

[42]

Para diseñar un algoritmo propio de recocido simulado partiendo del algoritmo 6 debemos tomar en cuenta los siguientes elementos:

Temperatura Inicial Se determina a partir de pruebas o de un diseño de experimentos para lograr una fracción determinada de movimientos aceptados.

Velocidad de enfriamiento Qué tan rápido se enfría el “sistema”, algunos investigadores han utilizado un factor muy pequeño mientras que otros, factores grandes para un enfriamiento rápido. Depende del problema y del investigador entre otros elementos determinar este valor.

Longitud L Por lo regular se toma el valor L en proporción con el tamaño esperado de $N(x)$.

Temperatura final Es el criterio de parada; cuando el “sistema” se congeló o enfrió. Depende también del problema y del investigador.

En [10] Bernábe y otros utilizan la metaheurística recocido simulado para resolver un problema de regionalización, mostrando un rendimiento muy superior en cuanto a tiempo de cómputo con respecto a la técnica clásica PAM, además que logra el objetivo de escapar de los óptimos locales, uno de los mayores inconvenientes de trabajar con paquetes estadísticos y se logra el cumplimiento de las restricciones de compacidad y contigüidad.

Uno de los inconvenientes de RS es que no existe un método o técnica exacta para determinar adecuadamente sus parámetros, depende en sí del investigador y del problema en cuestión. En [7] se utiliza un diseño factorial de experimentos junto con análisis de superficies de respuesta para calibrar los parámetros de un algoritmo de recocido simulado para el problema del agrupamiento geográfico.

Búsqueda Tabú

Los orígenes de la búsqueda tabú (Tabu Search, TS) pueden situarse en diversos trabajos publicados a finales de los 70. Oficialmente, el nombre y la metodología fueron introducidos posteriormente por Fred Glover y Manuel Laguna en 1989 en el libro Tabu Search [42, 24].

La filosofía de la búsqueda tabú es derivar y explotar una colección de principios inteligentes para la resolución de problemas. En este sentido, puede decirse que la búsqueda tabú está basada en selectos conceptos que unen los campos de la inteligencia artificial y la optimización.

La búsqueda tabú es una metaheurística que guía a una heurística local de búsqueda para explorar el espacio de soluciones más allá de los óptimos

locales. El procedimiento local es una búsqueda que utiliza una operación llamada *move* para definir el vecindario de cualquier solución dada. Uno de los componentes principales de TS es el uso de memoria adaptativa, la cual crea un comportamiento más flexible de búsqueda. Las estrategias basadas en memoria son el sello distintivo de los enfoques de búsqueda tabú.

La búsqueda tabú se preocupa por imponer restricciones para guiar un proceso de búsqueda para negociar regiones que de otra manera serían difíciles de acceder. Las restricciones son impuestas o creadas al hacer referencia a estructuras de memoria las cuales están diseñadas para este propósito en específico.

Está basada en la premisa de que la resolución de problemas, de tal manera que se considere inteligente, debe incorporar memoria adaptativa y exploración responsiva.

La búsqueda tabú contrasta con las metaheurísticas sin memoria en que las últimas dependen fuertemente en procesos semi-aleatorios que implementan una especie de muestreo. Ejemplos de métodos sin memoria son aquellos inspirados por metáforas de la física y biología y heurísticas semi-greedy [24].

Los siguientes trabajos muestran la eficacia de TS en los problemas de agrupamiento:

En [38] se propone un enfoque de búsqueda tabú modificado que consiste en 2 fases: la fase constructiva en la cual se genera la solución inicial utilizando el algoritmo K-medias y una fase de mejora, donde se emplea una búsqueda tabú modificada con el objetivo de mejorar la solución de la fase constructiva.

Logra mejores soluciones que K-medias y que búsqueda tabú estándar, sin embargo sólo está enfocado a la minimización de distancias, esto quiere decir que logra la formación de grupos compactos pero no considera restricciones de homogeneidad de ningún tipo.

En [16] se propone una metaheurística híbrida que combina la búsqueda dispersa (Scatter Search) y la búsqueda tabú para tratar problemas de optimización multiobjetivo, es decir, para encontrar soluciones óptimas para n funciones objetivo a minimizar o maximizar o bien cuando se utilizan más

de una matriz de disimilitud.

A esta metaheurística se le ha nombrado STS (Scatter Tabu Search), básicamente se realizan $m + 1$ búsquedas tabú como primera fase, donde m es el número de funciones objetivo. La primer búsqueda tabú busca un valor óptimo para una de las funciones objetivo seleccionada al azar, después la siguiente búsqueda tabú se realiza sobre esta solución óptima pero ahora buscará un valor óptimo para otra función objetivo y la última búsqueda vuelve a buscar un valor óptimo para la primer función objetivo seleccionada pero ahora sobre la solución óptima obtenida por la búsqueda número m .

En la segunda fase entra la búsqueda dispersa con el objetivo de combinar las soluciones obtenidas en la fase anterior y mejorarlas aún más.

Los resultados muestran que se logra obtener fronteras eficientes de gran calidad para diversos problemas multiobjetivo.

En [27] se concluye lo siguiente sobre TS: muchos experimentos computacionales han mostrado que la búsqueda tabú se ha convertido en una técnica de optimización establecida la cual puede competir con casi todas las técnicas conocidas y por su flexibilidad puede vencer a muchos procedimientos clásicos. Sin embargo, hasta ahora no hay explicación formal sobre este buen comportamiento.

Comparación de Metaheurísticas

A continuación se presenta una tabla comparativa de algunas metaheurísticas tomando en cuenta las siguientes características: si utilizan Memoria Adaptativa (A) o Sin Memoria (M). Por el tipo de exploración de vecindarios que utilizan; ya sea por alguna forma de búsqueda sistemática de vecindarios (N) o bien por muestreo aleatorio (S) y finalmente por el número de soluciones almacenadas de una iteración a otra, es decir el movimiento de una solución a la siguiente (1) o un enfoque basado en población con tamaño (P).

¿Por qué 2 clasificaciones? La primera clasificación corresponde a la concepción “popular” u original de las metaheurísticas descritas, mientras que la clasificación 2 representa las variantes que se han originado con el paso del tiempo para mejorar los métodos, aunque en algunos casos estas variantes

Metaheurística	Clasificación 1	Clasificación 2
Algoritmos Genéticos	M/S/P	M/N/P
Búsqueda Dispersa	M/N/P	A/N/P
Recocido Simulado	M/S/1	M/N/1
Búsqueda Tabú	A/N/1	A/N/P

Cuadro 1.2: Clasificación de Metaheurísticas.

fueron propuestas desde la concepción de la metaheurística como es el caso de búsqueda tabú que desde sus orígenes se propone el uso del almacenamiento de varias soluciones. Recientemente investigadores han realizado métodos híbridos que combinan las bondades de dos o más metaheurísticas, en muchos de los casos se incorporan aspectos de la búsqueda tabú como son la memoria y las restricciones sobre movimientos [24]

Capítulo 2

EL PROBLEMA

2.1. Definición del Problema

El problema del agrupamiento geográfico es un caso particular del Diseño Territorial, del cual se desprenden diversos problemas similares entre sí pero con restricciones y especificaciones particulares que requieren del uso de técnicas variadas, principalmente para lograr la partición óptima de grandes territorios en tiempos razonables de cómputo. El capítulo 1 presenta a grandes rasgos el panorama de investigación sobre el diseño territorial, particionamiento y técnicas metaheurísticas.

El diseño de territorio puede ser visto como un problema de agrupación de unidades geográficas pequeñas (áreas básicas) en clusters geográficos más grandes llamados territorios, del tal forma que la agrupación aceptable es aquella última que cumpla con ciertos criterios predeterminados del problema que ocupa [46]. Estos criterios a cumplir obedecen a la naturaleza de un particular problema donde restricciones espaciales son muy demandadas (Ver la Tabla 1.1). Los criterios que definen el objetivo de esta tesis son la compacidad y la homogeneidad en cuanto al número de elementos en cada grupo; esta homogeneidad es de tipo *intergrupo*, es decir los grupos comparten la característica de poseer más o menos la misma cantidad de unidades geográficas. Por otro lado la homogeneidad de tipo *intragrupo* busca otro tipo de objetivos; como dividir un territorio por zonas económicas donde cada grupo deberá ser conformado por unidades geográficas cuya población tiene un nivel económico similar, es decir, los elementos en cada grupo comparten

una o más características que son distintas a las de otro grupo.

La homogeneidad intergrupo es una necesidad para la distritación política para fines electorales que como se expone en [46] desde los 70's se ha trabajado este problema con el fin ubicar las casillas electorales a una distancia uniforme de los electores, a la vez de que a cada casilla debe atender a la misma cantidad de electores o a la misma extensión de territorio o al mismo número de viviendas, etc.

El objetivo principal, como ya se mencionó antes, es implementar un modelo matemático para agrupamiento geográfico capaz de resolver compacidad y contigüidad espacial de los AGEBS [31] y así mismo homogeneidad en cuanto al número de elementos en cada grupo, utilizando la metaheurística Búsqueda Tabú para resolver instancias de gran tamaño en tiempos razonables de cómputo.

En México, se cuenta con importantes contribuciones pioneras para esta línea de investigación, como lo son parcelación de territorio nacional y distritación electoral [53, 30]. En ambos casos consideran a las manzanas como las unidades geográficas y se trabaja la homogeneidad intergrupo, además las manzanas como se mencionó en el capítulo 1, facilitan el establecimiento de la compacidad geométrica entre ellas y la generación de un grafo de adyacencia por medio de la geometría computacional. Los grafos de adyacencia han permitido que la homogeneidad intergrupo haya sido tratada por diversos investigadores como un problema de particionamiento de grafos, logrando muy buenos resultados [33, 2, 40, 3, 4] sin embargo, al considerar a los AGEBS como unidades geográficas a zonificar, se complica el proceso de generar un grafo de adyacencia preciso por medio de las técnicas conocidas de la geometría computacional, dado que los AGEBS están separados por distancias no uniformes, no son totalmente contiguos y su forma es muy irregular y diferente entre sí. Justamente esta es la naturaleza de los AGEBS en México [7].

En los inicios de este trabajo se intentó abordar el problema de la misma manera que en [53] por medio de un grafo de adyacencia generado con la triangulación de Delaunay a partir de las coordenadas de cada AGEBS, sin embargo, por las características antes mencionadas de los AGEBS esta propuesta no tuvo éxito porque el grafo de adyacencia generado no es una

representación precisa de la topografía del mapa de AGEBS por lo que se decidió seguir tratando al problema como uno de agrupamiento geográfico modificando el modelo matemático de optimización combinatoria utilizado por Bernábe y otros en [7] para considerar también la homogeneidad inter-grupo.

Aunado al modelo matemático que resuelve compacidad y homogeneidad se implementa la técnica metaheurística Búsqueda Tabú adaptada a la nueva propuesta de modelo, con el fin de mejorar los tiempos de computo logrados por las técnicas de particionamiento clásicas como PAM.

2.2. Modelo matemático implementado

Se hace una modificación en el modelo matemático utilizado en [7] para considerar la homogeneidad en cuanto al número de unidades geográficas en cada grupo ya que este modelo solamente considera la compacidad como único objetivo. Por lo tanto el problema ahora es un problema de optimización multiobjetivo.

Cuando se desea encontrar el valor óptimo para más de una función objetivo el problema reside en que normalmente no existe un elemento en el espacio de soluciones que produzca un óptimo de forma simultánea para cada uno de los objetivos. Esto se debe a la existencia de conflictos entre objetivos, es decir, la mejora de uno de ellos puede dar lugar al empeoramiento de otro. Como ejemplo ilustrativo podemos pensar en el encordado de una raqueta de tenis. En este proceso hay 3 objetivos principales que se deben de tomar en cuenta: la potencia, el control y la durabilidad, que dependen del tipo de cuerda que se utilice y la tensión con la que serán colocadas con ayuda de una máquina especializada.

Los 3 tipos principales de cuerda son: Naturales (tripas), Sintéticas (Nylon) y Poliéster. Una cuerda de poliéster tardará mucho más tiempo en reventarse que una de nylon o de tripas, es decir, tienen mayor durabilidad que las demás, además proporciona mejor control pero poco poder. Por otro lado las cuerdas de nylon y de tripas proporcionan más poder debido a su flexibilidad pero son fáciles de reventar (poca durabilidad) y el control se

reduce.

El principio de las tensiones de las cuerdas es básicamente: entre más tensa la cuerda mayor control y menos poder mientras que con baja tensión se logra mayor poder y menos control.

Es el jugador quien deberá decidir la combinación *óptima* de tipo de cuerdas y la tensión para lograr el efecto deseado en su estilo de juego.

Este ejemplo ilustrativo muestra claramente el conflicto entre los objetivos, ya que si se requiere poder se deberá de sacrificar durabilidad y control y viceversa. En nuestro problema los objetivos deseados son la *compacidad* y la *homogeneidad* que también entran en conflicto y complica la búsqueda de soluciones óptimas para ambos objetivos.

A continuación se presenta el modelo de optimización multiobjetivo utilizado para lograr este propósito.

