

CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL

Laboratorio de Tecnologías de la Información

**Una metaheurística de recocido
simulado para el problema de
maximización del antibandwidth
en grafos**

Tesis que presenta:

Luis Carlos Betancourt Rodríguez

Para obtener el grado de:

**Maestro en Ciencias
en Computación**

Director de la Tesis:
Dr. Eduardo A. Rodríguez Tello

© Derechos reservados por
Luis Carlos Betancourt Rodríguez
2011.

Nota aclaratoria:

Las reglas ortográficas que se siguieron en la escritura de esta tesis son las publicadas en el año 2010 por la Real Academia Española y la Asociación de Academias de la Lengua Española. De acuerdo a estas reglas para las palabras *este, esta, estos, estas, ese, esa, esos, esas, aquel, aquella, aquellos, aquellas* no se toma en consideración la posible ambigüedad entre determinantes demostrativos y pronombres demostrativos. Antiguamente se acentuaban cuando tenían función de pronombre. Las palabras *esto, eso, aquello*, que solo pueden ser pronombres, nunca se han acentuado. La palabra '*solo*' puede funcionar como adjetivo o como adverbio. Antiguamente se acentuaba cuando tenía función de adverbio, equivalente a solamente (Real Academia Española, 2010).

La tesis presentada por Luis Carlos Betancourt Rodríguez fue aprobada por:

Dr. José Torres Jiménez

Dr. Gregorio Toscano Pulido

Dr. Eduardo A. Rodríguez Tello, Director

Cd. Victoria, Tamaulipas, México, 5 de diciembre de 2011.

Para mis padres, los pilares de mi vida.

Agradecimientos

Esta tesis ha sido parcialmente financiada por la beca de estudios de maestría No. 234549, así como los siguientes proyectos CONACyT: Algoritmos para la Canonización de Covering Arrays, No. 99276 y Fondo Mixto CONACyT-Gobierno del Estado de Tamaulipas, No. 51623.

Gracias al Centro de Investigación y de Estudios Avanzados del IPN (CINVESTAV-IPN), Unidad Tamaulipas por todas las facilidades otorgadas.

Agradezco al Dr. Eduardo A. Rodríguez Tello, por su atención y guía durante el desarrollo de esta tesis. Gracias por enseñarme a trabajar con mayor orden y hacerme ver que siempre puedo dar más de lo que creo.

De igual manera les agradezco a todos los investigadores del Centro que me impartieron clases y me enseñaron que siempre hay que buscar la manera de resolver los problemas y no esperar a que te ofrezcan las soluciones.

Gracias a todos mis compañeros, por esos momentos complicados y buenos que compartimos. En especial a Karla, Marco, Miguel, Oscar y Yazmin por brindarme su amistad y hacer más amenos estos años.

El agradecimiento más importante es para mis padres, nunca podré pagarles todo lo que me han dado sin condiciones. Gracias por llevarme por el buen camino y juntos forjar quien soy.

Índice General

Índice General	I
Índice de Figuras	III
Índice de Tablas	V
Índice de Algoritmos	VII
Resumen	IX
Abstract	XI
Nomenclatura	XIII
1. Introducción	1
1.1. Antecedentes	1
1.2. Motivación	3
1.3. Objetivos de la tesis	4
1.3.1. Objetivo general	4
1.3.2. Objetivos específicos	4
1.4. Organización de la tesis	5
2. Estado del arte	7
2.1. Definiciones y notación general	7
2.2. Definición formal del problema de MAG	11
2.3. Complejidad computacional	13
2.4. Métodos de resolución del problema MAG	14
2.4.1. Métodos exactos	14
2.4.1.1. Mallas	15
2.4.1.2. Ciclos	16
2.4.1.3. Caminos	16
2.4.1.4. Toroides	16
2.4.1.5. Árboles completos	17
2.4.1.6. Hipercubos	17
2.4.2. Métodos aproximados	20
2.4.2.1. Algoritmo Genético	20
2.4.2.2. Algoritmo Probabilístico	20
2.4.2.3. Algoritmo Memético	21
2.4.2.4. Procedimientos de búsqueda voraz aleatorizada y adaptativa (GRASP)	22

3. Enfoque de solución propuesto	27
3.1. Recocido simulado	28
3.2. Método para generar la solución inicial	30
3.3. Función de vecindad	31
3.4. Función de aceptación	34
3.5. Temperatura inicial	34
3.6. Longitud de la cadena de Markov	35
3.7. Esquema de enfriamiento	36
3.7.1. Enfriamiento lineal	36
3.7.2. Enfriamiento geométrico	37
3.7.3. Enfriamiento logarítmico	37
3.8. Condición de paro	38
4. Experimentación y resultados	41
4.1. Criterios de comparación	42
4.2. Instancias de prueba	42
4.2.1. Grafos con óptimo conocido	42
4.2.2. Instancias de grafos tipo malla	43
4.2.3. Instancias <i>Rome</i>	43
4.2.4. Instancias <i>Harwell Boeing</i>	43
4.3. Condiciones experimentales	44
4.4. Experimentación preliminar	44
4.4.1. Experimentación preliminar para obtener una solución inicial	44
4.4.2. Experimentación preliminar sobre diferentes vecindarios	47
4.5. Sintonización de parámetros	48
4.6. Experimentación final	51
4.6.1. Resultados sobre instancias del conjunto Rome	52
4.6.2. Resultados de instancias tipo malla	54
4.6.3. Resultados de instancias Harwell-Boeing	56
5. Aplicación práctica del problema de MAG	61
5.1. Introducción a la radiotransmisión	62
5.2. Caso de estudio	62
6. Conclusiones y trabajo futuro	67
6.1. Conclusiones	68
6.1.1. Conjunto Rome	69
6.1.2. Conjunto de grafos tipo malla	69
6.1.3. Conjunto Harwell-Boeing	70
6.2. Trabajo futuro	71

Índice de Figuras

2.1.	Grafo que modela el problema de los puentes de Königsberg	8
2.2.	(a) Grafo simple y (b) Grafo conexo	10
2.3.	Algunos ejemplos de tipos de grafos.	11
3.1.	Estructura de niveles de un grafo con 8 nodos y 7 arcos generada con búsqueda primero en anchura.	31
3.2.	Proceso de etiquetado del grafo de la Figura 3.1(a).	32
4.1.	Comparación del desempeño promedio entre métodos de inicialización aleatorio y BFSi sobre un conjunto de 18 grafos con óptimo conocido.	46
4.2.	Desempeño promedio de diferentes funciones de vecindad combinadas obtenido sobre 5 instancias del conjunto Harwell-Boeing.	48
4.3.	Acercamiento superior de la Figura 4.2.	49
4.4.	Comparación del desempeño entre MAAMP y RS-MAG con respecto al límite inferior teórico sobre 30 grafos del conjunto Rome.	54
4.5.	Gráfica de <i>Tiempo a la mejor solución</i> para la instancia ug1-80.	54
4.6.	Comparación del desempeño entre GRASP+evPR y RS-MAG con respecto al límite inferior teórico sobre 24 grafos tipo malla.	56
4.7.	Comparación del desempeño entre GRASP+evPR y RS-MAG sobre 12 grafos pequeños Harwell-Boeing	58
4.8.	Comparación del desempeño entre GRASP+evPR y RS-MAG sobre 12 grafos grandes Harwell-Boeing.	58
4.9.	Gráfica <i>Tiempo a la mejor solución</i> para la instancia 685_bus.	59
5.1.	Mapa de Cd. Victoria, Tamaulipas.	63
5.2.	Grafo con $AB(G, \varphi) = 1$	64
5.3.	Grafo con $AB(G, \varphi) = 13$	65
6.1.	Resultados obtenidos por RS-MAG sobre el total de instancias estudiadas. . . .	68
6.2.	Resultados obtenidos por RS-MAG sobre los tres conjuntos de instancias analizadas.	69

Índice de Tablas

2.1. Etiquetado óptimo para una malla $P_8 \times P_7$	15
2.2. Etiquetado óptimo para toroide $C_8 \times C_8$	16
2.3. Etiquetado óptimo para toroide $C_7 \times C_7$	17
2.4. Límites teóricos conocidos para el problema de MAG.	19
3.1. Analogía entre sistema físico y un problema de optimización	29
4.1. Comparación entre métodos de inicialización aleatorio y BFSi sobre un conjunto de grafos con óptimo conocido.	45
4.2. Comparación del desempeño entre diferentes combinaciones de vecindarios para RS-MAG.	47
4.3. Posibles parámetros de entrada para RS-MAG.	50
4.4. Matriz transpuesta de $MCA(63; 3, 6, (2^3, 3^2, 7^1))$	50
4.5. 10 mejores casos de prueba ejecutados por RS-MAG sobre grafos del conjunto Harwell-Boeing.	50
4.6. 10 mejores casos ejecutados por RS-MAG sobre grafos del conjunto Rome.	51
4.7. Comparativa de desempeño entre MAAMP y RS-MAG sobre 30 grafos del conjunto Rome.	53
4.8. Comparación del desempeño entre GRASP+evPR y RS-MAG sobre 24 grafos tipo malla.	55
4.9. Comparativa del desempeño entre GRASP+evPR y RS-MAG sobre 24 grafos Harwell-Boenig.	57

Índice de Algoritmos

2.1. Algoritmo memético MAAMP	22
2.2. Algoritmo GRASP	24
3.1. RS-MAG.	29

Una metaheurística de recocido simulado para el problema de maximización del antibandwidth en grafos

por

Luis Carlos Betancourt Rodríguez

Maestro en Ciencias del Laboratorio de Tecnologías de la Información

Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, 2011.

Dr. Eduardo A. Rodríguez Tello, Director

El objeto de estudio en esta tesis es el problema de optimización combinatoria conocido como *Maximización del Antibandwidth en Grafos*. Este consiste en encontrar un etiquetado para los nodos de un grafo de modo tal que se maximice la mínima diferencia absoluta entre las etiquetas de nodos adyacentes. La maximización del antibandwidth al igual que otros problemas de etiquetado de grafos es NP-completo. Actualmente existen reportados métodos que resuelven, de manera óptima y en tiempo polinomial, instancias muy específicas de este problema. Sin embargo, existe aún la necesidad de crear algoritmos que permitan resolver eficientemente el problema para el caso de grafos generales. En esta tesis se desarrolló un algoritmo metaheurístico basado en recocido simulado que permite resolver el problema de *Maximización del Antibandwidth en Grafos* de una manera competitiva con respecto a otras técnicas propuestas en el estado del arte.