Definición 1 Compacidad

Si denotamos por $Z = \{1, 2, \dots, n\}$ al conjunto de n objetos a clasificar, se trata de dividir Z en k grupos $\{G_1, G_2, \dots, G_k\}$ con $k < n$ de tal forma que:

$$\bigcup_{i=1}^k G_i = Z$$

$$G_i \cap G_j = \emptyset, i \neq j$$

$$|G_i| \geq 1, i = 1, 2, \dots, k$$

Un grupo G_m con $|G_m| > 1$ es compacto si para cada objeto $t \in G_m$ cumple:

$$\min_{i \in G_m} d(t, i) < \min_{j \in Z - G_m} d(t, j), i \neq t. \quad (2.1)$$

Un grupo G_m con $|G_m| = 1$ es compacto si su objeto t cumple:

$$\min_{i \in Z - \{t\}} d(t, i) > \min_{j, l \in G_f} d(j, l), \forall f \neq m.$$

El criterio de vecindad entre objetos para lograr compacidad está dado por los pares de distancias descritos en 2.1.

Definición 2 Homogeneidad en cuanto al número de elementos

Sea $T_i = |G_i|$ para $i = 1, 2, \dots, k$ y dado un porcentaje de tolerancia de homogeneidad p (.1, .15, .2, ...) se obtienen dos cotas: la cota inferior $I = \lfloor n/k \rfloor - \lceil n/k * p \rceil$ y la cota superior $S = \lfloor n/k \rfloor + \lceil n/k * p \rceil$ donde n es el número de unidades geográficas y k el número de grupos a formar. Entonces una solución no es homogénea cuando:

$$\exists T_i \mid T_i < I \vee T_i > S \quad (2.2)$$

Modelo para Agrupamiento Geográfico

Sea UG el número total de AGEBS. Sea el conjunto inicial de n UG , $UG = \{x_1, x_2, \dots, x_n\}$, donde: x_i es la i -ésima unidad geográfica, (i es el índice de UG), y k es el número de zonas (grupos). Dado que se desean formar grupos y para referirnos a éstos, definimos: Z_l como el conjunto de las UG que pertenecen a la zona l , C_t es el centroide, y $d(i, j)$ es la distancia euclidiana del nodo i al nodo j (de un AGEBS a otro). Entonces se tienen como restricciones: $Z_l \neq \emptyset$ para $l = 1, \dots, k$ (los grupos no son vacíos), $Z_l \cap Z_m = \emptyset$ para $l \neq m$ (no existen AGEBS repetidos en distintos grupos), y $\bigcup_{l=1}^k Z_l = UG$ (la unión de todos los grupos son todos los AGEBS).

Una vez que se ha decidido el número k de centroides $c_t, t = 1, \dots, k$, a utilizar, hay que seleccionarlos en forma aleatoria y enseguida asignar los $AGEBS$ a los centroides de la siguiente manera: para cada AGEBS i

$$\min_{t=1, \dots, k} \{d(i, c_t)\}$$

cada AGEBS es asignado al centroide más cercano c_t .

El costo de homogeneidad de la solución está definido por la siguiente función:

$$h(T) = \begin{cases} T - S & \text{si } T > S \\ I - T & \text{si } T < I \\ 0 & \text{en cualquier otro caso} \end{cases} \quad (2.3)$$

Donde T es el tamaño de un grupo, I es la cota inferior y S la cota superior que delimitan el tamaño de los grupos.

Para cada valor de k se calcula la suma de las distancias de los AGEBS asignados a cada centroide y también para cada valor de k se calcula el número de elementos que rebasan los valores dados por la cota inferior y superior. Estos valores son ponderados con pesos $w1$ y $w2$ tal que $w1+w2 = 1$ y se suman. Se escoge el mínimo de este resultado y nit es el número de iteraciones. Esto puede expresarse como:

$$\min_{l=1, \dots, nit} \left\{ w1 \left(\min \left\{ \sum_{t=1}^k \sum_{i \in c_t} d(i, c_t) \right\} \right) + w2 \left(\sum_{j=1}^k h(T_j) \right) \right\} \quad (2.4)$$

Donde $h(T_j)$ es la función dada por la ecuación 2.3, que indica el total de unidades geográficas en cada grupo que están por encima de la cota superior o bien por debajo de la cota inferior.

Capítulo 3

DISEÑO DE ALGORITMO

El modelo propuesto en el Capítulo 2 es el resultado de un proceso de análisis y diseño de algoritmos del cual surgieron varias propuestas que fueron evolucionando y cambiando conforme se evaluaban y ponían a prueba, hasta que se llegó al modelo matemático final y la implementación de la metaheurística búsqueda tabú.

A continuación se presentan los algoritmos que se desarrollaron a lo largo de esta investigación, sus fortalezas, debilidades y las razones por las cuales fueron descartados.

3.1. Algoritmo Versión 1

Analizando el modelo del agrupamiento geográfico y el algoritmo de PAM, se determinó que se podría realizar una modificación en la manera en que se asignan los AGEBS a los centroides “forzando” a que se distribuyan equitativamente a cada grupo para mantener el balance en cada iteración y se construyó así la primer versión del algoritmo.

Este algoritmo recibe como entrada la matriz de disimilitud almacenada en un archivo. Tomando en cuenta las distancias dadas por la matriz de disimilitud se ideó la siguiente estrategia: Repartir las unidades geográficas hacia el centroide más cercano pero una vez que ese grupo alcance un tamaño igual a $\lfloor n/k \rfloor$ o bien $\lfloor n/k \rfloor + 1$ (dónde n es el número de AGEBS y k el

número de grupos a formar) entonces ese grupo “se cierra” y ya no pueden ser asignados más AGEBS a tal grupo, por lo tanto los AGEBS que deberían estar en ese grupo ahora se asignarán al siguiente centroide más cercano que aún no alcance el límite de elementos. La Figura 3.1 es el resultado de esta versión, se puede observar que la compacidad es mala ya que hay muchos elementos dispersos en el mapa mezclándose con elementos de otros grupos, sin embargo se pensaba que una vez incorporada esta estrategia a un algoritmo de particionamiento como PAM se obtendría una mejor compacidad ya que esta versión por sí misma genera una sola solución aleatoria.

Algorithm 7 Versión 1

Require: matriz D de disimilitud

Require: arreglo P de particiones de tamaño n

Require: arreglo T de tamaño de grupos

Require: arreglo C de centroides

Require: M tamaño promedio de grupos n/k

$C' = C$

$costo = 0$

for $i = 0$ hasta n **do**

$j = \text{centroideMasCercano}(i, D)$

$P_i = j$

$costo = costo + D_{ij}$

if $T_j == M$ **then**

$\text{descartaCentroide}(j, C')$

end if

end for

return $costo$

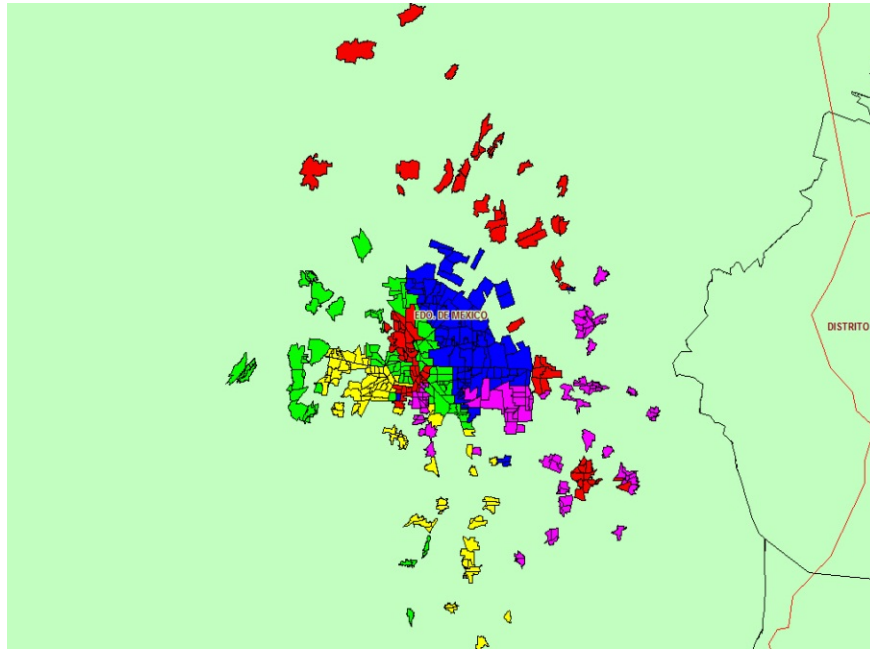


Figura 3.1: Versión 1. Homogeneidad y Compacidad para 5 grupos

3.2. Algoritmo Versión 2

La segunda versión consiste en haber incorporado la versión 1 del algoritmo en PAM. En la Figura 3.2 se puede observar que el resultado de este avance es prácticamente idéntico al de la versión 1.

Depurando el código y analizando los mapas se llegó a la conclusión de que la estrategia estaba equivocada en que el algoritmo no asegura que los AGEBS en cada grupo son realmente los más cercanos, es decir, a un grupo se le asignan AGEBS hasta que se alcanza el límite ($\lfloor n/k \rfloor$ o $\lfloor n/k \rfloor + 1$), entonces se da el caso en que AGEBS más cercanos al centroide que los ya asignados pasan a formar parte de otro grupo, sin embargo al corregir este inconveniente surgió otro problema: ahora los AGEBS se agrupan alrededor de los centroides y cuando los grupos alcanzan el límite, los AGEBS más lejanos al centroide pasan a formar parte de otro grupo con espacio disponible lo que provoca que algunas veces estos AGEBS terminen siendo asignados a un grupo no contiguo, por lo tanto vuelve a haber dispersión en los mapas

aunque en menor medida, sin embargo, el tiempo de cómputo se incrementó y para mejorar la compacidad se incrementaría aún más, por lo que esta nueva estrategia no resultó viable.

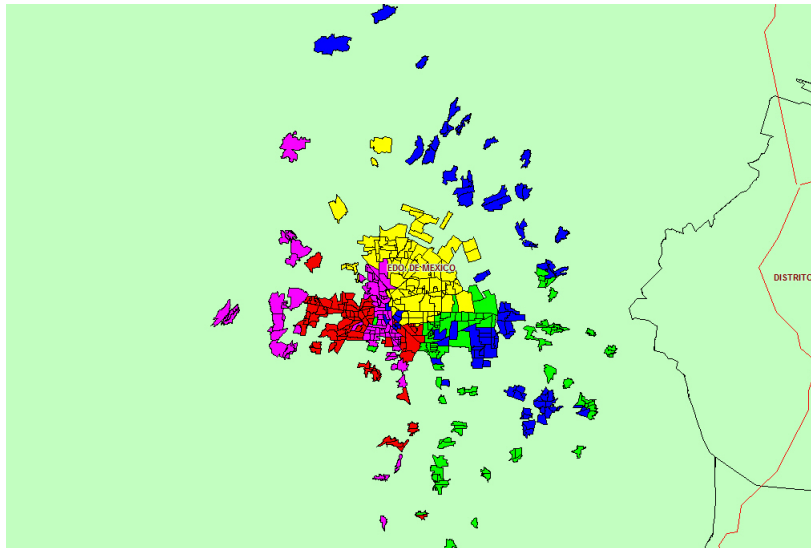


Figura 3.2: Versión 2. PAM modificado para 5 grupos

3.3. Algoritmo Versión 3

Debido a los problemas de dispersión de elementos y altos costos de cómputo de la versión 2 del algoritmo, esta versión 3 es en realidad un enfoque nuevo que rescata ciertos puntos de las versiones anteriores. En esta nueva estrategia los AGEBS se asignan al centroide más cercano como normalmente lo hace PAM y después se realiza un “ajuste” recursivo de homogeneidad: Se toma un grupo cuyo tamaño esté por debajo del límite ($|n/k|$ o $|n/k| + 1$), el cual debe de “robar” los AGEBS que le hacen falta del grupo que esté más cerca (se busca el centroide más cercano), a su vez si este grupo necesita AGEBS o bien no puede satisfacer la demanda del primer grupo debe de igual manera “robar” AGEBS al grupo más cercano distinto a los anteriores y así sucesivamente. En la figura 3.3 se puede ver el resultado de este nuevo algoritmo; funciona bien (en cuanto homogeneidad) para una cantidad de grupos pequeña pero conforme se incrementa esta cantidad, el tiempo de cómputo aumenta y como se detalla más adelante; nuevamente se presentan

problemas de dispersión (mínimos).

Algorithm 8 Ajuste Recursivo de Homogeneidad

Require: *costo* de la solución compacta
Require: *robar* elementos que necesita el grupo dado
Require: *UG* arreglo de Unidades Geográficas, tamaño n
Require: matriz D de disimilitud
Require: arreglo P de particiones de tamaño n
Require: arreglo T de tamaño de grupos
Require: arreglo C de centroides
Require: M tamaño promedio de grupos n/k
Require: centroide c_i con tamaño inferior a M

```

C' = C
descartaCentroide( $c_i, C'$ )
 $c_j$  = centrideMasCercano( $c_i, C'$ )
excedenete =  $T_j - M$ 
if robar <= excedente then
  cola = push( $c_i, robar$ )
  while cola no vacía do
     $u$  = pop(cola)
    for all  $P_l == c_j$  do
      lista = add( $UG_l$ )
    end for
    ordenaPorDistancia(lista)
    for  $h = 0$  hasta  $u \rightarrow robar$  do
       $v$  = remove(lista)
       $P_v = C_{u \rightarrow c}$ 
       $T_{C_{u \rightarrow c}} = T_{C_{u \rightarrow c}} + 1$ 
       $T_{c_j} = T_{c_j} - 1$ 
      costo = nuevoCosto( $P$ )
    end for
     $c_j = C_{u \rightarrow c}$ 
    limpia(lista)
  end while
else
  cola = push( $c_i, robar$ )
  robar = robar - excedente
  costo = recursión( $c_j, C, robar, costo$ )
end if
return costo

```

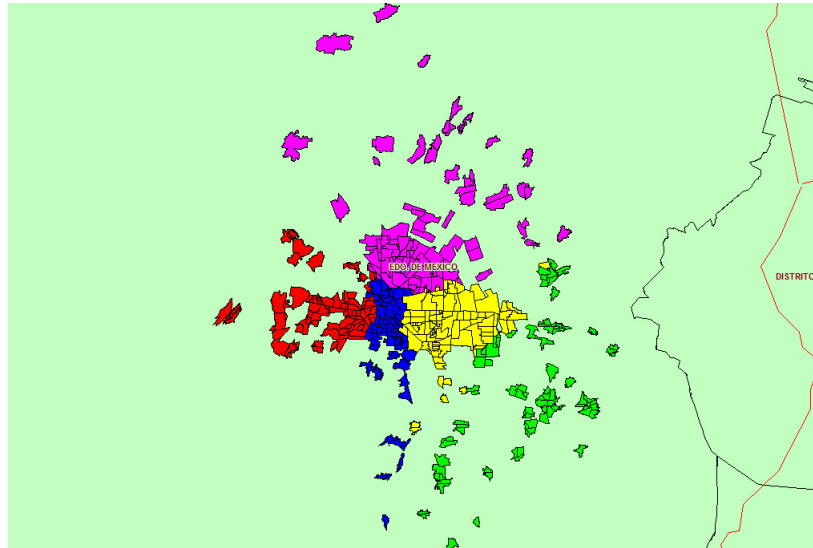


Figura 3.3: PAM modificado con versión 3 para 5 grupos.

Inconvenientes de la Versión 3

A partir de 15 grupos se presentan nuevamente los problemas de dispersión como se puede apreciar en la figura 3.4 y en el caso de 20 grupos es más evidente (figura 3.5).

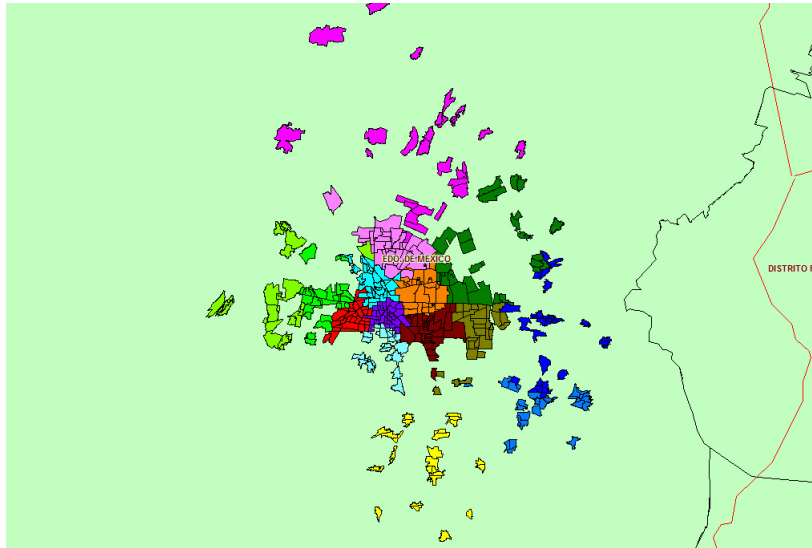


Figura 3.4: PAM modificado con versión 3 para 15 grupos, AGEBS dispersos.

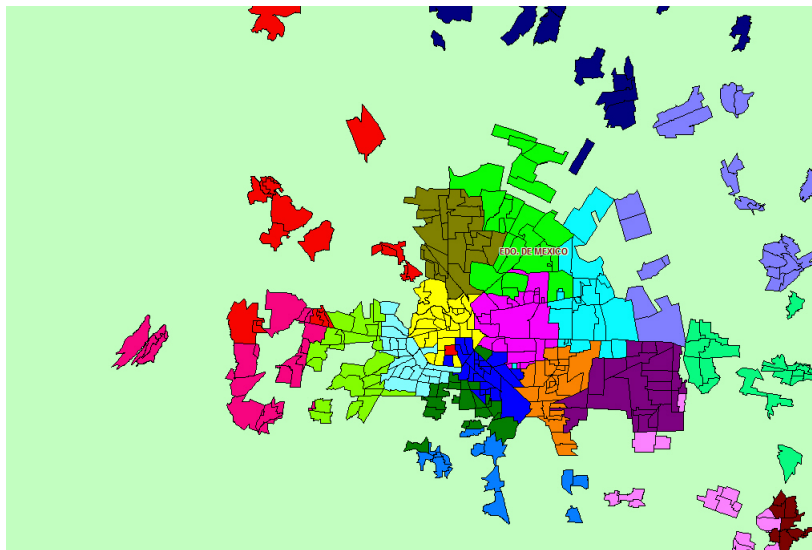


Figura 3.5: PAM modificado con versión 3 para 20 grupos, AGEBS dispersos.

Por estos problemas de dispersión y tiempo de cómputo considerablemente mayor al de PAM normal este algoritmo se ha descartado y se procede a diseñar e implementar nuevas estrategias.

3.4. Algoritmo de Particionamiento de Grafos

Después de revisar más literatura computacional se encontró que problemas muy similares eran resueltos por medio del particionamiento de grafos, el cual consiste en dividir un grafo de n nodos en k sub-grafos con un número de nodos igual a $\lfloor n/k \rfloor$ o bien $\lfloor n/k \rfloor + 1$ cuando las unidades geográficas no pueden dividirse exactamente en k grupos. Esto es la homogeneidad en la cardinalidad de los grupos formados.

Uno de los esquemas más utilizados para lograr el particionamiento de grafos es el denominado k -way multinivel, que consiste en 3 fases: “Graph Coarsening”, “Partición Inicial” y “Graph Uncoarsening y Refinamiento”.

Durante la fase de “coarsening”, una secuencia de grafos más pequeños $G_i = (V_i, E_i)$, es construida con base en el grafo original $G_0 = (V_0, E_0)$ de tal manera que $|V_i| < |V_{i-1}|$. En la mayoría de las técnicas de coarsening, un conjunto de vértices de G_i es combinado para formar uno solo en el grafo del siguiente nivel G_{i+1} . Sea V_i^v el conjunto de vértices de G_i combinados para formar el vértice v de G_{i+1} . Al terminar la etapa de coarsening se obtiene un grafo “reducido” que posee toda la información del grafo original. Para que una partición de este grafo reducido sea buena con respecto al original, el peso del vértice v es igual a la suma de los pesos de los vértices en V_i^v . Así mismo, para preservar la información de conectividad en el grafo reducido, las aristas de v son la unión de las aristas de los vértices en V_i^v .

Las técnicas más utilizadas para esta fase son las de “matching”. Un matching de un grafo es un conjunto de aristas, de las cuales ningún par de estas inciden en el mismo vértice. Así, el grafo del siguiente nivel G_{i+1} se construye a partir de G_i encontrando un matching de G_i y colapsando los vértices emparejados en multinodos. Los vértices sin pareja simplemente se copian al grafo G_{i+1} .

El concepto de matching puede entenderse mejor al observar la figura 3.6, donde se aprecia cómo ciertos vértices adyacentes se unen para formar un solo vértice.

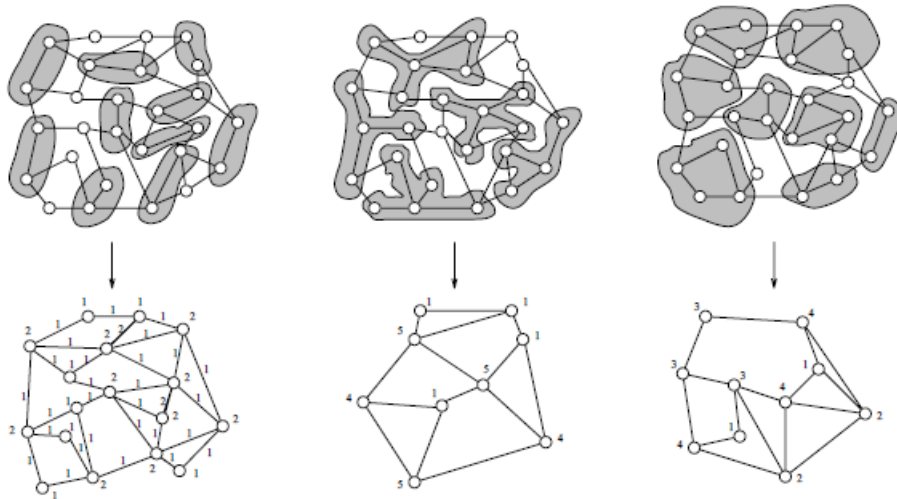


Figura 3.6: Técnica de Matching - los vértices se emparejan con otros para formar uno nuevo.

Para generar la solución inicial se “colapsa” el grafo hasta cierto límite (número de vértices) y sobre ese grafo representativo del original pero de menor número de vértices y aristas, se genera una partición inicial que puede ser generada con métodos como la bi-partición recursiva o bien un método capaz de generar las k particiones deseadas como algunos algoritmos Greedy, otra alternativa es “colapsar” el grafo hasta que tenga k nodos, siendo estos nodos las k particiones del original.

Una vez obtenida la partición inicial se procede a restaurar el grafo original nivel por nivel, en cada nivel los grafos reducidos adoptan la partición del nivel anterior. Partiendo de la solución inicial se hacen refinamientos en cada nivel para mejorar la solución y el balance, ya que es común que la solución inicial no sea del todo buena debido al coarsening. El refinamiento se hace por medio de heurísticas o metaheurísticas para reducir el tiempo de cómputo. Cuando se llega al último nivel (grafo original) se ha encontrado la solución, por lo regular perfectamente balanceada [34].

En la figura 3.7 se aprecia visualmente el funcionamiento del esquema multinivel para particionamiento de grafos, en el cual un grafo de gran tamaño se reduce a uno equivalente de menor tamaño utilizando técnicas de

matching, sobre el cual se realiza una partición inicial que se va trasladando hacia cada grafo generado del nivel superior hasta llegar al grafo original, corrigiendo y adaptando la partición en cada nivel.

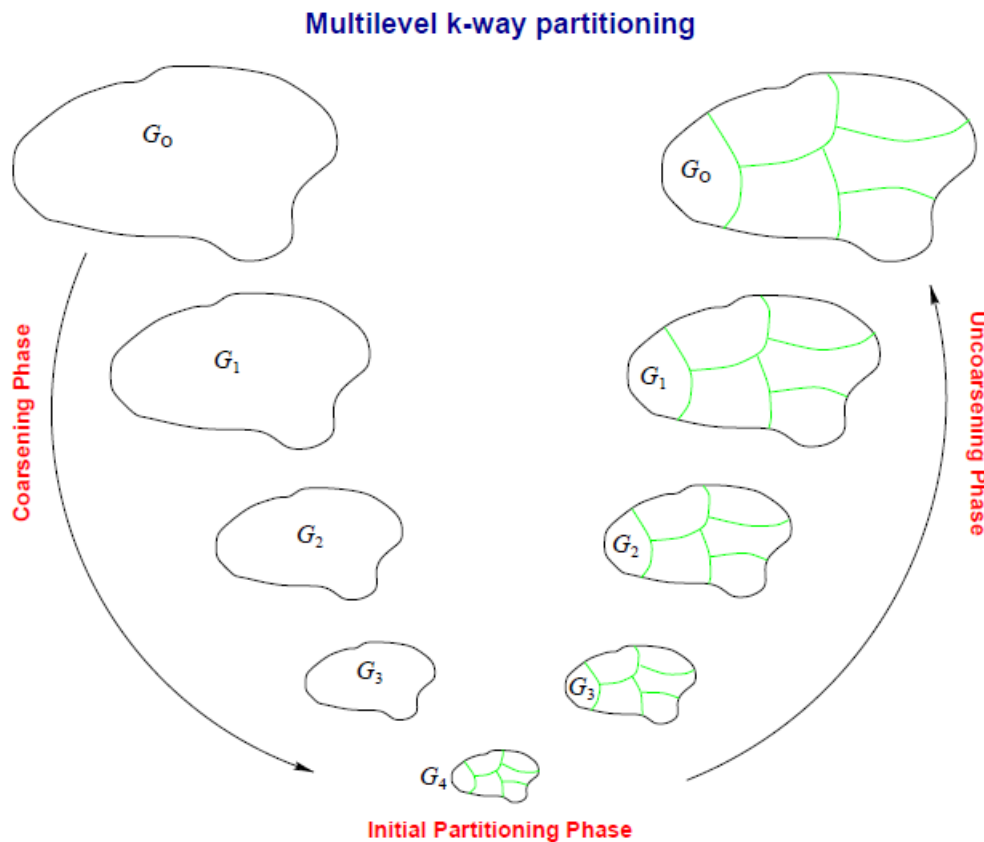


Figura 3.7: Esquema de particionamiento multinivel

Se desarrolló este esquema hasta la fase de la solución inicial ya que, por los siguientes motivos, se llegó a la conclusión de que este método no funciona para nuestro problema en particular.