La sintonización de parámetros para este algoritmo se realizó mediante una técnica conocida como pruebas de interacción combinatoria que emplea arreglos de cobertura (covering arrays) para representar dichas pruebas. El algoritmo que se propone en este trabajo de tesis se comparó contra dos diferentes enfoques del estado del arte: un algoritmo memético y un algoritmo de tipo GRASP. Se probó sobre un total 78 instancias provenientes de 3 diferentes conjuntos. Para el 66.67% de los casos se obtuvieron resultados superiores a los reportados en la literatura y para otro 17.95% se lograron igualar estas soluciones.

A simulated annealing metaheuristic for the antibandwidth maximization problem for graphs

by

Luis Carlos Betancourt Rodríguez

Master of Science from the Information Technology Labotory

Research and Advanced Studies Center of the National Polytechnic Institute of Mexico, 2011.

Dr. Eduardo A. Rodríguez Tello, Advisor

The purpose of this thesis is to study the combinatorial optimization problem known as Antibandwidth Maximization problem for graphs. It consists in finding a labeling for the nodes of a graph in such a way that the minimum absolute difference between labels of adjacent nodes is maximized. The Antibandwidth Maximization like other graph labeling problems is NP-complete. Cuerrently, there are some reported methods that solve, optimally and in polynomial time, very specific instances of this problem. However, there is still a need to develop algorithms that efficiently solve the problem for the case of general graphs. In this thesis we have developed an algorithm based on the simulated annealing metaheuristic, which permits to solve the antibandwidth maximization problem for graphs competitively with respect to other techniques proposed in the state-of-the-art.

The parameter tuning process, for this algorithm was performed by employing a technique known as combinatorial interaction testing that uses covering arrays for representing these tests. The algorithm proposed in this thesis was tested over three different sets of instances and compared against two different approaches from the state-of-the-art: a memetic algorithm and a GRASP algorithm. It was tested over 78 instances taken from 3 different sets. For 66.67 % of the cases the results were better than those reported in the literature, and for another 17.95 % these solutions were matched.

Nomenclatura

AB	Antibandwidth
MAG	Maximización del Anti-ancho de banda en Grafos
POC	Problema de Optimización Combinatoria
TG	Teoría de Grafos
AG	Algoritmos Genéticos
AM	Algoritmos Meméticos
BPA	Búsqueda Primero en Anchura
ER	Encadenamiento de Rutas
ERM	Encadenamiento de Rutas Mezclado
RS	Recocido Simulado
MCA	Mixed Level Covering Array

1

Introducción

1.1 Antecedentes

La optimización combinatoria es una rama de las matemáticas aplicadas que se sitúa en la intersección de diversos dominios, entre ellos las ciencias computacionales, la investigación de operaciones y la inteligencia artificial. Esta disciplina se enfoca en desarrollar métodos de resolución eficaces para los problemas de optimización combinatoria.

Un Problema de Optimización Combinatoria (POC) consiste en encontrar entre un conjunto finito de soluciones potenciales (llamado espacio de búsqueda) la o las mejores soluciones que verifiquen un criterio particular. Este tipo de problemas aparecen en diversas áreas de la computación como por ejemplo la inteligencia artificial, la investigación de operaciones, la ingeniería de software y la bioinformática. Desde el punto de vista computacional, los POC inducen comúnmente complejidades algorítmicas elevadas debido a que buen número de ellos pertenecen a la clase de problemas NP-difíciles (Garey y Johnson, 1979).

Los POC se pueden formalizar como un par (S, f) , donde el espacio de soluciones S denota el conjunto finito de todas las soluciones posibles y la función de evaluación f es una

función definida como $f : S \rightarrow \mathbb{R}$. Para resolver los POC se pueden usar métodos exactos o aproximados. Los algoritmos exactos permiten explorar sistemáticamente un conjunto de soluciones potenciales por lo que siempre proporcionan la mejor solución (óptima) al problema en cuestión. Sin embargo, tienen la desventaja de consumir demasiado tiempo, incluso para resolver pequeñas instancias. Por esta razón los algoritmos aproximados han ganado una gran popularidad en años recientes.

Los métodos aproximados a su vez se puede dividir en dos categorías: algoritmos a la medida (*tailored algorithms*) y algoritmos generales (*general algorithms*) (Laarhoven y Aarts, 1987). Los algoritmos a la medida son aquellos que fueron diseñados para resolver un problema específico. Los algoritmos generales, en cambio, son fácilmente adaptables a la solución de una amplia variedad de POC.

En los últimos 30 años, una nueva clase de algoritmo aproximado llamado metaheurística emergió y el cual básicamente es un proceso maestro iterativo que guía y modifica las operaciones de las heurísticas subordinadas para producir, de manera eficiente, soluciones de calidad. Las metaheurísticas se pueden clasificar en algoritmos bioinspirados y no bioinspirados. Dentro de los algoritmos bioinspirados se encuentran: colonias de hormigas (Dorigo, 1992), algoritmos genéticos (Holland, 1975) y recocido simulado (Kirkpatrick, Gelatt y Vecchi, 1983). Entre los algoritmos que no están inspirados en procesos naturales se encuentran la búsqueda tabú (Glover, 1986) y la búsqueda local iterativa. El término metaheurística fue introducido por Glover (1986) y se deriva de dos palabras griegas: el verbo *heuriskein* que significa encontrar y el sufijo *meta* que significa más allá de.

Hasta ahora no existe una definición comúnmente aceptada para el término metaheurística, sin embargo, algunos investigadores del campo han tratado de proponer una. La definición de metaheurística más comúnmente aceptada es la siguiente (Blum y Roli, 2003):

Definición 1.1 (Metaheurística). *Una metaheurística es formalmente definida como un proceso de generación iterativo que guía una heurística subordinada, combinando de forma inteligente distintos conceptos para la explotación del espacio de búsqueda, en donde se utilizan estrategias de aprendizaje para estructurar la información con el fin de encontrar de manera*

eficiente, soluciones cercanas a las óptimas.

En esta tesis se propone un algoritmo aproximado de tipo metaheurístico basado en recocido simulado que permite resolver un problema particular de optimización combinatoria conocido como Maximización del Antibandwidth en Grafos (MAG)¹. El cual consiste en encontrar un etiquetado para los nodos de un grafo, de modo tal que se maximice la mínima diferencia absoluta entre las etiquetas de nodos adyacentes.

1.2 Motivación

El problema de MAG ha captado la atención para ser estudiado debido principalmente a su importancia tanto teórica como práctica. En teoría de grafos, el estudio del problema de MAG es importante debido a que existen pocos algoritmos conocidos capaces de resolverlo. Los algoritmos existentes han permitido calcular límites inferiores teóricos para clases particulares de grafos² pequeños (con menos de 50 nodos). Sin embargo, existe aún la necesidad de crear algoritmos que permitan resolver de manera competitiva el problema de MAG para el caso de grafos generales con respecto a las técnicas reportadas en la literatura.

Otro aspecto teórico que motiva este trabajo es el hecho de que el conocimiento adquirido en cuanto a la estructura del problema podría también ser aplicado a la resolución de otros problemas de etiquetado de grafos existentes.

Desde el punto de vista práctico el problema de MAG es importante debido a que existen diversos problemas reales que pueden ser modelados mediante el uso de grafos y cuya solución implica encontrar un etiquetado para los nodos, de modo tal que se maximice la mínima diferencia absoluta entre las etiquetas de nodos adyacentes. Tal es el caso de los problemas de asignación de frecuencias (Hale, 1980), de calendarización de tareas (Leung, Vornberger y Witthoff, 1984), la asignación de canales, el T -coloreo de grafos (Roberts, 1993), así como la ubicación de instalaciones problemáticas (Cappanera, 1999) y peligrosas (Burkard, Donnani, Lin y Rote, 2001, Yixun y Jinjiang, 2003).

¹El término antibandwidth es ampliamente utilizado en la literatura en español.

²Caminos, ciclos, árboles completos, etc.

Más adelante en el Capítulo 5 se presenta una aplicación del mundo real. En la cual se muestra como el problema de asignación de frecuencias puede ser resuelto modelandolo como una instancia del problema de MAG.

1.3 Objetivos de la tesis

1.3.1 Objetivo general

El objetivo general de este trabajo de tesis es contribuir al estado del arte con el análisis e implementación de un algoritmo metaheurístico que permita resolver el problema de MAG para grafos generales de manera altamente competitiva con respecto a las técnicas de referencia existentes.

1.3.2 Objetivos específicos

Los objetivos particulares para este trabajo de investigación son:

- Estudiar extensivamente los diferentes métodos de solución propuestos en la literatura
- Implementar una metaheurística para resolver, el caso general, del problema de MAG
- Sintonizar los componentes de la metaheurística propuesta, con la finalidad de mejorar su desempeño tanto en calidad de solución entregada como en tiempo de cómputo consumido
- Efectuar un estudio comparativo de la metaheurística propuesta con respecto a las mejores técnicas estudiadas en la literatura

1.4 Organización de la tesis

A continuación se presenta una breve descripción del contenido de los siguientes capítulos:

- **Capítulo 2:** Se presentan algunos conceptos básicos de la teoría de grafos para introducir de manera formal el problema de MAG. La complejidad computacional del problema es analizada y se describen los métodos existentes mediante los cuales se ha intentado resolver el problema de MAG
- **Capítulo 3:** El enfoque propuesto en esta tesis para resolver el problema de MAG es descrito en este capítulo. Se detalla cada componente utilizado en la metaheurística de recocido simulado.
- **Capítulo 4:** Las etapas de la experimentación de esta investigación son descritas, así como los criterios de comparación y los conjuntos de instancias utilizadas. También se presenta un método de sintonización del algoritmo mediante una estructura combinatoria conocida como arreglo de cobertura. Dicha metodología ayuda a disminuir el número de pruebas experimentales. Además se presenta un análisis comparativo entre los resultados obtenidos por nuestro algoritmo y aquellos reportados en la literatura
- **Capítulo 5:** Se plantea un caso de estudio donde el problema de asignación de frecuencias de radio puede ser resuelto como una instancia del problema de MAG. Este problema consiste en asignar las frecuencias de emisión a un conjunto de antenas de radio de forma tal que se eviten las interferencias ocasionadas por antenas emisoras cercanas. Dicho problema puede ser resuelto si se encuentra una solución de calidad al problema de MAG.
- **Capítulo 6:** Se presentan las conclusiones a las que se llegan en este trabajo de investigación y se abordan algunas posibilidades de trabajo futuro

2

Estado del arte

En este capítulo se presentan de forma resumida algunos de los conceptos básicos que conciernen a la teoría de grafos así como la definición formal del problema MAG y su complejidad computacional. También, se describen brevemente algunos de los métodos reportados en la literatura para resolver el problema que da origen a este trabajo de tesis.

2.1 Definiciones y notación general

Se considera que la Teoría de Grafos (TG), comenzó en el siglo XVIII, cuando Leonhard Euler demostró que no era posible atravesar los siete puentes de la ciudad rusa de Königsberg (situada actualmente en Kaliningrado) exactamente una vez y volver al punto de partida. El grafo que modela este problema se muestra en la Figura 2.1.

La TG es una importante área de las matemáticas que también se utiliza en otras disciplinas como la biología (obtener una clasificación jerárquica del conjunto de todas las proteínas conocidas), la química (modelado de las estructuras, donde los átomos son representados por nodos y los enlaces covalentes por los arcos), ciencias sociales (modelado de relaciones, donde se sustituye los nodos por los actores sociales y verifica la posición, centralidad e importancia

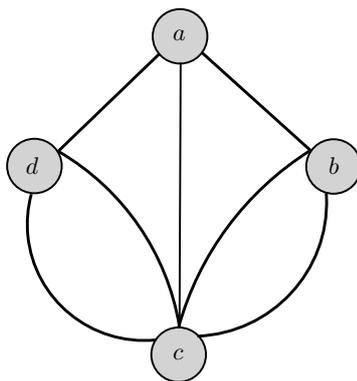


Figura 2.1: Grafo que modela el problema de los puentes de Königsberg

de cada actor dentro del grafo) o aplicaciones industriales (diseño de circuitos VLSI). Además, es considerada como uno de los instrumentos más eficaces para resolver problemas discretos que plantea la Investigación de Operaciones.

En general, un grafo se utiliza para modelar muchas situaciones prácticas, donde los objetos involucrados tienen interacción. Por ejemplo, los componentes de un circuito, las vías del ferrocarril, árboles genealógicos, etc.

En el resto de la sección se presentan una serie de definiciones y notaciones generales que conciernen a la teoría de grafos y que utilizamos para introducir formalmente el problema de estudio de esta tesis.

Definición 2.1 (Grafo no dirigido). *Un grafo no dirigido $G(V, E)$ es una estructura que consta de un conjunto de elementos llamados nodos y un conjunto de pares de nodos que son llamados arcos. Para denotar los conjuntos de nodos y arcos de un grafo, podemos utilizar las anotaciones $V(G)$ y $E(G)$, respectivamente.*

Definición 2.2 (Orden de un grafo). *El orden de un grafo $G = (V, E)$ es el número de nodos $|V| = n$ que componen el grafo.*

Definición 2.3 (Nodos adyacentes). *Dos nodos u y v de un grafo G se llaman adyacentes, si $\{u, v\}$ es un arco de G . Si todos los pares de nodos de un grafo G son adyacentes entonces es un grafo completo.*

Definición 2.4 (Arco incidente). *Un arco es incidente a un nodo v si v es uno de sus*

extremos.

Definición 2.5 (Arco múltiple). *Un arco es múltiple si existe más de un arco que une los mismos nodos.*

Definición 2.6 (Lazo). *Un arco e es llamado lazo si sus dos nodos son idénticos ($e = \{v, v\} \in E$), es decir, un arco que une un nodo a sí mismo. También se le llama arco reflexivo.*

Definición 2.7 (Grafo simple). *Un grafo simple $G = (V, E)$ es un grafo sin lazos o arcos múltiples (ver Figura 2.2(a)).*

Definición 2.8 (Grado de un nodo). *Sea v un nodo de un grafo $G = (V, E)$. El grado de v , denotado $\delta(v)$, es el número de arcos de E incidentes a v .*

Definición 2.9 (Secuencia de grados). *Una secuencia de grados o lista de grados es una secuencia de números, los cuales son los grados de todos los vértices de un grafo $G = (V, E)$.*

Definición 2.10 (Grado de un grafo). *El grado mínimo de $G = (V, E)$, se denota $d(G)$, es el valor mínimo de la secuencia de grados. El máximo grado de G , denotado por $D(G)$, es el valor máximo de la secuencia de grados.*

Definición 2.11 (Grafo vacío). *Es un grafo $G = (V, 0)$ que no tiene arcos por lo tanto sus nodos son aislados.*

Definición 2.12 (Grafo nulo). *Un grafo nulo es un grafo de orden cero.*

Definición 2.13 (Camino). *Un camino entre dos nodos u y v de un grafo $G = (V, E)$ es una secuencia $\{\{v_1, v_2\}, \{v_2, v_3\}, \dots, \{v_{k-1}, v_k\}\}$ de arcos de E tal que $u = v_1$ y $v = v_k$. El número de arcos del camino se conoce como longitud del camino.*

Definición 2.14 (Distancia entre nodos). *La distancia entre dos nodos de un grafo $G = (V, E)$ es la longitud del camino más corto que los conecta.*

Definición 2.15 (Grafo conexo). *Un grafo se dice conexo si existe un camino entre todos los pares de nodos de un grafo (ver Figura 2.2(b)).*

Definición 2.16 (Diametro de un grafo). *El diámetro θ de un grafo es la distancia más grande entre dos nodos de un grafo $G = (V, E)$.*

Definición 2.17 (Hipercubo). *Un grafo hipercubo Q_n es un grafo de 2^n nodos y sus arcos forman un cubo n -dimensional. (ver Figura 2.3(b)).*

Definición 2.18 (Malla). *La estructura de una malla es definida como el producto cartesiano de dos caminos no dirigidos de tamaños m y n : $P_m \times P_n$ (ver Figura 2.3(a)).*

Definición 2.19 (Toroide). *Las mallas n -dimensionales son definidas como $P_{m_1} \times P_{m_2} \times \dots \times P_{m_n}$ donde P_{m_i} es un camino no dirigido de longitud m_i . Los grafos toroidales son construídos de la misma forma que las mallas con el uso de ciclos C_i dentro de los caminos P_i , para $i = 1, 2, \dots, n$.*

Definición 2.20 (Completo k -ario). *Los grafos completos son denotados por $T_{k,h}$. Este tipo de grafo es un árbol en donde la raíz y cada nodo interno tienen k hijos y cada hoja está a una distancia h de la raíz.*

Definición 2.21 (Bipartito). *Es un grafo no dirigido cuyos nodos se pueden separar en conjuntos disjuntos y los arcos siempre unen nodos de conjuntos diferentes (ver Figura 2.3(c)).*

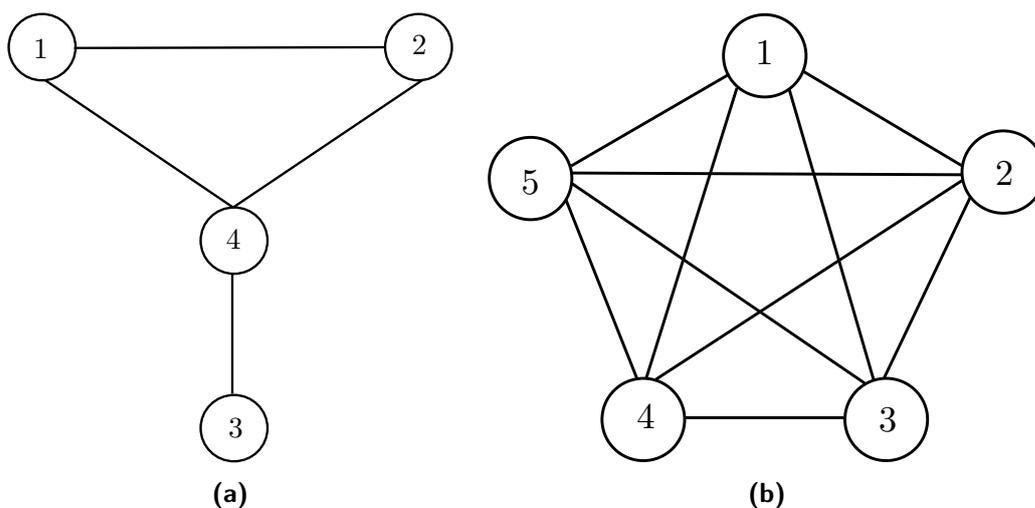


Figura 2.2: (a) Grafo simple y (b) Grafo conexo

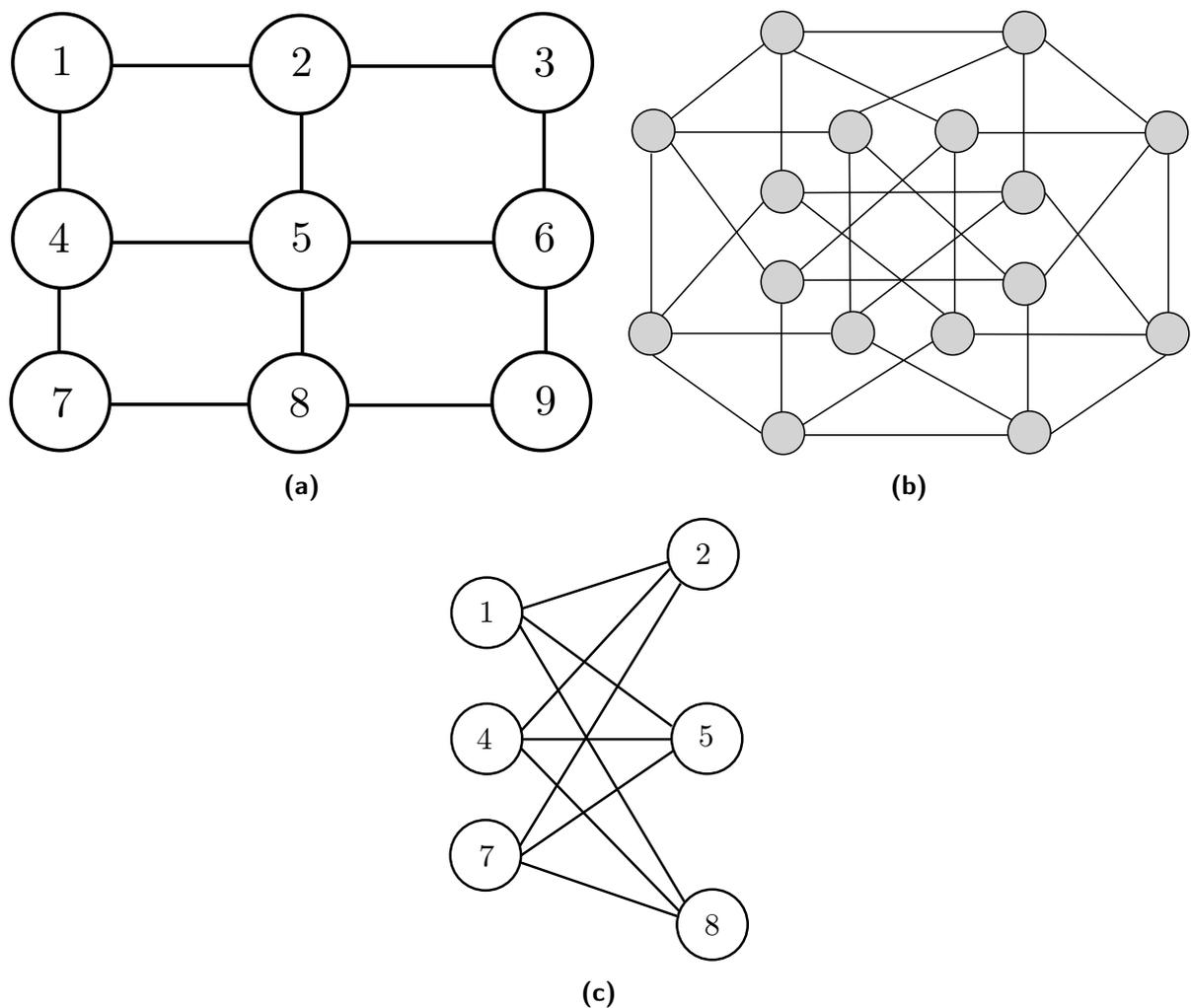


Figura 2.3: Algunos ejemplos de tipos de grafos.