Con base en el algoritmo 9, se intentaron 2 maneras para generar la solución inicial: un algoritmo greedy llamado *MinP - MaxNGreedy* [32] con el cual no se logró una buena solución inicial en el sentido que el algoritmo presenta problemas para generar sub-grafos (particiones) conexos, esto viola las restricciones de contigüidad y compacidad y por lo tanto la solución es

infactible. Otra manera de generar una solución inicial es como se sugiere en [34]: colapsar el grafo original hasta que se obtenga un grafo en el nivel i -ésimo que tenga k nodos. Esta estrategia logra generar grafos conexos pero como se puede ver en la figura 3.8 los grupos no cumplen con la restricción de compacidad porque su forma es irregular y no “circular”, esta última es una característica deseable y algoritmos como PAM o K-medias producen particiones con esta forma.

Algorithm 9 Particionamiento Multinivel k-way

Require: G Grafo de entrada

Require: factor f tamaño final para la reducción del grafo

Require: k número de particiones

if $tamano(G) > f * k$ **then**

$G_i = coarser(G)$

$p =$ recursión (G_i)

else

$p_i = SolucionInicial(G_i)$

return p_i

end if

$p_c = refina(p)$

return p_c

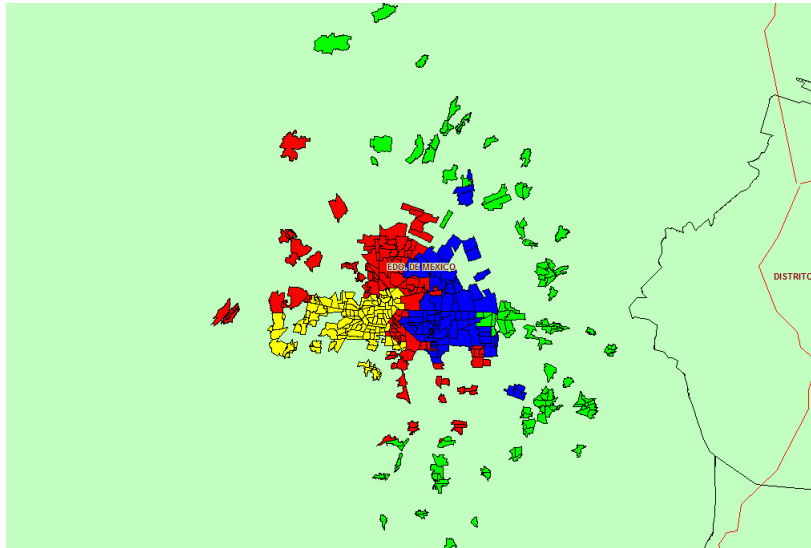


Figura 3.8: Solución Inicial con Particionamiento de Grafos

Razones por las que falla la estrategia.

1. El grafo que se genera con un algoritmo de Triangulación de Dealunay (TD) con base en las coordenadas de los AGEBS no es una representación precisa de las adyacencias debido a las características geográficas de los mapa de AGEBS, que no pueden ser representadas exactamente por la topología de un grafo generado por métodos de la geometría computacional como lo es TD. Cada AGEB presenta una forma bastante irregular muy distinta a la de los demás AGEBS y su tamaño varía mucho a diferencia de los mapas de AGEBS de manzanas utilizados en [53] cuyo grafo generado con el mismo algoritmo se puede apreciar en la figura 3.9 y es una representación fidedigna del mapa, mientras que el grafo generado para los AGEBS de Toluca que se aprecia en la figura 3.10 no es uniforme; hay aristas de diversas longitudes y nodos muy juntos o muy separados que son adyacentes. La forma irregular de los AGEBS en particular ocasiona que muchos de los nodos adyacentes en el grafo en realidad no son AGEBS adyacentes en el mapa. Todas estas razones generan problemas para obtener la solución inicial y para las metaheurísticas de refinamiento.

2. Otra de las razones por las que esta estrategia falla es la estructura de datos denominada “cubetas” (buckets) propuesta por Fiduccia y Mattheyses [19] para mejorar el algoritmo para bisección de grafos Kernighan-Lin [37] que consisten básicamente en un arreglo de listas ligadas para almacenar la “ganancia” que se obtendría de hacer un movimiento en la partición actual, como por ejemplo, mover un nodo frontera de una partición a otra. Cada partición tiene su propia cubeta que consiste en una lista de nodos para cada “nivel” de ganancia, por ejemplo, una lista para todos los nodos que al moverse generarían una ganancia de 2, esta ganancia se calcula en base a los pesos de los aristas, por lo tanto estos deben ser de tipo entero o de lo contrario no se podría pre-calcular la cantidad de “niveles” de ganancia, es este pre-cálculo el fuerte de los algoritmos de particionamiento de grafos ya que esto reduce significativamente el tiempo de cómputo de calcular las ganancias en tiempo real en cada iteración para todos los nodos y todos los movimientos posibles, siendo intratable para grafos de gran tamaño como lo es por ejemplo el de AGEBS de Toluca (469 nodos). El problema de utilizar la estructura de cubetas surge cuando el grafo a particionar maneja pesos con decimales, en nuestro caso el peso de cada arista debería de ser la distancia que hay entre los AGEBS que une, esta distancia está dada por la matriz de disimilitud y todas las distancias son valores decimales. La alternativa sería calcular en tiempo real las ganancias pero como se mencionó anteriormente el calculo en tiempo real de las ganancias es prohibitivo por el tiempo de cómputo, por lo tanto se decidió abandonar este enfoque ya que de la literatura revisada, todas las metaheurísticas de refinamiento utilizan esta estructura y algunas técnicas para generar la solución inicial también, como $MinP - MaxN$.



Figura 3.9: Grafo Uniforme generado con Triangulación de Delaunay. (Romero, 2002)

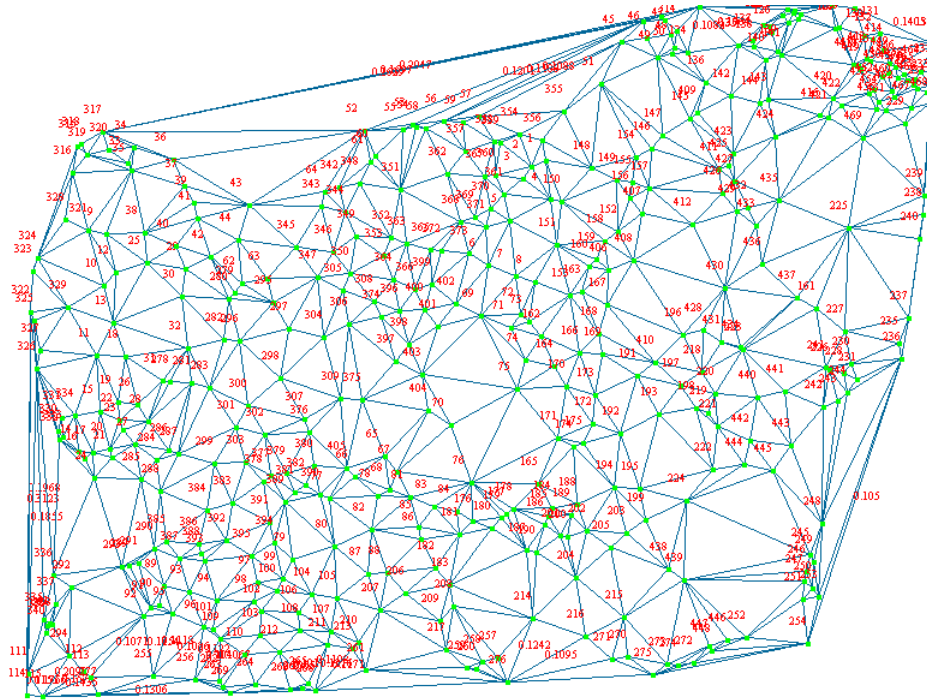


Figura 3.10: Grafo Irregular generado con Triangulación de Delaunay

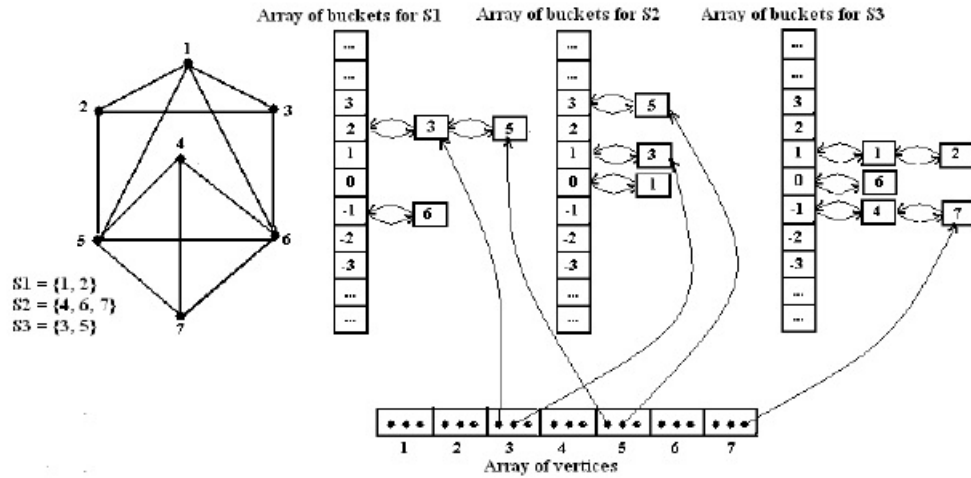


Figura 3.11: Estructura de Cubetas. (Benlic, 2010)

3.5. Versión 4. Modelo Matemático

Finalmente se decidió modificar el modelo matemático propuesto en [7] para considerar la homogeneidad en cuanto al número de unidades geográficas en cada grupo en la función objetivo.

3.5.1. Modelo Matemático Primera Propuesta

Definición 3 Homogeneidad en cuanto al número de elementos σ

Sea $T_i = |G_i|$ para $i = 1, 2, \dots, k$ y $M = n/k$ donde n es el número de unidades geográficas y k el número de grupos a formar. M es la media o el promedio de elementos que corresponderían a cada grupo (± 1 cuando no es entero). Entonces la desviación estándar está dada por:

$$DE = \sqrt{\frac{\sum_{i=1}^k (T_i - M)^2}{n}}$$

La desviación estándar nos indica qué tanto se desvían en promedio los valores T_i del tamaño promedio M . Por lo tanto al minimizar la desviación estándar se mejora la homogeneidad.

Modelo para Agrupamiento Geográfico

Sea UG el número total de AGEBS. Sea el conjunto inicial de n UG , $UG = \{x_1, x_2, \dots, x_n\}$, donde: x_i es la i -ésima unidad geográfica, (i es el índice de UG), y k es el número de zonas (grupos). Dado que se desean formar grupos y para referirnos a éstos, definimos: Z_l como el conjunto de las UG que pertenecen a la zona l , C_t es el centroide, y $d(i, j)$ es la distancia euclidiana del nodo i al nodo j (de un AGEBS a otro). Entonces se tienen como restricciones: $Z_l \neq \emptyset$ para $l = 1, \dots, k$ (los grupos no son vacíos), $Z_l \cap Z_m = \emptyset$ para $l \neq m$ (no existen AGEBS repetidos en distintos grupos), y $\bigcup_{l=1}^k Z_l = UG$ (la unión de todos los grupos son todos los AGEBS).

Una vez que se ha decidido el número k de centroides $c_t, t = 1, \dots, k$, a utilizar, hay que seleccionarlos en forma aleatoria y enseguida asignar los $AGEBS$ a los centroides de la siguiente manera: para cada AGEBS i

$$\min_{t=1, \dots, k} \{d(i, c_t)\}$$

cada AGEBS es asignado al centroide más cercano c_t .

Se realiza una suma ponderada: primero para cada valor de k se calcula la suma de las distancias de los AGEBS asignados a cada centroide y este valor se pondera con $w1$ y se calcula la desviación estándar de los tamaños de cada grupo representados por T_i y este valor se pondera con $w2$. Sumados estos dos valores ponderados se escoge el mínimo y nit es el número de iteraciones. Esto puede expresarse como:

$$\min_{l=1, \dots, nit} \left\{ w1 \left(\min_{t=1}^k \sum_{i \in c_t} d(i, c_t) \right) + w2 \left(\sqrt{\frac{\sum_{j=1}^k (T_j - M)^2}{n}} \right) \right\} \quad (3.1)$$

$w1$ y $w2$ son los pesos tales que $w1 + w2 = 1$.

Algoritmo de PAM utilizando el nuevo modelo

Se modifica PAM utilizando 3.1 como Función Objetivo a minimizar y partiendo de una configuración inicial de centroides aleatoria.