2.2 Definición formal del problema de MAG

Antes de introducir formalmente el problema de MAG, se presentan algunos conceptos que son necesarios para definir el problema al cual se enfoca esta tesis.

Definición 2.22 (Etiquetado de un grafo). *Un etiquetado de un grafo $G = (V, E)$ de orden n es una función biyectiva de la forma $\varphi : V \rightarrow \{1, 2, \dots, n\}$, que a cada vértice del grafo asocia un número entero distinto entre 1 y n . Para designar la etiqueta de un vértice u utilizaremos la notación $\varphi(u)$.*

El antibandwidth es un parámetro fundamental que interviene en la formulación de diversos problemas modelados a través de grafos y se define como sigue:

Definición 2.23 (Antibandwidth). Sea $G = (V, E)$ un grafo no dirigido de orden n , y $\varphi : V \rightarrow \{1, 2, \dots, n\}$ un etiquetado para G . El antibandwidth AB de G para φ está definido con la ecuación siguiente:

$$AB(G, \varphi) = \min\{|\varphi(i) - \varphi(j)| : \{i, j\} \in E\} \quad (2.1)$$

Partiendo de este concepto, ahora se puede definir el problema de MAG.

Definición 2.24 (Problema de MAG). Sea $G = (V, E)$ un grafo no dirigido de orden n . El problema de MAG consiste en encontrar un etiquetado φ^* para el cual el antibandwidth $AB(G, \varphi^*)$, sea máximo:

$$AB(G, \varphi^*) = \max\{AB(G, \varphi) : \varphi \in \mathcal{L}\}, \quad (2.2)$$

donde \mathcal{L} es el conjunto de todos los etiquetados posibles.

Se observa que un etiquetado puede ser visto también como una *permutación*. De esta forma, es fácil verificar que un etiquetado φ puede ser transformado en otro etiquetado φ' aplicando como máximo $n - 1$ operaciones de intercambio entre dos etiquetas.

Debido a que hay $n!/2$ posibles etiquetados¹ para un grafo con n nodos, el problema de MAG es un problema altamente combinatorio.

El problema de MAG fue presentado originalmente por Leung *et al.* (1984) en relación con problemas de calendarización de tareas en varios procesadores, quienes también demostraron que pertenece a la clase de problemas NP-completos. Este problema también se puede formular como el dual del problema de minimización del ancho de banda en grafos.

El término antibandwidth se introdujo por Raspaud *et al.* (2009). Otros nombres usados en la literatura para referirse al antibandwidth son el problema de Separación de Números

¹Por que cada etiquetado puede ser invertido para obtener el mismo valor del antibandwidth.

(*Separation Number problem*) (Leung, Vornberger y Witthoff, 1984), y el problema del Ancho de Banda Dual (*Dual Bandwidth problem*) (Yixun y Jinjiang, 2003).

Actualmente existen técnicas que resuelven instancias muy específicas de este problema. Por ejemplo, Raspaud *et al.* (2009) propusieron soluciones exactas para grafos conocidos como mallas, toroides e hipercubos. En dicha investigación se desarrolló por primera vez una heurística para realizar la aproximación de los límites inferiores en estos tipos de grafos. Uno de los problemas a los que se enfrentaron los autores fue el no tener resultados previos obtenidos con otras técnicas para realizar sus comparaciones.

Por otra parte Dobrev *et al.* (2009) se enfocaron en el estudio de grafos de tipo hamiltoniano. En específico analizaron los límites teóricos sobre dicho tipo de grafos. Además lograron realizar un algoritmo que ajusta los límites superiores en grafos hamiltonianos.

Algunas relaciones básicas entre el antibandwidth y otras características de los grafos tales como los grados mínimos y máximos y el número cromático fueron derivados por Miller y Pritikin (1989). Calamoneri *et al.* (2006) realizaron una investigación, en la cual se obtuvieron resultados exactos para árboles completos k -arios. Para llegar a dichos resultados se basaron en sus investigaciones previas realizadas sobre árboles completos binarios.

Otros problemas que se pueden modelar como una instancia del problema de MAG son el problema de asignación de frecuencias (Hale, 1980), la asignación de canales, el T -coloreo de grafos (Roberts, 1993), la ubicación de instalaciones problemáticas (Cappanera, 1999) y peligrosas (Burkard, Donnani, Lin y Rote, 2001, Yixun y Jinjiang, 2003).

2.3 Complejidad computacional

La teoría de la complejidad computacional es la rama de la teoría de la computación que estudia, de manera teórica, la complejidad inherente a la resolución de un problema computable. Los recursos comúnmente estudiados son el tiempo (mediante una aproximación al número y tipo de pasos de ejecución de un algoritmo para resolver un problema) y el espacio (mediante una aproximación a la cantidad de memoria utilizada para resolver un problema). La mayor

parte de los problemas en teoría de la complejidad tienen que ver con los problemas de decisión.

Los problemas de decisión se clasifican en conjuntos de complejidad comparable llamados clases de complejidad.

La clase de complejidad P es el conjunto de los problemas de decisión que pueden ser resueltos en una máquina determinista en tiempo polinómico, lo que corresponde intuitivamente a problemas que pueden ser resueltos aún en el peor de sus casos.

La clase de complejidad NP es el conjunto de los problemas de decisión que pueden ser resueltos por una máquina no determinista en tiempo polinómico. Esta clase contiene muchos problemas que se desean resolver en la práctica, incluyendo el problema de satisfactibilidad booleana (SAT) y el problema del agente viajero, etc. Todos los problemas de esta clase tienen la propiedad de que su solución puede ser verificada efectivamente.

MAG es un problema que pertenece a la clase NP-Completo (Leung *et al.*, 1984). El problema MAG es conocido también como el problema de separación el cual puede reducirse al problema del camino hamiltoniano (*hamiltonian path problem*) el cual es NP-Completo. En dicho problema tenemos un grafo $G = (V, E)$ y se nos pide determinar si hay un camino que vaya a través de cada uno de sus nodos pertenecientes a V exactamente una vez. Papadimitriou y Steiglitz (1998) presentan una demostración ayudándose del problema 3-SAT.

2.4 Métodos de resolución del problema MAG

Debido a la importancia práctica y teórica de los problemas de optimización combinatoria, se han desarrollado diversos métodos para tratar de resolverlos. Estos se pueden clasificar en dos grandes grupos: métodos exactos y aproximados.

2.4.1 Métodos exactos

Los métodos exactos dan garantía de encontrar para cada instancia de un problema de tamaño finito, una solución óptima si existe, y determinar con seguridad si no existe, cuando no hay restricción de tiempo (Papadimitriou y Steiglitz, 1998).

La mayoría del trabajo previo sobre el problema de MAG se enfoca en el estudio teórico de sus propiedades para encontrar soluciones óptimas para casos específicos.

Existen algoritmos exactos que resuelven el problema en tiempo polinómico para cierto tipo de grafos: caminos, ciclos, árboles, grafos completos y bipartitos completos, mallas, hipercubos y toroides (Miller y Pritikin, 1989, Yixun y Jinjiang, 2003, Yao, Ju y Xiaoxu, 2003, Calamoneri, Missini, Török y Vrt'o, 2006, Török, 2007, Raspaud, Schröder, Sykora, Török y Vrt'o, 2009, Wang, Wu y Dimitrescu, 2009).

2.4.1.1. Mallas

Existen investigaciones que reportan soluciones exactas para el problema del antibandwidth en grafos. A continuación las fórmulas para encontrar los límites inferior y superior del antibandwidth.

La construcción del límite inferior es mostrada en Miller y Pritikin (1989), se obtiene mediante la Fórmula 2.3 para cualquier $m \geq n \geq 2$

$$AB(P_m \times P_n) = \left\lceil \frac{(m-1)n}{2} \right\rceil \quad (2.3)$$

En la tabla 2.1 se presenta un etiquetado óptimo para una malla $P_8 \times P_7$.

38	13	45	20	51	25	55	28
7	39	14	46	21	52	26	56
33	8	40	15	47	22	53	27
3	34	9	41	16	48	23	54
30	4	35	10	42	17	49	24
1	31	5	36	11	43	18	50
29	2	32	6	37	12	44	19

Tabla 2.1: Etiquetado óptimo para una malla $P_8 \times P_7$

Raspaud *et al.* (2009) comentan que la malla es un grafo bipartito con $|V_1| = \lceil mn/2 \rceil$, $|V_2| = \lfloor mn/2 \rfloor$. El límite superior se define en un intervalo $I = [L, R]$ donde:

$$L = \left(\left\lfloor \frac{n}{2} \right\rfloor - 1 \right)^2 + 1 \text{ y } R = \left\lceil \frac{mn}{2} \right\rceil - \left\lfloor \frac{n}{2} \right\rfloor \left\lfloor \frac{n-1}{2} \right\rfloor - 1$$

2.4.1.2. Ciclos

Resultados exactos para los ciclos se obtienen mediante la Ecuación 2.4 (Yixun y Jinjiang, 2003). Sea C_n un ciclo de longitud n :

$$AB(C_n) = \left\lceil \frac{n}{2} \right\rceil - 1 \quad (2.4)$$

2.4.1.3. Caminos

Los grafos tipo camino (*path*) son estudiados también y la Fórmula 2.5 permite obtener resultados exactos (Yixun y Jinjiang, 2003). Sea P_n un camino de longitud n :

$$AB(P_n) = \left\lfloor \frac{n}{2} \right\rfloor \quad (2.5)$$

2.4.1.4. Toroides

Para hacer un calculo correcto del límite inferior se debe utilizar la fórmula adecuada según la paridad de n (Raspaud *et al.*, 2009). Para n par tenemos:

$$AB(C_n \times C_n) = \frac{n(n-2)}{2} \quad (2.6)$$

Un ejemplo de etiquetado óptimo de un toroide para n par se muestra en la Tabla 2.2

35	11	51	27	63	23	47	7
3	43	19	59	31	55	15	39
34	10	50	26	62	22	46	6
2	42	18	58	30	54	14	38
33	9	49	25	61	21	45	5
1	41	17	57	29	53	13	37
32	8	48	24	60	20	44	4
0	40	16	56	28	52	12	36

Tabla 2.2: Etiquetado óptimo para toroide $C_8 \times C_8$

Cuando n impar entonces se utiliza la siguiente ecuación:

$$AB(C_n \times C_n) = \frac{(n-2)(n+1)}{2} \quad (2.7)$$

Un ejemplo de un etiquetado óptimo se presenta en la Tabla 2.3

27	48	20	41	13	34	6
47	19	40	12	33	5	26
18	39	11	32	4	25	46
38	10	31	3	24	45	17
9	30	2	23	44	16	37
29	1	22	43	15	36	8
0	21	42	14	35	7	28

Tabla 2.3: Etiquetado óptimo para toroide $C_7 \times C_7$

2.4.1.5. Árboles completos

En la investigación de Calamoneri *et al.* (2006) se extendieron los resultados de los árboles completos (*complete k-ary trees*). La demostración la dividieron en dos partes de acuerdo a la paridad de k . El límite inferior es calculado como se muestra enseguida:

Para el caso de k par, $k \geq 4$

$$AB(T(k, n)) = \frac{n+1-k}{2} \quad (2.8)$$

Para k impar, $k \geq 3$ y $h = 2$:

$$AB(T(k, n)) = \frac{k^2+1}{2} \quad (2.9)$$

2.4.1.6. Hipercubos

Raspud *et al.* (2009) comentan que los nodos del hipercubo se representan mediante cadenas binarias de longitud n y dos nodos son adyacentes si y solo si las cadenas se diferencian en una posición. En dicha investigación muestran como obtener el límite inferior

con la ecuación:

$$AB(Q_n) = 2^{n-1} - \frac{2^{n-1}}{\sqrt{2\Pi n}}(1 + o(1)) \quad (2.10)$$

En la Tabla 2.4 se sintetizan las investigaciones y aportaciones realizadas a lo largo de los últimos años en torno a los límites teóricos conocidos para el problema de MAG para diferentes tipos de grafos.

Tipo de grafo	Límite Superior	Límite Inferior	Autor
Mallas	$I = [(\lfloor \frac{n}{2} \rfloor - 1)^2, \lceil \frac{mn}{2} \rceil - \lfloor \frac{n}{2} \rfloor \lfloor \frac{n-1}{2} \rfloor - 2]$	$AB(P_m \times P_n) = \lceil \frac{(m-1)n}{2} \rceil$	Raspaud <i>et al.</i> (2009)
Ciclos		$AB(C_n) = \lfloor \frac{n}{2} \rfloor - 1$	Yixun y Jinjiang (2003)
Caminos		$AB(P_n) = \lfloor \frac{n}{2} \rfloor$	Yixun y Jinjiang (2003)
Toroides		Para n par: $AB(C_n \times C_n) = \frac{n(n-2)}{2}$ Para n impar: $AB(C_n \times C_n) = \frac{(n-2)(n+1)}{2}$	Raspaud <i>et al.</i> (2009)
Árboles completos		Para k par: $AB(T(k, n)) = \frac{n+1-k}{2}$ Para k impar: $AB(T(k, n)) = \frac{k^2+1}{2}$	Calamoneri <i>et al.</i> (2006)
Hipercubos		$AB(Q_n) = 2^{n-1} - \frac{2^{n-1}}{\sqrt{2\pi n}}(1 + o(1))$	Raspaud <i>et al.</i> (2009)

Tabla 2.4: Límites teóricos conocidos para el problema de MAG.

2.4.2 Métodos aproximados

En contraposición a los métodos exactos, los metaheurísticos no garantizan la obtención de soluciones óptimas. Sin embargo, suelen obtener soluciones satisfactorias en tiempos de procesamiento generalmente polinomiales.

2.4.2.1. *Algoritmo Genético*

Los principios básicos de los Algoritmos Genéticos (AG) fueron establecidos por Holland (1975). Son métodos adaptativos que pueden usarse para resolver problemas de búsqueda y optimización. Se basan en el proceso genético de los organismos vivos. A lo largo de las generaciones, las poblaciones evolucionan en la naturaleza de acorde con los principios de la selección natural y la supervivencia de los más fuertes (Darwin, 1859).

Los AG usan una analogía directa con el comportamiento natural. Trabajan con una población de individuos, cada uno de los cuales representa una solución factible a un problema. Cuanto mayor sea la adaptación de un individuo al problema, mayor será la probabilidad de que el mismo sea seleccionado para reproducirse, cruzando su material genético con otro individuo seleccionado de igual forma. De esta manera se produce una nueva población de posibles soluciones, la cual reemplaza a la anterior y verifica la propiedad de que contiene una mayor proporción de buenas características en comparación con la población anterior.

2.4.2.2. *Algoritmo Probabilístico*

Un algoritmo probabilista (o probabilístico) basa su resultado en la toma de algunas decisiones al azar, de tal forma que en promedio, obtiene una buena solución al problema planteado para cualquier distribución de los datos de entrada. Uno de los algoritmos más representativos es el de Montecarlo. Fue bautizado así por su clara analogía con los juegos de ruleta de los casinos. Este método puede dar la respuesta correcta o la respuesta errónea (con probabilidad baja)(Krauth, 2006).

2.4.2.3. Algoritmo Memético

La idea esencial de hacer una hibridación de Algoritmos Genéticos (AG) con métodos de búsqueda local es aprovechar al máximo la capacidad operativa de los métodos basados en vecindarios, y la capacidad de exploración de los AG. Los Algoritmos Meméticos (AM) siguen dicho enfoque (Moscato, 1999). En diferentes contextos y situaciones, los AM también se conocen como la búsqueda local genética o algoritmos genéticos híbridos.

Recientemente se desarrolló un AM (Bansal y Srivastava, 2011) llamado *MAAMP* (ver Algoritmo 2.1), cuya implementación emplea la técnica de Búsqueda Primero en Anchura (*Breadth First Search*, BPA) sobre el grafo para generar una estructura de niveles. Esta es empleada posteriormente por una nueva heurística llamada *LAH* para etiquetar los nodos. El proceso principal consta de tres partes que son el proceso de selección, reproducción y mutación. En la etapa de selección se eligen buenas soluciones sobre una población de padres. Enseguida las soluciones seleccionadas participan en el proceso de reproducción basado en BPA para generar una población de hijos y enseguida se inicia el proceso de mutación con probabilidad pm . Las soluciones nuevas producidas son empleadas para actualizar la población. Este proceso se repite hasta que no se haya encontrado mejora sobre la solución actual.

Los resultados presentados, demuestran que *MAAMP* fue capaz de resolver con éxito diversos tipos de grafos con entre 50 y 1000 nodos. Sin embargo, los autores también reportan que en el caso de árboles no balanceados su algoritmo no es capaz de obtener buenos resultados.

Algoritmo 2.1 Algoritmo memético MAAMP

Entrada: Grafo G , Tamaño de población TP , número máximo de iteraciones $maxIter$, probabilidad de mutación pm
 Inicializar $TP, numIter, pm$
 $mejorSolución \leftarrow 1$
 $LAH(padres)$ // Genera población inicial $padres$ aplicando LAH
repetir
 $torneoBinario(padres)$
 $hijos \leftarrow LAH(padres)$ // Genera población $hijos$ aplicando LAH a la población $padres$
 $mutación(hijos, pm)$
 $actualizar(mejorSolución)$
 para $i=1$ hasta TP **hacer**
 si $\max\{AB(G, padres[i])\} < \max\{AB(G, hijos[i])\}$ **entonces**
 $padres[i] := hijos[i]$
 fin si
 fin para
hasta que no haya mejora en $mejorSolución$ durante $maxIter$ generaciones
Regresa: Mejor Solución

2.4.2.4. Procedimientos de búsqueda voraz aleatorizada y adaptativa (GRASP)

GRASP (Greedy Randomized Adaptive Search Procedures) es una metaheurística simple que combina heurísticas constructivas y búsqueda local (Feo y Resende, 1995). Su estructura general se puede observar en el Algoritmo 2.2 tomado de (Blum y Roli, 2003). GRASP es un procedimiento iterativo, que se compone de dos fases: la construcción de una solución y su posterior mejora. La mejor solución es devuelta hasta que el proceso de búsqueda local termina.

El mecanismo de la construcción de la solución es caracterizado principalmente por: una heurística dinámica constructiva y la aleatoriedad. Asumiendo que una solución parcial s consiste en un subconjunto de elementos (componentes de la solución), la solución es construida paso a paso añadiendo un elemento a la vez. La elección del siguiente elemento es hecha de manera aleatoria de la lista de candidatos. Después se evalúa la solución parcial. La lista de candidatos, es nombrada *lista restringida de candidatos (LRC)* y está compuesta con los mejores η elementos. La mejor solución es actualizada en cada iteración.

La segunda fase del algoritmo es un proceso de búsqueda local, el cual puede ser un

algoritmo de búsqueda local básico o algunos otros más avanzados como recocido simulado o búsqueda tabú.

En el año 2009 se realizó una investigación (Duarte *et al.*, 2011) en la cual diseñaron un algoritmo de búsqueda local básico y dos algoritmos constructivos llamados *C1* y *C2*. El algoritmo *C1* implementa una heurística GRASP típica en donde cada solución inicial es evaluada por una función voraz para construir una lista de candidatos. Posteriormente se elige aleatoriamente un candidato de la lista generada. Este procedimiento trata de asignar etiquetas pequeñas al nodo seleccionado y etiquetas más grandes a sus nodos adyacentes. El algoritmo *C2* trabaja de forma similar solo que la elección aleatoria se realiza antes de la evaluación con la función voraz, en otras palabras escoge candidatos aleatorios y sobre esa lista generada se evalúa cada elemento.