Algorithm 10 CalculaDE

Require: arreglo T de tamaño de grupos.**Require:** M tamaño promedio de grupos (n/k). $op = 0$ **for** $i = 0$ hasta k **do** $op = op + T_i^2$ **end for** $S2 = op/n - M^2$ **return** $\sqrt{S2}$

Algorithm 11 Función Objetivo

Require: arreglo T de tamaño de grupos.**Require:** arreglo C de centroides.**Require:** pesos $w1$ y $w2$. $costo = 0$ **for** $i = 0$ hasta n **do** $j = CentroideMasCercano(i, C)$ $costo = costo + d(i, j)$ $T_j = T_j + 1$ **end for****return** $w1 * CalculaDE(T) + w2 * costo$

Algorithm 12 PAM con Desviación Estándar

Require: matriz de disimilitud de tamaño $n \times n$ **Require:** entero k número de grupos a formar

- 1: Inicializar: selecciona k de los n objetos como medoides
 - 2: Asociar cada objeto al medoide más cercano
 - 3: **for** cada medoide m **do**
 - 4: **for** cada objeto no-medoide o **do**
 - 5: intercambiar m por o y computar el costo total de la configuración utilizando *Función Objetivo*
 - 6: **end for**
 - 7: **end for**
 - 8: Seleccionar la configuración con el menor costo
 - 9: Repetir 2 y 8 hasta que no haya cambios en los medoides
-

3.6. Modelo Matemático Versión Final

Después de hacer varias pruebas con el primer modelo y de revisar literatura computacional sobre esta línea de investigación se modificó la definición de homogeneidad; ahora se penalizan los grupos cuyo tamaño sea menor que una cota inferior dada o bien mayor a una cota superior. Las cotas se calculan en base a un porcentaje de tolerancia de homogeneidad, entre mayor sea este porcentaje los tamaños de los grupos varían también en mayor medida. Estas penalizaciones se suman y este valor se minimiza junto con la distancia total de las unidades geográficas hacia su respectivo centroide en una función objetivo ponderada de la misma manera que en la primera versión del modelo.

Esta versión final se encuentra en la sección 2.2 y consiste de las ecuaciones 2.1, 2.2 y 2.4 de esta tesis.

Capítulo 4

METAHEURÍSTICA BÚSQUEDA TABÚ

Con base en el modelo matemático de optimización multiobjetivo para resolver homogeneidad sobre agrupamiento de objetos de tipo geográfico, presentado en el capítulo 2 y a la función multiobjetivo del mismo, la ecuación 2.4, se ha implementado el método metaheurístico Búsqueda Tabú. En el capítulo 1 se ha expuesto cómo este método ha tenido mucho éxito para resolver problemas de optimización, mostrando en promedio mejor rendimiento que otros métodos metaheurísticos y además ha logrado obtener mejores soluciones.

4.1. Estructuras de Datos

La primer y principal estructura de datos es un arreglo de tamaño inicial k (número de grupos a formar) en el cual se almacenan los centroides de cada grupo.

Para la búsqueda tabú surge la necesidad de definir una lista que almacene los centroides tabú. Un centroide que *recientemente* reemplazó a otro centroide de la solución es considerado como tabú activo durante cierto número de iteraciones, de esta manera se *prohíbe* que este nuevo centroide sea reemplazado tempranamente. El tamaño de esta lista es dinámico pero el tamaño del arreglo de centroides más el tamaño de la lista de centroides tabú en todo

momento de la ejecución es igual a k , es decir, los centroides se dividen en 2 partes, los que se pueden reemplazar y los que no (centroides tabú).

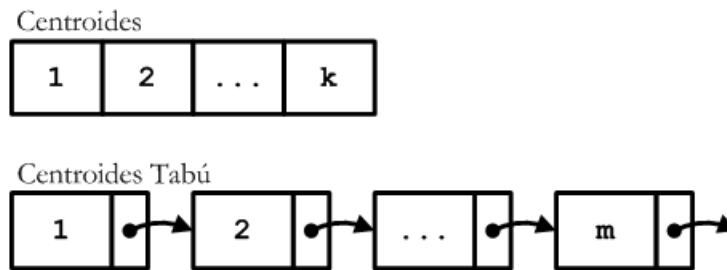


Figura 4.1: Estructura de Centroides

Las unidades geográficas AGEBS, se almacenan en un arreglo de tamaño inicial n (número de AGEBS) y para la búsqueda tabú se ha definido una lista en la que se almacenan los AGEBS tabú. Un Centroide que *recientemente* fue reemplazado por uno nuevo (antes un AGEBS) pasa a ser tabú activo durante cierto número de iteraciones para *prohibir* que forme parte de la solución muy pronto nuevamente. De esta manera la búsqueda se diversifica evitando volver a visitar soluciones antiguas.

La lista de AGEBS tabú es de tamaño dinámico y en todo momento su tamaño más el tamaño del arreglo de AGEBS es igual a $n - k$, ya que k AGEBS pasan a formar parte del arreglo de centroides.

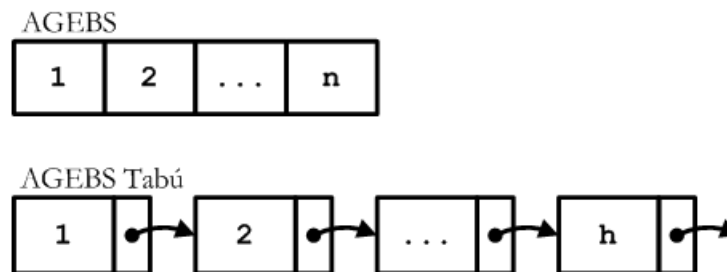


Figura 4.2: Estructura de AGEBS

Debido al objetivo de homogeneidad surge la necesidad de tener control

sobre los AGEBS asignados a cada centroide, por lo tanto se a implementado una matriz de tamaño $n - k \times k$. Cada columna representa un grupo o centroide y cada grupo puede tener como máximo $n - k$ elementos, sin contar a los centroides.

Se define también un arreglo de tamaño k en el cual se almacena el tamaño de cada grupo, inicialmente cada grupo tiene un tamaño igual a 1, ya que el centroide se contabiliza como elemento del grupo. Este arreglo se actualiza cada vez que se calcula el costo de cada solución aceptada durante la búsqueda, ya que es necesario asignar los AGEBS a cada centroide y llevar el control del tamaño de cada grupo en cada solución generada.

Matriz de Grupos

1,1	1,2	...	1,k
2,1	2,2	...	2,k
...
n,1	n,2	...	n,k

Figura 4.3: Matriz de Agrupamiento

4.2. Algoritmo de Búsqueda Tabú

El siguiente algoritmo es una adaptación de la búsqueda tabú a la propuesta de modelo presentada en el capítulo 2, cuyo elemento principal es la función objetivo dada por la ecuación 2.4.

Algorithm 13 Búsqueda Tabú para Agrupamiento Geográfico Homogéneo

Require: número de grupos k **Require:** número de iteraciones nit **Require:** número de iteraciones para la segunda fase $nit2$ **Require:** número de iteraciones para perturbación ip

```

1:  $pc = 0$ 
2:  $ic = 1$ 
3:  $S = SolucionInicial()$ 
4:  $S^* = S$ 
5: while  $PenalidadTotal(S^*) > 0 \wedge ic < nit$  do
6:   if  $PenalidadSuperior(S) == 0$  then
7:      $modoN = 1$ 
8:   else
9:      $modoN = 2$ 
10:  end if
11:   $costop = Costo(S)$ 
12:   $Movimiento(S, modoN)$ 
13:  if  $Costo(S) > costop$  then
14:     $pc = pc + 1$ 
15:  end if
16:  if  $Costo(S) < Costo(S^*)$  then
17:     $S^* = S$ 
18:  end if
19:  if  $pc > ip$  then
20:     $PerturbarSolucion(S)$ 
21:     $pc = 0$ 
22:  end if
23:   $ActualizaListasTabu()$ 
24:   $ic = ic + 1$ 
25: end while
26:  $S = S^*$ 
27:  $LimpiaEstadosTabu()$ 
28: for  $i = 0$  hasta  $nit2$  do
29:   $Movimiento(S, modoN)$ 
30:  if  $Costo(S) < Costo(S^*)$  then
31:     $S^* = S$ 
32:  end if
33:   $ActualizaListasTabu()$ 
34:   $ic = ic + 1$ 
35: end for
36: return  $S^*$ 

```

En la línea 1, se inicializa el contador de perturbación que empieza en 0 y en las líneas 13 y 14 si el costo de la nueva solución es peor que el de la solución anterior, este contador de perturbación se incrementa en 1, así, cuando este contador alcanza el valor máximo dado ip se perturba la solución actual en las líneas 19-21. La perturbación consiste en generar una nueva solución aleatoria y reiniciar las listas tabú, a partir de tal solución se reiniciará la búsqueda.

En la línea 3 se genera una solución inicial, eligiendo k objetos al azar como centroides de grupo y esta solución se almacena en S , en la línea 4 esta solución pasa a ser la mejor solución encontrada hasta el momento y es representada por S^* .

En la línea 5 inicia el primer ciclo de búsqueda que finaliza cuando la penalidad de la mejor solución encontrada llega a ser igual a 0 o bien cuando el contador de iteraciones ic alcanza el máximo número de iteraciones dado por el usuario.

La condición `if` dentro del ciclo, en las líneas 6 a 10 modifica la manera en que se elige el centroide a reemplazar en la función de vecindario. Cuando la penalidad superior es igual a 0, es decir, cuando no hay elementos en los grupos que sobrepasen la cota superior de homogeneidad (Consultar el modelo en el capítulo 2), entonces se cambia el modo de elección de centroide a 1, de esta manera se empiezan a elegir los centroides cuyo grupo es el de menor tamaño de entre los centroides no tabú. Si esta condición no se cumple el modo de elección 2 elige el centroide cuyo grupo es el de mayor tamaño de entre los centroides no tabú. Este tipo de elección de centroides permite que se reduzca más rápido la cantidad de objetos que violan las cotas de homogeneidad; el modo 1 tiende a elegir grupos cuyo tamaño está por debajo de la cota inferior, mientras que el modo 2, los de tamaño mayor a la cota superior. Se ha comprobado por medio de experimentación que esta estrategia logra disminuir más rápidamente la penalidad de homogeneidad que la selección aleatoria de centroides.

En la línea 11 se guarda el costo de la solución S antes de realizar el movimiento en la línea 12 sobre la misma. Una vez realizado el movimiento, en la condición `if` de las líneas 13-15 se comprueba si el costo de la nueva solución S es mejor que el costo previo, si no lo es, se incrementa el contador

de perturbación pc . La condición `if` de las líneas 16-18 se encarga de actualizar la mejor solución encontrada S^* si la nueva solución encontrada S es aún mejor.

En la línea 23 se actualizan las listas tabú (en la siguiente sección se proporcionan detalles de cómo se han implementado las listas tabú y otras estrategias tabú para este problema en particular). Después en la línea 24 se incrementa el contador de iteraciones ic .

Por último se realiza una segunda fase de búsqueda sobre la mejor solución encontrada en la primera fase, para esto se deben vaciar las listas tabú y después se realizan movimientos sobre la mejor solución durante cierto número de iteraciones ($nit2$) con el fin de encontrar aún una mejor solución que pueda estar cerca de S^* (a unos cuantos movimientos de distancia).

La solución retornada es la mejor encontrada después de las 2 fases de búsqueda.

4.3. Elementos y Estrategias Tabú

Tabu list Es una lista de los últimos movimientos (o soluciones). La memoria o movimientos pueden ser basados en recencia o bien frecuencia. La memoria a corto plazo basada en recencia prohíbe dar vueltas en un vecindario local en el espacio de soluciones a través de marcar a los últimos T movimientos como Tabú.

Por otro lado la memoria a largo plazo permite que la búsqueda sea conducida en los vecindarios más prometedores. La memoria basada en frecuencia provee información adicional sobre cuántas veces un movimiento tabú ha sido intentado.

Para el algoritmo de búsqueda tabú descrito anteriormente se han implementado 2 listas tabú, una para centroides y otra para AGEBS. La primer lista almacena los centroides *recientemente* agregados a la solución y que por lo tanto no pueden ser reemplazados durante cierto número de iteraciones (este valor está dado por *tabutenure*), mien-

tras que la lista tabú de AGEBS almacena los centroides *recientemente* reemplazados en la solución, que ahora pasan a ser AGEBS pero no pueden volver a formar parte de la solución durante cierto número de iteraciones (*tabutenure*).

Tabu Tenure Representa el número de iteraciones que un movimiento o solución permanecerá en estado tabú activo (en la lista tabú). Este elemento es un valor que puede ser fijo durante toda la búsqueda o cambiar de manera dinámica de acuerdo a la información recolectada durante la búsqueda o las circunstancias de las misma. Si el *tabutenure* tiene un valor grande se propicia la búsqueda intensa alrededor de las soluciones y un valor pequeño incita la diversificación de la búsqueda.

Por medio de experimentación se ha fijado este valor en $k - 1$ logrando una convergencia rápida en la función objetivo, sin embargo se le otorga al usuario la capacidad de alterar este valor.

Lista de candidatos TS hace uso de una lista de candidatos que provee una lista de movimientos a evaluar. Un movimiento de la lista de candidatos es escogido para proceder con la búsqueda. La lista de candidatos juega un rol importante en el rendimiento de TS. Esta lista en TS actúa como un vecindario reducido, donde solamente se consideran los movimientos válidos, es decir, los movimientos que no son tabú activos.

Para este problema se ha definido una función de vecindad que consiste en reemplazar un centroide de la solución actual por un AGEBS no centroide (el mejor posible). El centroide a reemplazar se elige en base a dos estrategias: Mientras haya elementos que sobrepasen la cota superior de homogeneidad se elige el centroide cuyo grupo es el de mayor tamaño de entre los centroides no tabú y cuando no hay elementos que sobrepasen la cota superior pero sí la cota inferior, entonces se elige el centroide cuyo grupo es el de menor tamaño. La función vecindario se explica con mayor detalle en el algoritmo 4.3.