En dicha investigación agregaron a los algoritmos encadenamiento de rutas (ER). Los autores utilizaron la variante de encadenamiento de rutas mezclado (ERM) (Glover, 1996; Ribeiro and Rosseti, 2002). ERM comienza con una solución inicial y gradualmente se va transformando en otra solución por el intercambio de elementos, generando soluciones intermedias. Esta lógica se mantiene para ambas soluciones hasta que se unen por una sola ruta.

GRASP+evPR es el nombre del algoritmo que presentan. Aplica en cada iteración las fases de construcción y mejora del GRASP, así como también el método de ER para obtener un conjunto de soluciones *ES*. Una vez que terminan las iteraciones pre-establecidas se lleva a cabo una fase de post-procesamiento en donde se aplica una vez más ER a pares de soluciones que pertenezcan al conjunto *ES*. Las soluciones obtenidas se consideran candidatas para ingresar a *ES*. Después ER es aplicado de nuevo tantas veces como nuevas soluciones entran a *ES*. Así es como el conjunto *ES* evoluciona y se diversifica.

Los autores evaluaron los algoritmos con dos diferentes conjuntos de instancias. Un conjunto de grafos tipo malla (descrito en la Sección 4.2.2) y un conjunto de grafos pertenecientes al conjunto de prueba Harwell-Boeing (descritos en la Sección 4.2.4). Para los grafos de tipo malla obtuvieron buenos resultados, en algunos casos su algoritmo encuentra

el mejor etiquetado posible para resolver el problema MAG. Sin embargo, en la mayoría de las instancias no lo logran. Por ejemplo la instancia *can_715* que tiene la mayor cantidad de nodos perteneciente al conjunto Harwell-Boeing, su valor óptimo teórico es de 357 mientras que con su algoritmo llegan a 121, es decir, una diferencia de 236 unidades.

Para el conjunto de grafos Harwell-Boeing, los autores obtienen buenos resultados para los grafos pequeños (entre 30 y 100 nodos) en donde para algunos casos consiguen obtener el mejor etiquetado posible para resolver el problema MAG. Para las instancias más grandes no presentan resultados cercanos a los valores óptimos. Dichos valores son conocidos por construcción como el caso de las instancias tipo malla, es decir, por medio de fórmulas publicadas en las investigaciones que conforman el estado del arte del problema de MAG. Para el conjunto Harwell-Boeing no existe una fórmula en la literatura, los valores óptimos son obtenidos mediante algoritmos exactos.

Algoritmo 2.2 Algoritmo GRASP

```

s ← 0 // s denota una solución parcial
η ← DeterminaLongitudDeRCL // Definición de la longitud de la lista RCL
mientras la solución no está completa hacer
  RCLη ← GenerarListaDeCandidatos(s)
  s ← s ∪ {x} // x es un elemento de LRC
  FunciónGreedy(s)
fin mientras
Regresa: Mejor Solución
  
```

Resumen del capítulo

En este capítulo se presentó una introducción de algunos conceptos básicos de la teoría de grafos, enfatizando la importancia de los problemas teóricos y prácticos derivados de esta rama de las matemáticas, en especial de problemas NP. Como parte de este capítulo también se presenta formalmente el problema que da origen a esta investigación así como la complejidad computacional del mismo. Además se mostró una clasificación de los algoritmos desarrollados para resolver el problema MAG.

En el próximo capítulo se detallan aspectos del enfoque de solución que se propone para resolver el problema de MAG. Presentando la metaheurística basada en recocido simulado y sus diferentes componentes como: método de inicialización, funciones de vecindad, etc.

3

Enfoque de solución propuesto

Introducción

Durante el estudio del estado del arte del problema de MAG, se encontraron dos metaheurísticas que resuelven el problema: GRASP (Duarte *et al.*, 2011) y un algoritmo memético llamado *MAAMP* (Bansal y Srivastava, 2011). Para el desarrollo de esta investigación se propone implementar un algoritmo de Recocido Simulado (RS) para dar solución al problema de MAG sobre diferentes tipos de grafos. El RS es uno de los algoritmos probabilísticos más representativos por sus buenos resultados en diferentes áreas de aplicación (Johnson *et al.*, 1989, Sankar, 1999). Tiene la ventaja de que al no ser de tipo poblacional, sus tiempos de cómputo pueden llegar a ser reducidos.

En este capítulo se hace una breve introducción al RS, mostrando todos los componentes necesarios para su implementación. Para cada componente se describe su objetivo, así como los detalles de la implementación que se emplean en la solución del problema de MAG.

3.1 Recocido simulado

Los algoritmos de búsqueda local trabajan con una solución inicial, la cual se va transformando en otras soluciones (configuraciones) mejores conforme avanza la búsqueda. Lo anterior se logra mediante pequeñas perturbaciones (cambios) en la solución actual. Si los cambios mejoran la solución actual, se sustituye esta por la nueva solución encontrada. Este proceso continua hasta que ninguna mejor solución pueda ser encontrada o se cumpla con un criterio de paro. La solución que se encuentra no necesariamente es el óptimo global. Este tipo de algoritmo básico de búsqueda local, tiene como desventaja que es susceptible a quedar atrapado con facilidad en óptimos locales.

Para evitar este problema, usualmente se permite que algunos movimientos produzcan configuraciones que empeoran la solución actual. Dichos movimientos ayudan a escapar de óptimos locales, pero se deben realizar de manera controlada. En el RS esto se lleva a cabo de manera probabilística.

El RS aplicado a problemas de optimización surgió en los trabajos independientes de Kirkpatrick *et al.* (1983) y Cerny (1985), en donde se aplicó dicha técnica a la solución de problemas como el particionamiento de grafos y el diseño de circuitos VLSI.

El RS tiene gran impacto en el campo de las heurísticas por su simplicidad y eficiencia para resolver problemas de optimización combinatoria. Está inspirado en el proceso de recocido de los metales, en el cual se requiere someter a estos a altas temperaturas para después enfriarlos lentamente. El proceso de enfriamiento tiene gran impacto en la fortaleza del metal. Si la velocidad de enfriamiento es lenta se obtendrá un metal resistente. Por otra parte si la temperatura inicial no es suficientemente alta se llega a un estado llamado metaestable, en el cual el metal tiene imperfecciones y carece de resistencia (Talbi, 2009).

En la Tabla 3.1 se muestra la analogía entre el sistema físico del RS y un problema de optimización, en la cual se puede observar que cada parámetro de la versión computacional del recocido representa un componente del sistema físico que se aplica al recocido de metales.

El RS es fácil de implementar, pero requiere de un cuidadoso proceso de sintonización tanto

Sistema físico	Problema de optimización
Energía	Función objetivo
Temperatura	Parámetro T
Metaestable	Óptimo local
Estado estable	Óptimo global

Tabla 3.1: Analogía entre sistema físico y un problema de optimización

de sus componentes como de sus parámetros con la finalidad de lograr su mejor desempeño.

En el Algoritmo 3.1 se puede observar el pseudocódigo del RS que se propone para dar solución al problema de MAG, el cual es llamado RS-MAG en lo sucesivo.

Algoritmo 3.1 RS-MAG.

Entrada: Función de evaluación f , temperatura inicial $t_0 > 0$, esquema de enfriamiento EF , número de iteraciones $nrep$ y criterio de paro cp

$s_0 \leftarrow \text{BFSi}(\text{nodo de inicio})$
 $s^* \leftarrow s_0$
 $i \leftarrow 0$
 $contIteraciones \leftarrow totIter \leftarrow 0$

mientras cumpla con el criterio de paro cp **hacer**
 $contIteraciones \leftarrow 0$
mientras $contIteraciones = nrep$ **hacer**
 Obtiene una solución $s \in N(s_0)$
 $\Delta = f(s) - f(s_0)$
 genera aleatorio $r \in]0, 1[$
si $(\Delta > 0) \vee (r < e^{(-\Delta/t_i)})$ **entonces**
 $s_0 := s$
si $s > s^*$ **entonces**
 $s^* := s$
fin si
fin si
 $contIteraciones := contIteraciones + 1$
 $totIter := totIter + 1$
fin mientras
 $i := i + 1$
 $t_i := \text{actualizaTemperatura}(t_{i-1}, EF)$
fin mientras

Regresa: s^*

Como parte del RS que se implementó para resolver el problema MAG se desarrollaron los siguientes componentes esenciales:

- Método para obtener una solución inicial

- Función de vecindad
- Función de aceptación
- Método para inicializar la temperatura
- Longitud de la cadena de Markov
- Esquema de enfriamiento
- Condición de paro

Es importante hacer notar que para algunos de estos componentes se desarrollaron diversas propuestas. A continuación se explica el objetivo de estos componentes. Posteriormente se detallan sus diferentes propuestas.

3.2 Método para generar la solución inicial

Existen dos estrategias principales para la creación de soluciones iniciales en los algoritmos de búsqueda local: el enfoque aleatorio y el voraz (*greedy*). Entre estos dos enfoques existen diferencias en cuanto a calidad de solución y tiempo de cómputo. Regularmente, se cree que el enfoque de inicialización voraz permite a los algoritmos de búsqueda local obtener mejores resultados al final del proceso de búsqueda. Sin embargo, esto depende de la eficiencia y efectividad del algoritmo voraz que se implemente así como de las propiedades del método de búsqueda local seleccionado Talbi (2009).

Un algoritmo voraz ofrece soluciones de mejor calidad, por lo tanto, se van a requerir de menos iteraciones para converger a una mejor solución o para poder alcanzar el óptimo global.

La solución inicial que se utilizará para comenzar la búsqueda con RS-MAG, se puede construir utilizando alguno de los dos métodos siguientes: aleatorio y basado en búsqueda primero en anchura el cual es llamado BFSi.

La solución aleatoria consiste en etiquetar aleatoriamente y de manera diferente todos los nodos de un grafo sin tomar en cuenta la estructura del mismo. La solución generada por BFSi

construye una estructura basada en niveles en la cual los nodos están organizados.

Para generar la solución basada en BFSi, se realiza una búsqueda primero en anchura sobre el grafo, lo cual genera una estructura por niveles en donde los nodos pertenecientes a los niveles alternos no son adyacentes (Figura 3.1). Para dar inicio al proceso de etiquetado se elige aleatoriamente entre el nivel 1 o el nivel 2 de la estructura generada. Si se inicia con el nivel 1 todos los nodos pertenecientes a los niveles impares se etiquetan primero y se continúa con los nodos en los niveles pares, ver ejemplo de la Figura 3.2. La asignación de etiquetas se hace de forma consecutiva y ascendente para los nodos de un mismo nivel de acuerdo al orden en que estos fueron visitados en el proceso de la búsqueda primero en anchura.

Esta forma de construir la solución inicial es una variante del algoritmo *LAH* propuesto por Bansal y Srivastava (2011). A diferencia del enfoque que se presenta, *LAH* asigna las etiquetas de forma voraz, es decir, calculando que el etiquetado parcial actual sea el de mejor AB.

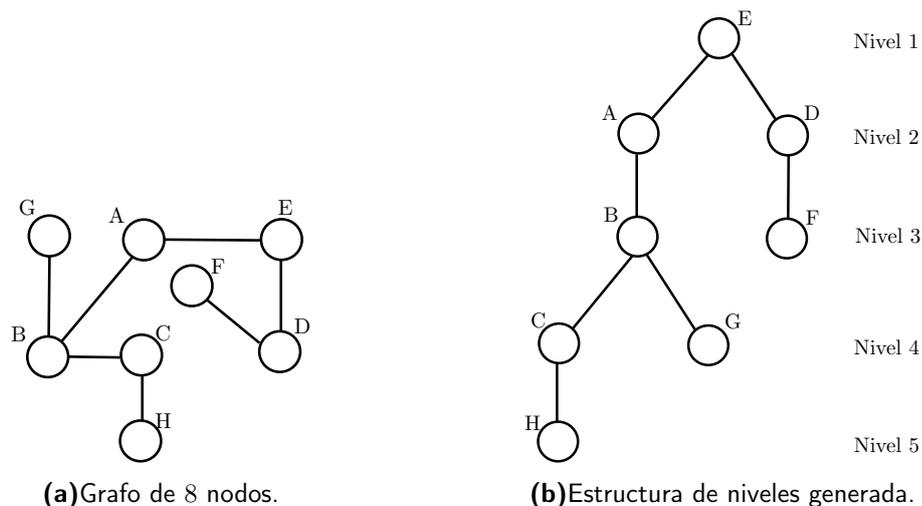


Figura 3.1: Estructura de niveles de un grafo con 8 nodos y 7 arcos generada con búsqueda primero en anchura.

3.3 Función de vecindad

La eficiencia del RS es altamente influenciada por la función de vecindad utilizada (Moscato, 1993). El propósito de la función de vecindad es generar una solución s' mediante la aplicación de un operador de movimientos que perturba de forma sutil una solución s . Uno de los

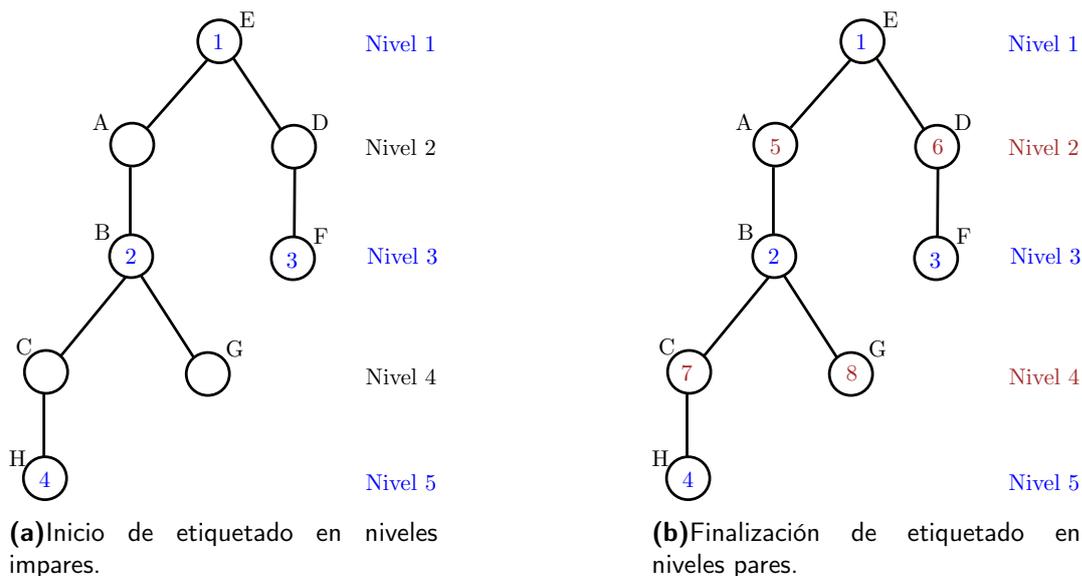


Figura 3.2: Proceso de etiquetado del grafo de la Figura 3.1(a).

operadores más utilizados es el llamado *swap* o intercambio de elementos, que puede ser totalmente aleatorio o se pueden realizar aplicando algún tipo de movimiento inteligente.

Para este trabajo de tesis se analizaron cuatro funciones de vecindad básicas y seis combinaciones derivadas de las mismas. Antes de explicar cada una de ellas primero se presentan algunos conceptos que son utilizados en sus definiciones. Sea $A_1(u) \subseteq V$ el conjunto de nodos adyacentes a u (vecinos de jerarquía 1), y $A_2(u) \subseteq V$ el conjunto de nodos (vecinos de jerarquía 2) que tienen en común un arco con un nodo perteneciente a $A_1(u)$. Sea $swap(\varphi, u, v)$ una función que permite el intercambio de las etiquetas entre dos nodos u y v para un etiquetado φ dado.

Las cuatro funciones de vecindad básicas propuestas para su uso en RS-MAG son:

- F_1 (*aleatoria*): Se eligen de manera aleatoria dos nodos diferentes del grafo y se intercambian las etiquetas de dichos nodos. La definición formal de esta función de vecindad está descrita en la Ecuación 3.1

$$F_1(\varphi) = \{\varphi' = swap(\varphi, u, v), (u, v) \in V, u \neq v\} \quad (3.1)$$

- F_2 (*jerarquía 1*): Se elige aleatoriamente un nodo del grafo y se obtiene una lista con los

vecinos de jerarquía 1 del nodo seleccionado, enseguida se elige un nodo de la lista de forma aleatoria y se intercambian las etiquetas de los dos nodos que fueron seleccionados. La definición matemática de esta función se muestra en la Ecuación 3.2

$$F_2(\varphi) = \{\varphi' = \text{swap}(\varphi, u, v), u \in V, v \in A_1(u)\} \quad (3.2)$$

- F_3 (jerarquía 2): Se elige de manera aleatoria un nodo del grafo y se obtiene una lista con los vecinos de jerarquía 2, después se elige aleatoriamente un nodo de la lista y se intercambian las etiquetas entre los dos nodos que han sido seleccionados. Matemáticamente esta función se describe como se muestra en la Ecuación 3.3

$$F_3(\varphi) = \{\varphi' = \text{swap}(\varphi, u, v), u \in V, v \in A_2(u)\} \quad (3.3)$$

- F_4 (jerarquía 1 y 2): Se elige aleatoriamente un nodo del grafo y se obtiene una lista de vecinos tanto de jerarquía 1 y 2, después se elige aleatoriamente un nodo de la lista y se intercambian las etiquetas de los dos nodos que se han seleccionado. La definición formal está descrita en la Fórmula 3.4

$$F_4(\varphi) = \{\varphi' = \text{swap}(\varphi, u, v), u \in V, v \in A_1(u) \cup A_2(u)\} \quad (3.4)$$

Una vez que se tienen definidas las diferentes funciones de vecindad básicas se describen de manera formal las seis combinaciones formadas a partir de pares ordenados de funciones de vecindad diferentes $F_i(\varphi)$ y $F_j(\varphi)$. La combinación de los vecindarios $F_{i,j}(\varphi, x)$ se aplica con una probabilidad p . Siguiendo la relación descrita en la Ecuación 3.5.

$$F_{i,j}(\varphi, x) = \begin{cases} F_i(\varphi) & \text{si } x \leq p \\ F_j(\varphi) & \text{si } x > p \end{cases} \quad (3.5)$$

donde x es un número aleatorio del intervalo $[0, 1]$ e i, j son los índices de las dos funciones de vecindad a combinar.

En la Sección 4.4.2 se muestran los resultados obtenidos de las experimentaciones realizadas para estudiar la efectividad de cada una de las diez funciones de vecindad anteriores.

3.4 Función de aceptación

Gracias a la función de aceptación el RS puede escapar de óptimos locales. La probabilidad de aceptación de un vecino que no mejora es proporcional a la temperatura t_i actual y se calcula mediante la Ecuación 3.6.

$$P(\Delta, t_i) = \exp\left(\frac{-\Delta}{kt_i}\right) > r \quad (3.6)$$

donde Δ es el cambio en la función de evaluación, t_i es la temperatura actual, k es una constante conocida como constante de Boltzmann y r es un aleatorio entre 0 y 1.

Esta función es conocida como criterio de aceptación de Metrópolis (Metrópolis *et al.*, 1953).

3.5 Temperatura inicial

El RS es altamente sensible a la temperatura. Si la temperatura inicial es muy alta, los movimientos que son aceptados serán demasiados. Por otra parte si la temperatura es baja puede caer en óptimos locales con mayor facilidad. Por lo tanto, se debe tener un balance al momento de inicializar la temperatura.

Para inicializar la temperatura t_0 de RS-MAG, se utilizó el procedimiento descrito en Laarhoven y Aarts (1987). Este procedimiento emplea la Ecuación 3.7, la cual usa datos estadísticos recolectados durante una experimentación preliminar con 10^4 soluciones aleatoriamente generadas:

$$t_0 = \frac{\overline{\Delta}^{(-)}}{\ln(\chi_0)^{-1}} \quad (3.7)$$

donde $\overline{\Delta}^{(-)}$ representa el promedio del decremento del valor de AB de las soluciones generadas, y χ_0 define la tasa de movimientos aceptados, es decir, el número de soluciones aceptadas (soluciones que incrementan el AB) dividido entre el número total de soluciones en el experimento.

3.6 Longitud de la cadena de Markov

La cadena de Markov en el RS está formada por el conjunto de las transiciones de soluciones visitadas en cada temperatura para alcanzar un estado de equilibrio. La longitud de la cadena de Markov es el número de transiciones entre soluciones visitadas en cada temperatura (Talbi, 2009).

Para definir la longitud adecuada de la cadena de Markov, Talbi (2009) sugiere las siguientes formas:

- Cadena estática: Con esta estrategia, el número de transiciones es determinado antes de que inicie el proceso de búsqueda
- Cadena adaptativa: El número de transiciones depende de las características del espacio de búsqueda del problema en cuestión

Para nuestro RS-MAG se implementaron dos maneras de calcular la longitud de la cadena de Markov. La primera es asignar de manera fija para todas las instancias un número de transiciones predeterminado con base en la experimentación previa. La segunda propuesta es tomando en cuenta la densidad de arcos para cada instancia de prueba. La densidad de arcos es el número total de conexiones entre nodos que existen en el grafo. La densidad se obtiene mediante la Ecuación 3.8.

$$Densidad = \frac{|E| \times 100 \times 2}{|V| \times |V| - 1} \quad (3.8)$$

donde $|E|$ es el número de arcos y $|V|$ el número de nodos.