Intensificación y diversificación Son dos estrategias fundamentales de la búsqueda tabú basadas en memoria. Con la estrategia de intensificación se exploran minuciosamente las regiones alrededor de soluciones

atractivas, típicamente opera al reiniciar una búsqueda desde una solución previa que se encontró que da buenos resultados. El reinicio se logra através de la candidate list que contiene soluciones atractivas. La diversificación por otra parte, promueve la examinación de regiones no visitadas and la generación de soluciones que difieren en varias formas significativas de las soluciones previas.

La intensificación empleada consiste en utilizar un *tabu tenure* alto que logra una mejor convergencia y una etapa final donde se realiza una última búsqueda alrededor de la solución mejor encontrada en la primera fase de búsqueda.

Por otro lado la diversificación se maneja a través del reinicio de la búsqueda dado por la estrategia de perturbación que consiste en generar una nueva solución aleatoria y limpiar las listas tabú, a partir de la nueva solución se prosigue con la búsqueda.

Condición de parada Comúnmente la búsqueda se detiene cuando un número predeterminado de iteraciones es alcanzado. Otros criterios de parada son: cuando un valor aceptable es alcanzado en la función objetivo, cuando no se presenta mejora en la función objetivo después de cierto número de iteraciones, entre otros de acuerdo al problema en cuestión.

Algorithm 14 Función Movimiento (Vecindario)

Require: solución S
Require: arreglo de AGEBS A
Require: modo de selección m

- 1: $localS = S$
- 2: $localA = A$
- 3: **if** $m == 1$ **then**
- 4: $indice = GrupoMasGrande()$
- 5: **else**
- 6: $indice = GrupoMasPequeno()$
- 7: **end if**
- 8: $Intercambia(localS[indice], localA[0])$
- 9: $costo_{local} = Costo(S)$
- 10: $Sn = localS$
- 11: $An = localA$
- 12: **for** cada $AGEBa \in An$ **do**
- 13: $Intercambia(Sn[indice], a)$
- 14: $costo_n = Costo(Sn)$
- 15: **if** $costo_n < costo_{local}$ **then**
- 16: $localS = Sn$
- 17: $localA = An$
- 18: $costo_{local} = costo_n$
- 19: **end if**
- 20: **end for**
- 21: $S = localS$
- 22: $A = localA$
- 23: $costo = costo_{local}$
- 24: $IniciaEstadosTabu()$
- 25: $ActualizaMemoria()$

En las líneas 4 y 5 se hace un respaldo de la solución actual S y los AGEBS no centroides A . En seguida en la condición IF se determina qué centroide será reemplazado, si el de mayor tamaño ($m = 1$) o el de menor tamaño (en otro caso). Después se realiza una búsqueda local donde cada AGEBS no centroide reemplaza al centroide elegido y se elige el mejor intercambio posible (el de menor costo).

Finalmente el mejor movimiento local encontrado ahora es solución actual S junto con sus AGEBS no centroides y su costo. En las últimas 2 líneas los elementos involucrados en el intercambio inician su estado tabú activo y se actualiza la memoria de la frecuencia en que los centroides aparecen en la solución.

4.4. Modificación Swap Method

Esta modificación al algoritmo de búsqueda tabú descrito en la sección anterior se desarrolló con el fin de mejorar el tiempo de cómputo para instancias aún más grandes.

En la sección anterior también se presentó un algoritmo de vecindario, el cual consiste básicamente reemplazar un centroide a la vez por un AGEBS no centroide de entre todos los AGEBS válidos posibles, esto implica una búsqueda exhaustiva pero muy lenta, por lo tanto se ideó un nuevo vecindario muy similar basado en *Swap Method* [38], en el cual los candidatos para ser centroides son solamente los AGEBS asignados al centroide a reemplazar, excepto cuando el tamaño del grupo es igual a 1, entonces se elige el mejor candidato de entre todos los AGEBS válidos posibles sin importar a que centroide pertenecen.

Por lo tanto solo se realizan pequeñas modificaciones al algoritmo de vecindario presentado anteriormente para obtener esta nueva versión llamada Búsqueda Tabú SM.

En el siguiente capítulo se exponen los resultados de las pruebas realizadas con las implementaciones de los algoritmos de Búsqueda Tabú y Búsqueda Tabú SM definidos en este capítulo y se comparan con una versión extendida de PAM que utiliza a la ecuación 2.4 del modelo propuesto en el capítulo 2 como nueva función objetivo.

Capítulo 5

Resultados y Conclusiones

5.1. Pruebas de las Implementaciones

Una vez definido el modelo matemático, se implementó en primer lugar una extensión de PAM que cambia la función objetivo de minimización de distancias estándar de PAM por la función multiobjetivo del modelo presentado en el sección 2.2 de esta tesis. Con esta extensión de PAM se realizaron varias pruebas sobre la instancia de AGEBS de Toluca, estas pruebas preliminares motivaron el desarrollo de una implementación de búsqueda tabú con el fin de mejorar el tiempo cuando se requiere mayor cantidad de grupos y así mismo mejorar los costos de la función multiobjetivo logrados por PAM extendido.

La tabla 5.1 presenta una comparación de los resultados obtenidos con los algoritmos de PAM extendido, Búsqueda Tabú y Búsqueda Tabú SM, estos dos últimos implementados a partir del modelo matemático propuesto de la sección 2.2 y cuyos algoritmos están expuestos en el capítulo 4.

El mapa de la ciudad de Toluca que consta de 469 AGEBS (Unidades Geográfica), sobre este se han resuelto varias instancias del problema que van desde formar 2 grupos hasta 200.

Características de la Máquina utilizada

CPU Intel Core 2 Duo 1.8 Ghz

Memoria RAM 3 GB

Sistema Operativo Windows 7 Ultimate

Versión de Java Java 7

N de Grupos	PAM Extendido			Búsqueda Tabú			Búsqueda Tabú SM		
	Compacidad	Penalidad	Tiempo(seg)	Compacidad	Penalidad	Tiempo(seg)	Compacidad	Penalidad	Tiempo(seg)
2	36.54853057861328	0	0.075	36.38659998588264	0	.990	36.38659999	0	52.204
4	27.456995010375977	0	0.222	27.366699991282076	0	2.010	27.40760009	0	49.113
6	31.428407669067383	0	0.670	23.39530006190762	0	3.360	23.21009999	0	44.539
8	27.248689651489258	0	1.051	21.03180003585294	0	4.293	20.30470004	0	43.303
10	19.942487716674805	0	3.042	17.988800055347383	0	5.531	17.34870006	0	42.721
20	13.654200553894043	0	24.041	13.684400002006441	0	24.261	12.3385	0	50.194
40	8.627695083618164	1	114.813	8.953100036014803	0	659.232	8.321600022	0	51.622
60	7.402600288391113	9	261.200	7.470600009779446	12	1730.979 *	5.952800002	22	114.057
80	5.318598747253418	17	469.071	5.637399987666868	15	1994.720 *	4.97729998	18	1153.886
100	4.524801731109619	6	860.164	5.15710001764819	10	2198.926 *	3.925900002	17	1476.732
120	3.999199867248535	11	1391.139	4.108800005138619	15	2102.419 *	3.3865	30	2279.279
140	2.8315000534057617	0	1894.436	3.7468000025255606	0	2197.013 *	2.972500009	3	3492.011
160	2.8784005641937256	11	2249.452	3.1313000022200868	13	2071.007 *	2.592800008	29	5329.231
180	2.048799991607666	0	2325.100	2.870880001026671	3	2011.102 *	2.413500007	9	3833.726
200	1.6022003889083862	0	2482.393	2.188500009593554	1	1857.146 *	1.8699	0	2222.87

Cuadro 5.1: Corridas de PAM Extendido y Búsqueda Tabú para la ciudad de Toluca.

*Los tiempos para búsqueda tabú marcados con * están sujetos a 5,000 iteraciones estrictas, mientras los tiempos que no tienen esta marca son corridas con búsqueda tabú en que la primer fase de búsqueda emplea una condición de parada en la que al encontrar una solución con penalidad 0 termina la primer fase y se realiza una segunda fase que consiste en 1,000 iteraciones (movimientos) sobre la mejor solución encontrada en la primer fase para estas pruebas.*

PARÁMETROS GENERALES

- **Peso 1 (Penalidad) 0.7**
- **Peso 2 (Compacidad) 0.3**
- **Tabu Add Tenure $k - 1$**
- **Tabu Drop Tenure 10**

PARÁMETROS DE BÚSQUEDA TABÚ

- **Número de Iteraciones Fase 1:** 5000 o bien cuando se encuentra una solución con penalidad 0 (sólo de 2 a 40 grupos)
- **Número de Iteraciones Fase 2:** 1000
- **Número de iteraciones de no mejora antes de perturbación:** 15

PARÁMETROS DE BÚSQUEDA TABÚ SM

- De 2 a 60 grupos
 - **Número de Iteraciones Fase 1:** 5000
 - **Número de Iteraciones Fase 2:** 1000
 - **Número de iteraciones de no mejora antes de perturbación:** 50
- De 80 a 160 grupos
 - **Número de Iteraciones Fase 1:** 100000
 - **Número de Iteraciones Fase 2:** 1000
 - **Número de iteraciones de no mejora antes de perturbación:** 500
- 180 y 200 grupos
 - **Número de Iteraciones Fase 1:** 50000 (180) y 20000 (200)
 - **Número de Iteraciones Fase 2:** 1000
 - **Número de iteraciones de no mejora antes de perturbación:** 200

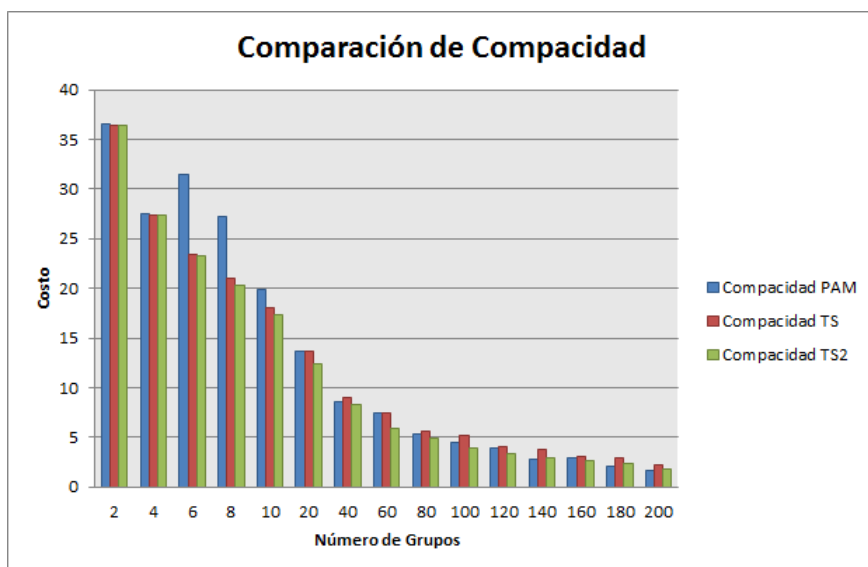


Figura 5.1: Comparación de costo de compacidad

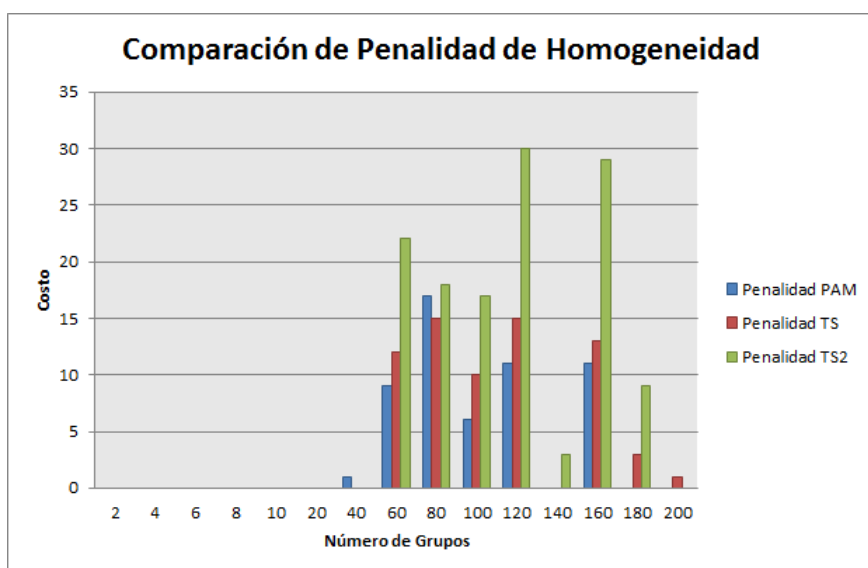


Figura 5.2: Comparación de la penalidad de homogeneidad

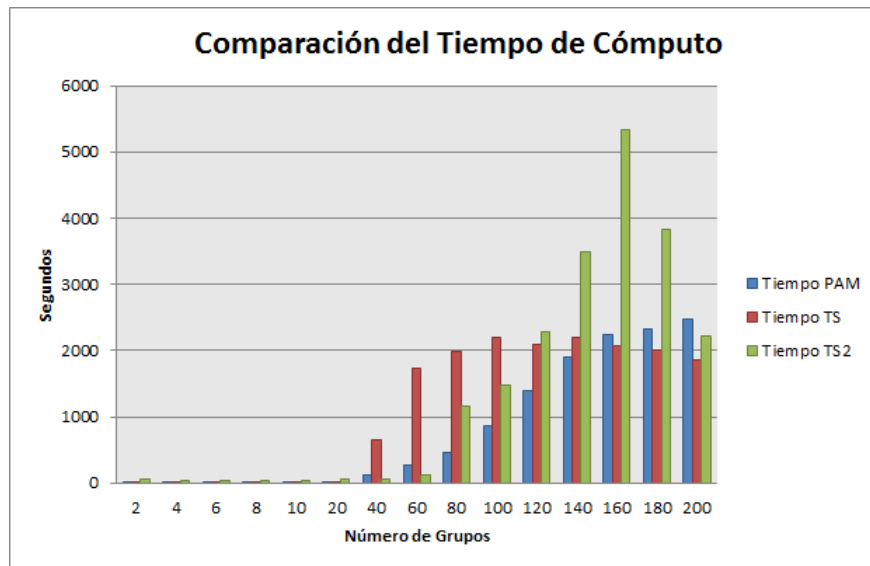


Figura 5.3: Comparación del tiempo de cómputo

Se puede observar que desde 2 a 40 grupos los 3 algoritmos logran sin problemas una penalidad igual a 0, esto quiere decir que son soluciones factibles para $\pm 10\%$ de tolerancia de homogeneidad en el tamaño de los grupos, sin embargo de 60 a 200 grupos PAM extendido es el algoritmo que logra mejores resultados en cuanto a la penalidad, logrando soluciones factibles para 140, 180 y 200 grupos, mientras que ambas versiones de Búsqueda Tabú no logran superar a PAM extendido en este rubro. Búsqueda Tabú logra una solución factible para 140 grupos y Búsqueda Tabú con Swap Method logra una solución factible para 200 grupos.

En cuanto a la compacidad, Búsqueda Tabú logra mejor compacidad que PAM extendido para 2, 4, 6, 8, 10 y 20 grupos. Búsqueda Tabú con Swap Method supera a PAM extendido en cuanto a la compacidad para todos los casos excepto para 140, 180 y 200 grupos y logra superar en todos los casos a Búsqueda Tabú.

En cuanto al tiempo de cómputo para pocos grupos PAM es muy rápido pero llega a tener un tiempo excesivo conforme se incrementa el número de unidades geográficas y la cantidad de grupos a formar, mientras que Búsqueda Tabú con Swap Method logra mantener un tiempo estable desde 2 a 40

grupos sujeto a 5000 iteraciones y en 60 el tiempo se incrementa al doble con la misma cantidad de iteraciones. Para 100000 iteraciones el tiempo es estable para 80, 100 y 120, para 140 y 160 se incrementa a más del doble y cabe destacar la última corrida de 200 grupos a 20000 iteraciones que logra una solución factible en 2222.87 segundos superior a Búsqueda Tabú pero no a PAM extendido.

5.1.1. Pruebas preliminares para una instancia grande

Se cuenta con un mapa de AGEBS de tamaño considerable, el mapa del Valle de México, que consta de 5087 unidades geográficas y se realizaron las siguientes pruebas con el algoritmo más prometedor, Búsqueda Tabú SM, que es el que logra mejores resultados considerando los 3 parámetros evaluados: Compacidad, Penalidad y Tiempo de Cómputo. La siguiente tabla muestra los resultados logrados en estas pruebas preliminares poco intensivas con respecto a la cantidad de iteraciones.

N de Grupos	Compacidad	Penalidad	Tiempo(minutos)
2	668.8969003177481	0	21.138
10	331.6363005102612	0	22.978
100	113.79350021231221	5	52.036
500	45.99800008238526	257	36.265
1000	31.125500035108416	749	36.619

Cuadro 5.2: Corridas de Búsqueda Tabú SM para el mapa del valle de México.

PARÁMETROS DE BÚSQUEDA TABÚ SM

- **Número de Iteraciones Fase 1:** 1000
- **Número de Iteraciones Fase 2:** 500 (100 para 500 y 1000 grupos)
- **Número de iteraciones de no mejora antes de perturbación:** 50

5.1.2. Conclusiones de las Pruebas

Los 3 algoritmos muestran puntos fuertes en los 3 aspectos analizados, sin embargo Búsqueda Tabú con Swap Neighborhood muestra ser muy efectivo en todos los aspectos para instancias pequeñas y, para las instancias más grandes, aunque se obtuvieron soluciones peores que PAM extendido y Búsqueda Tabú, el tiempo de cómputo es muy bueno tomando en cuenta la cantidad de iteraciones que lleva a cabo (100000) a comparación de la cantidad de iteraciones para Búsqueda Tabú (5000), en específico en la instancia de 160 grupos el tiempo de cómputo de Búsqueda Tabú con Swap Method es mucho mayor que PAM extendido y Búsqueda Tabú: 5329.231 contra 2249.452 y 2071.007 respectivamente, un incremento de más de el doble, sin embargo, la cantidad de iteraciones es 20 veces mayor en Búsqueda Tabú con Swap Method. De aquí que se sospecha que una calibración adecuada de los parámetros del algoritmo podría lograr superar a los otros 2 algoritmos para grandes instancias.

La solución inicial de los 3 algoritmos se genera de manera aleatoria, este factor influye mucho en la solución final de PAM extendido y en ambas versiones de Búsqueda Tabú se ha comprobado, por las diversas pruebas realizadas durante el desarrollo, que afecta el tiempo en el que encuentra soluciones factibles (Penalidad igual a 0). Así mismo los vecindarios son pieza clave de la metaheurística para lograr encontrar buenas soluciones factibles. Los vecindarios implementados en ambas versiones de búsqueda tabú son prácticamente los utilizados en la literatura para resolver agrupamiento geográfico pero con pequeñas modificaciones para favorecer el objetivo de homogeneidad.

Por lo tanto es lógico pensar que se deben desarrollar e implementar algoritmos propios para el agrupamiento geográfico con restricciones de homogeneidad que apoyen a la metaheurística para encontrar las soluciones factibles homogéneas de manera más rápida. Por ejemplo, como se observa en la Figura 5.4 del comportamiento de Búsqueda Tabú con Swap Method para una corrida de 40 grupos; el primer punto de la gráfica es la solución inicial aleatoria la cual presenta un alto costo debido a su gran penalidad, la cual se reduce gradualmente hasta que después de cierto número de iteraciones sin mejora en el costo de la solución, se realiza una perturbación, que simplemente genera una nueva combinación de centroides aleatoria que nuevamente tiene un alto costo por su alta penalidad. Una manera de mejo-

rar esto es desarrollando un algoritmo que genere una buena solución inicial factible tanto para homogeneidad como para compacidad, otra propuesta podría ser proporcionar al algoritmo la solución inicial, una buena solución factible obtenida por este mismo algoritmo anteriormente u por otro. Esta propuesta fue implementada para un caso de estudio en particular sobre el mosquito del dengue, en el cual se desean colocar ovitrampas en ciertas localidades estratégicas para capturar al mosquito transmisor del dengue. De 130 unidades geográficas se reportaron casos de contagio en 29, que vendrían siendo los centroides y se desea conocer las ovitrampas más cercanas a estos de tal manera que sólo se inspeccionen cierto número de ovitrampas para reducir costos. Por lo tanto se realizó una modificación en PAM que permite al usuario proporcionar los centroides a partir de los cuales se genera una solución (Más información en el Apéndice A).

Otra propuesta es diseñar un vecindario más adecuado para este problema que guíe con mayor facilidad a la búsqueda hacia soluciones factibles para los 2 objetivos deseados, en particular la homogeneidad.

Por otro lado las pruebas preliminares para el mapa del Distrito Federal muestran que para formar grupos pequeños (hasta 100 en las pruebas) el algoritmo de Búsqueda Tabú con Swap Method se comporta de manera aceptable considerando la poca cantidad de iteraciones que se realizaron para poder realizar las pruebas en poco tiempo. Para 2 y 10 grupos el algoritmo logra soluciones factibles, es decir, homogéneas bajo la restricción de las cotas de homogeneidad al 10% y así mismo compactas por las mismas restricciones dadas por el modelo mientras que para 100 grupos solamente 5 unidades geográficas violan la restricción de homogeneidad (penalidad), sin embargo para los casos más grandes examinados, 500 y 1000 grupos, se observa que la penalidad es de 257 y 749 respectivamente, esto muestra la dificultad que presenta el algoritmo para trabajar con instancias grandes y formar gran cantidad de grupos, aunque sin duda, ejecutar la búsqueda tabú por más de 1000 iteraciones sacrificando tiempo de cómputo, generará mejores resultados que los presentados aquí.

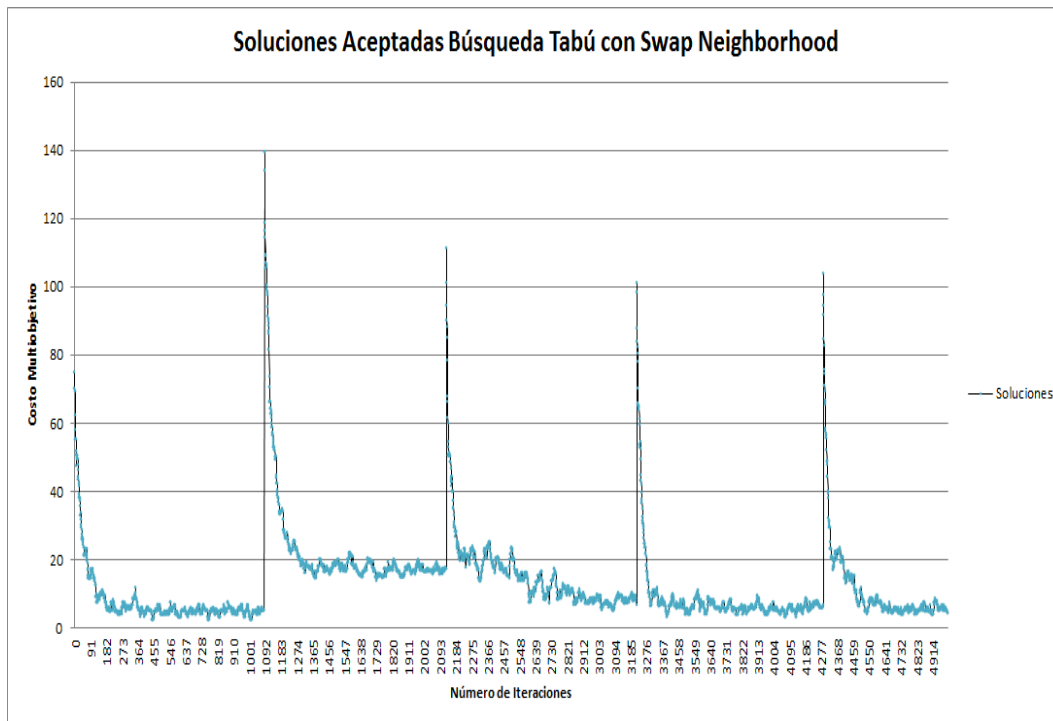


Figura 5.4: Comportamiento de Búsqueda Tabú SM

5.2. Trabajo a Futuro

- Realizar un Diseño de Experimentos para calibrar los parámetros de la metaheurística.
- Permitir al usuario establecer la solución inicial.
- Desarrollar un algoritmo que construya una buena solución inicial compacta y homogénea.
- Definir un vecindario adecuado al problema que guíe a la búsqueda hacia soluciones factibles bajo las restricciones de homogeneidad establecidas.
- Implementar estrategias más avanzadas y complejas de la Búsqueda Tabú, como criterios de aspiración, registro de soluciones élite, reinicio en puntos estratégicos, etc.

- Añadir al programa la capacidad de cargar variables geoestadísticas asociadas a cada unidad geográfica para trabajar la homogeneidad en cuanto a variables.
 - Adoptar la medida de inercia como se presenta en [49] para poder comparar resultados en base a esta medida con algoritmos como el de búsqueda tabú propuesto en [45].

5.3. Conclusiones Finales

Se ha comprobado la eficacia del modelo matemático propuesto para resolver homogeneidad sobre unidades de tipo geográfico, si bien las pruebas realizadas no lograron encontrar soluciones factibles para ciertas instancias, sobre todo las de tamaño considerable, la implementación de Búsqueda Tabú con sus debilidades actuales logra encontrar soluciones factibles y con muy buena compacidad para instancias de poco tamaño sin problemas y también lo logra para la instancia más grande de la tabla 5.1, con esto se demuestra que el modelo funciona adecuadamente y son aspectos de algoritmia y programación los que deben atenderse, por ejemplo, la construcción de una buena solución inicial en relación al algoritmo y con respecto a la programación, es posible añadir más estrategias propias de la búsqueda tabú para mejorar su desempeño (lista de soluciones élite, reinicio de búsqueda) y mejor manejo de memoria para grandes cantidades de unidades geográficas, de este modo es posible lograr una implementación que funcione aún para fuertes restricciones como lo es la tolerancia de homogeneidad al 10% en grandes instancias.

Se desea mejorar la implementación e incorporar la homogeneidad en cuanto a variables geoestadísticas y poder así realizar comparaciones con otros trabajos importantes de la literatura que miden la calidad de las soluciones en base al valor de la inercia de cada grupo formado [49, 45] en lugar de distancias métricas como la euclidiana entre unidades geográficas y centros de grupo como se utiliza en este trabajo.

La implementación desarrollada logra superar a trabajos relacionados como [11] obteniendo soluciones factibles para cotas de homogeneidad más estrictas, sin problemas, para instancias pequeñas y otras de moderado tamaño

en un tiempo de cómputo menor.

Es importante seguir trabajando en esta línea de investigación relativamente nueva en México, en particular la homogeneidad inter clase sobre la cardinalidad de los grupos formados, que traerá grandes beneficios al estudio geoestadístico así como al comercio y la industria.

Apéndices

Apéndice A

Póster en NaBIC 2012

Título: Adaptación de un algoritmo de clustering y enjambre de moscos a un problema de ovitrampas para el vector del mosquito del Dengue

En [12] se ha presentado una extensión de PAM que permite al usuario seleccionar los centroides para generar una solución con respecto a ellos o bien para iniciar la búsqueda a partir de esta solución. Esta necesidad surgió por un problema real: la colocación de ovitrampas en una comunidad para recolectar material del mosquito transmisor del dengue con el fin de estudiarlo. En concreto se desea saber la relación que hay entre los casos registrados de dengue en diversos puntos de la comunidad y las ovitrampas colocadas en otros puntos, en donde los casos son entonces los posibles centroides y las ovitrampas las unidades geográficas que se asignarán al centroide (caso) más cercano y poder determinar así cuáles ovitrampas deben de inspeccionarse de acuerdo a la cercanía con los puntos donde hay registro de contagio de dengue.

Resumen - Los esfuerzos para ubicar las ovitrampas de manera homogénea en una comunidad determinada deben ser redoblados, la realidad es que la colocación de estas depende en gran manera de la voluntad de la comunidad y del uso que le es dado al pedazo de terreno donde la ovitrampa debería ser colocada.

Dado que alrededor del punto de los casos de dengue es más probable la transmisión (moscos infectados) y que ha sido documentado que la trans-

misión vertical (mosco adulto a los huevos), para la vigilancia entomológica y viral de las comunidades endémicas es importante conocer y monitorear, a través del material obtenido por las ovitrampas, las sepas del virus del dengue que circulan entre la población humana y la de los mosquitos.

Si las ovitrampas fueran colocadas de acuerdo a un diseño representativo y homogéneo aleatorio en la comunidad bajo estudio, el sistema de vigilancia descrito arriba consistiría en estudiar las ovitrampas en el rango de vuelo del mosquito (200 metros aproximadamente). Sin embargo, en la realidad este no es el caso, por lo tanto una aproximación probabilista es necesaria para establecer cuáles ovitrampas deberían de ser evaluadas por el sistema de vigilancia sanitaria para tener una mayor probabilidad de éxito en el diagnóstico, haciendo el proceso eficiente con respecto al costo.

En este escenario, de acuerdo con la movilidad del mosquito, un algoritmo de agrupamiento ha sido asociado y adaptado basado en P-medias que promete construir grupos donde el centro de cada grupo es un caso y las ovitrampas más cercanas son asociadas para establecer una configuración sistemática y homogénea de la relación entre casos registrados y las ovitrampas.

Esta elección de centroides puede beneficiar otros algoritmos de particionamiento ya que se podría iniciar una búsqueda para cierta instancia de un problema a partir de una buena solución ya conocida con anterioridad.

Bibliografía

- [1] ANDERBERG, M. *Cluster Analysis for Applications*. Academic Press, 1975.
- [2] ANDREEV, K., AND RÄCKE, H. Balanced graph partitioning. In *Proceedings of the sixteenth annual ACM symposium on Parallelism in algorithms and architectures (2004)*, SPAA '04, pp. 120–124.
- [3] BATTITI, R., BERTOSSI, A., AND CAPPELLETTI, A. Multilevel reactive tabu search for graph partitioning. Tech. rep., Università di Trento, 1999.
- [4] BENLIC, U., AND KAO HAO, J. An effective multilevel tabu search approach for balanced graph partitioning, 2010.
- [5] BERNÁBE, B., OLIVARES, E., RODRÍGUEZ, M., ET AL. El problema de homogeneidad y compacidad en diseño territorial. Congreso Latino-Iberoamericano de Investigación Operativa, Unpublished, Sept. 2012.
- [6] BERNÁBE, B., OSORIO, M., ET AL. An adjusted variable neighborhood search algorithm applied to the geographical clustering problem. *Advances in Computing Science, Research Computing Science 42* (2009), 113–125.
- [7] BERNÁBE, B., RAMÍREZ, J., AND ESPINOSA, J. Evaluación de un algoritmo de recocido simulado con superficies de respuestas. *Revista de Matemáticas Teoría y Aplicaciones 16* (2009), 159–177.
- [8] BERNÁBE, M., AGUIRRE, R., AND LÓPEZ, R. Utilidad del componente principal para caracterizar rasgos poblacionales. In *2do. Congreso Latinoamericano de Matemáticos (UMALCA)* (2004).

- [9] BERNÁBE, M., LÓPEZ, R., AND AGUIRRE, V. Application of non-supervised classification to population data. In *International Conference on Electrical and Electronics Engineering (ICEEE/CIE2004)* (2004).
- [10] BERNÁBE, M., OSORIO, M., ACEVES, R., AND SÁNCHEZ, A. Conglomerados y heurísticas en problemas de regionalización. In *2do Taller Latinoamericano de Investigación de Operaciones* (2007).
- [11] BERNÁBE, M. B., PINTO, D., OLIVARES, E., GONZÁLEZ, R., FLORES, J. L., AND VANOYE, J. R. A hybrid metaheuristic for the partitioning problem with homogeneity constraints on the number of objects. *Lecture Notes in Management Science* 4 (2012), 202–211.
- [12] BERNÁBE, M. B., RODRÍGUEZ, M. A., MARTÍNEZ, R. A., RAMOS, J., AND OLIVARES, E. Adaptation of a clustering algorithm and mosquito swarm to a problem of ovitraps for the dengue mosquito vector (poster). In *2012 World Congress on Nature and Biologically Inspired Computing (NaBIC)* (2012), IEEE.
- [13] BODIN, L. D. A districting experiment with a clustering algorithm. *Annals of the New York Academy of Sciences* 19 (1973), 209–214.
- [14] BOURJOLLY, J. M., LAPORTE, G., AND ROUSSEAU, J. M. Découpage électoral informatisé : application à l’Île de Montréal. *INFOR* 19 (1981), 113–124.
- [15] BOZKAYA, B., ERKUT, E., AND LAPORTE, G. A tabu search heuristic and adaptive memory procedure for political districting. *Europ. J. Opl. Res.* 144 (2003), 12–26.
- [16] CABALLERO, R., LAGUNA, M., MARTÍ, R., AND MOLINA, J. Scatter tabu search for multiobjective clustering problems. *JORS* (2011), 2034–2046.
- [17] CIRINCIONE, C., A., D. T., AND O’ROURKE, T. G. Assessing south carolina’s 1990s congressional districting. *Political Geography* 19 (2000), 189–211.
- [18] DUQUE, J. C., LOBO, R. R., AND CARALT, J. S. Design of homogeneous territorial units: A methodological proposal. Working Papers in

- Economics 115, Universitat de Barcelona. Espai de Recerca en Economia, 2004.
- [19] FIDUCCIA, C., AND MATTHEYSES, R. A linear-time heuristic for improving network partitions. In *Proceedings of the 19th Design Automation Conference* (1982), pp. 171–185.
- [20] FORMAN, S. L., AND YUE, Y. Congressional districting using a tsp-based genetic algorithm. In *Genetic and evolutionary computation conference (GECCO 2003)* (July 2003), pp. 2072–2083.
- [21] GARFINKEL, R. S., AND NEMHAUSER, G. L. Optimal political districting by implicit enumeration techniques. *Management Science* (1970), 495–508.
- [22] GARFINKEL, R. S., AND NEMHAUSER, G. L. Optimal political districting by implicit enumeration techniques. *Management Science* 16 (1970), 495–508.
- [23] GEORGE, J. A., LAMAR, B. W., AND WALLACE, C. A. Political district determination using large-scale network optimization. *Socio-Economic Planning Sciences* 31 (1997), 11–28.
- [24] GLOVER, F., AND LAGUNA, M. *Tabu Search*. Kluwer Academic Publishers, 1997.
- [25] HARTIGAN, J. *Clusterin Algorithms*. John Wiley and Sons, 1973.
- [26] HELBIG, R. E., ORR, P. K., AND ROEDIGER, R. R. Political redistricting by computer. *Comm. ACM* 15 (1972), 735–741.
- [27] HERTZ, A., TAILLARD, E., AND DE WERRA, D. A tutorial on tabu search, 1992.
- [28] HESS, S. W., WEAVER, J. B., SIEGFELDT, H. J., ET AL. Nonpartisan political redistricting by computer. *Operations Research* 13 (1965), 998–1008.
- [29] HOJATI, M. Optimal political districting. *Comput. and Opns. Res.* 23 (1996), 1147–1161.
- [30] INEGI. Marco geostadístico nacional. <http://www.inegi.gob.mx>.

- [31] INEGI. Compendio de criterios y especificaciones técnicas para la generación de datos e información de carácter fundamental, 2010.
- [32] JAIN, S., SWAMY, C., AND BALAJI, K. Greedy algorithms for k-way graph partitioning.
- [33] KARYPIS, G., AND KUMAR, V. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.* 20, 1 (Dec. 1998), 359–392.
- [34] KARYPIS, G., AND KUMAR, V. Multilevel k-way partitioning scheme for irregular graphs. *Journal of Parallel And Distributed Computing* 48 (1998), 96–129.
- [35] KAUFMAN, L., AND ROUSSEEUW, P. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley and Sons, 1990.
- [36] KAUFMAN, L., AND ROUSSEEUW, P. J. Clustering by means of medoids. In *Statistical Data Analysis based on the L1 Norm* (1987), pp. 405–416.
- [37] KERNIGHAN, B. W., AND LIN, S. An efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal* 49 (1970), 291–307.
- [38] KHARROUSHEH, A., ABDULLAH, S., AND NAZRI, M. Z. A. A modified tabu search approach for the clustering problem. *Journal of Applied Sciences* 19 (2011), 3447–3453.
- [39] LEIVA, S. A., AND TORRES, F. J. Una revisión de los algoritmos de partición más comunes de conglomerados: un estudio comparativo. *Revista Colombiana de Estadística* 33, 2 (2010), 321–339.
- [40] LIM, A., AND CHEE, Y.-M. Graph partitioning using tabu search. In *IEEE International Symposium on Circuits and Systems* (1991), pp. 1164–1167.
- [41] MACQUEEN, J. B. Some methods for classification and analysis of multivariate observations. In *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability* (1967), L. M. L. Cam and J. Neyman, Eds., vol. 1, University of California Press, pp. 281–297.

- [42] MARTÍ, R. Procedimientos metaheurísticos en optimización combinatoria. Tech. rep., Departamento de Estadística e Investigación Operativa, Facultad de Matemáticas, Universidad de Valencia, 2002.
- [43] MEHROTRA, A., JOHNSON, E. L., AND NEMHAUSER, G. L. An optimization based heuristic for political districting. *Management Science* 44 (1998), 1100–1114.
- [44] MELIÁN, B., PEREZ, J., ET AL. Metaheurísticas: una visión global. *Revista Iberoamericana de Inteligencia Artificial* 19 (2003), 7–28.
- [45] MURILLO, A. Aplicación de la búsqueda tabú en la clasificación por particiones. *Revista de Ciencias Matemáticas* 21 (2000), 183–194.
- [46] NICKEL, S., SCHRÖDER, M., AND KALCSICS, J. Towards a unified territorial design approach - applications, algorithms and gis integration. *Top (Journal of Operations Research)* 13 (2005), 1–74.
- [47] NYGREEN, B. European assembly constituencies for wales: comparing for methods for solving a political districting problem. *Math Program* 42 (1988), 159–169.
- [48] OSMAN, I., AND KELLY, J. *Meta-Heuristics: Theory and Applications*. Kluwer Academic, 1996.
- [49] PIZA, E., MURILLO, A., AND TREJOS, J. Nuevas técnicas de particionamiento en clasificación automática. *Revista Matemática: Teoría y Aplicaciones* 6 (1999), 51–66.
- [50] RICCA, F., AND SIMEONE, B. Political districting: Traps, criteria, algorithms and tradeoffs. *Ricerca Operativa AIRO* 27 (1997), 81–119.
- [51] RIOJAS, A. Conceptos, algoritmos y aplicación al problema de las n-reinas. Tesis de licenciatura, Universidad Nacional Mayor de San Marcos, 2005.
- [52] ROMERO, D. Formación de unidades primarias de muestreo. No Publicado, 2002.
- [53] ROMERO, D., BURGUETE, J., ET AL. Un enfoque de optimización combinatoria para la construcción de marcos de muestreo de hogares.

Tech. rep., Instituto Nacional de Estadística Geografía e Informática, 2006.

- [54] ROUSEEUW, P. J., HUBERT, M., AND STRUYF, A. Clustering in an object-oriented environment. *Journal of Statistical Software* (1997), 02–10.