3.7 Esquema de enfriamiento

El RS requiere que la temperatura se decremente conforme avanza la búsqueda. Romeo y Sangiovanni-Vincentelli (1991) señalan que un esquema de enfriamiento eficaz es primordial para reducir la cantidad de tiempo requerido por el algoritmo para encontrar una solución óptima. Por consiguiente, existen varios estudios enfocados en esquemas de enfriamiento (Fox y Heine, 1993, Cardoso *et al.*, 1994, Nourani y Andresen, 1998)

Un buen esquema de enfriamiento es aquel que provee un buen compromiso entre la calidad de la solución final y el tiempo de cómputo consumido. Para ello la actualización de la temperatura puede realizarse de diferentes formas. Estos esquemas de enfriamiento pueden dividirse en dos tipos: estáticos y adaptativos (Melício *et al.*, 1998, Henderson *et al.*, 2003).

El esquema de enfriamiento estático es aquel que es definido complementamente *a priori*. En el esquema de enfriamiento adaptativo la tasa en que se decrementa la temperatura es dinámica y depende de la información obtenida durante la búsqueda, es decir, el cambio de temperatura se basa en información estadística como por ejemplo el número de movimientos que mejoran la solución actual.

Para llevar a cabo la experimentación con el algoritmo RS-MAG, se implementaron diferentes funciones de enfriamiento que enseguida se presentan.

3.7.1 Enfriamiento lineal

La velocidad de enfriamiento que proporciona este esquema es rápida comparada con otros métodos. Debido a la velocidad con la que disminuye la temperatura, realiza una búsqueda muy rápida y la solución obtenida al final puede ser probablemente un óptimo local aún cuando el algoritmo pueda escapar de algunos de estos. En este esquema de enfriamiento, la temperatura actual t_i es calculada siguiendo la Ecuación 3.9 (Talbi, 2009):

$$t_i = t_0 - i \times \beta \quad (3.9)$$

donde t_i representa la temperatura en la iteración i .

3.7.2 Enfriamiento geométrico

El esquema de enfriamiento geométrico es fácil de implementar y es utilizado en diversas investigaciones (Elmohamed *et al.*, 1998, Herault, 1998, Fielding, 2000). El motivo por el cual este esquema es tan popular es debido al hecho de que la velocidad de decremento de la temperatura es más mesurada que en otros esquemas. En el esquema de enfriamiento geométrico, para calcular la temperatura actual se utiliza la Ecuación 3.10

$$t_i = \alpha t_{i-1} \quad (3.10)$$

donde $\alpha \in [0, 1]$.

Diversos trabajos que aplican este esquema, muestran que valores altos de α entre 0.80 y 0.99 proporcionan buenos resultados (Fielding, 2000, Rodriguez-Tello *et al.*, 2008, Torres-Jimenez y Rodriguez-Tello, 2010).

3.7.3 Enfriamiento logarítmico

Este esquema de enfriamiento es muy lento para implementarse en la práctica (Talbi, 2009). Albrecht y Wong (2000) realizaron un estudio detallado que se enfoca en este tipo de esquemas. Con un enfriamiento logarítmico se pueden encontrar buenas soluciones pero la desventaja es la gran cantidad de tiempo de cómputo que puede llegar a consumir. Sin embargo, puede combinarse con la utilización de diferentes criterios de paro.

$$t_i = \frac{t_0}{\log(i)} \quad (3.11)$$

donde t_0 es la temperatura inicial.

3.8 Condición de paro

La teoría sugiere que la condición de paro ideal es aquella donde la temperatura final sea igual a cero. Sin embargo es algo impráctico. Otten y van Ginneken (1988) realizaron el análisis de un criterio de paro que toma en cuenta estadísticas recolectadas durante la ejecución. Además del anterior existen diferentes criterios de paro, como los que describe Talbi (2009).

- **Definir una temperatura final**

En la práctica es muy poco factible fijar una temperatura final igual a cero. Se podría consumir más tiempo de cómputo del necesario, pues en algunos casos se puede llegar a la solución óptima o sub-óptima antes de que la temperatura llegue a cero. Por lo cual, es recomendable fijar temperaturas cercanas a cero en base a experimentación previa.

- **Número de iteraciones**

Existen dos maneras de aplicar este criterio de paro. El primero es predefinir un número fijo de iteraciones que realiza el algoritmo .

El segundo enfoque es tomar en cuenta el comportamiento de cada ejecución. Es decir, terminar el algoritmo cuando después de cierto número de iteraciones no se supera la mejor solución actual.

- **Movimientos aceptados**

Este criterio de paro toma en cuenta los movimientos que son aceptados por cada cambio de temperatura. Cuando se tiene una tasa de aceptación demasiado baja la ejecución termina.

Para RS-MAG se estableció el criterio de paro basado en el número de iteraciones fijas.

Resumen del capítulo

En este capítulo se presentó una breve introducción del algoritmo de RS, el cual se propone para dar solución al problema de MAG. La versión del recocido simulado presentado es llamado RS-MAG y sus principales componentes analizados son los siguientes:

- Dos métodos para obtener una solución inicial
- Cuatro diferentes funciones de vecindad
- Un criterio de aceptación de movimientos
- Un método para inicializar la temperatura
- Dos criterios para definir la longitud de la cadena de Markov
- Cuatro distintos esquemas de enfriamiento
- Una condición de paro

En el capítulo siguiente se presentan los resultados obtenidos en las experimentaciones realizadas para evaluar RS-MAG. Además se describen los conjuntos de instancias utilizados y se presentan las condiciones experimentales bajo las cuales se ejecutó RS-MAG.

También se detalla la fase de sintonización una de las etapas importantes. En la cual se utilizó una método basado en arreglos de cobertura (covering arrays) que está inspirada en pruebas de interacción combinatoria (Cohen *et al.*, 1996).

4

Experimentación y resultados

Introducción

En este capítulo se presentan los experimentos realizados para la evaluación del recocido simulado el cual es nombrado RS-MAG. También se describen las instancias de prueba que fueron tomadas de la literatura y se detallan las condiciones experimentales utilizadas para la evaluación de RS-MAG.

El proceso experimental se dividió en una etapa preliminar y otra final. Para la experimentación preliminar se compararon las diferentes formas de obtener una solución inicial. En esta etapa se experimentó con el método que se basa en BFS, el cual se ha nombrado BFSi. Así mismo, se comparó el desempeño de las diferentes funciones de vecindad presentadas en el capítulo anterior para elegir aquellas que favorecen el desempeño global de RS-MAG. La sintonización es una fase muy importante de cualquier algoritmo, por ello en este capítulo se describe la técnica de sintonización empleada para RS-MAG que hace uso de arreglos de cobertura (*covering arrays*) y que está inspirada en las pruebas de interacción combinatoria (Cohen *et al.*, 1996). En la etapa de experimentación final se realizó una comparación entre

RS-MAG y los algoritmos del estado del arte en términos de calidad de solución y tiempo de cómputo consumido.

4.1 Criterios de comparación

El primer criterio de comparación empleado es el mismo que se utiliza ampliamente en la literatura: el mejor valor del antibandwidth encontrado. En los resultados de la experimentación entre más alto sea el AB encontrado este es mejor.

El segundo criterio es el tiempo de cómputo consumido por los algoritmos para resolver cada instancia del problema.

Con la finalidad de hacer una comparación justa entre los algoritmos, estos fueron probados sobre las mismas instancias de prueba comúnmente utilizadas en la literatura, y que son descritas a continuación.

4.2 Instancias de prueba

Las instancias de prueba empleadas en este trabajo de investigación se encuentran divididas en tres conjuntos, los cuales fueron propuestos por diferentes investigadores que han trabajado anteriormente sobre el problema de MAG. En las siguientes subsecciones se detallan dichos conjuntos.

4.2.1 Grafos con óptimo conocido

Este conjunto de instancias es utilizado para la experimentación preliminar. Se conforma de 12 grafos: 3 ciclos, 3 caminos, 3 árboles binarios y 3 mallas. Su tamaño varía entre 81 y 1170 nodos. El valor óptimo del antibandwidth de estos grafos se puede obtener empleando fórmulas reportadas en la literatura.

4.2.2 Instancias de grafos tipo malla

Este tipo de instancias son construidas como el producto cartesiano de dos caminos (Raspaud *et al.*, 2009) y fueron propuestas por Duarte *et al.* (2011). Un grafo tipo camino (*path*) es aquel que puede ser dibujado de tal forma que todos sus nodos y arcos sigan una sola línea.

Este conjunto de instancias se divide en dos subconjuntos. El primero conformado por 12 instancias con un número de nodos que varía entre 81 y 120. El segundo subconjunto está compuesto por 12 instancias cuyo número de nodos oscila entre 960 y 1170.

4.2.3 Instancias *Rome*

Las instancias de tipo *Rome* son grafos generados aleatoriamente y cada uno de ellos es no dirigido, biconectado y planar. De este conjunto fueron seleccionados 30 grafos diferentes, específicamente del tipo RND BUP. Este tipo de instancias fueron originalmente propuestas por Bansal y Srivastava (2011). Estas instancias están disponibles en la siguiente dirección electrónica: <http://www.graphdrawing.org/data>.

4.2.4 Instancias *Harwell Boeing*

Este tipo de instancias se derivan de la colección *Harwell Boeing Sparse Matrix* y son problemas de una amplia variedad de disciplinas científicas y de ingeniería. Fueron propuestas por Duarte *et al.* (2011) y son un total de 30. Dicha colección se encuentra en la siguiente dirección electrónica: <http://math.nist.gov/MatrixMarket/data/Harwell-Boeing>.

Este conjunto se conforma de dos subconjuntos. El primer grupo consta de 12 instancias de entre 30 y 100 nodos. El segundo subconjunto está formado de 12 instancias de entre 400 y 900 nodos.

4.3 Condiciones experimentales

RS-MAG fue codificado en lenguaje C y compilado con `gcc` usando la bandera de optimización `-O3`. Se ejecutó de manera secuencial en un cluster con 4 CPUs Xeon X5650 a 2.66 GHz (24 núcleos), 32 GB de RAM con sistema operativo Linux.

Los parámetros utilizados para la ejecución de RS-MAG se describen en secciones posteriores.

4.4 Experimentación preliminar

En la etapa de experimentación preliminar, se realizaron pruebas para observar el desempeño de los diferentes métodos para obtener una solución inicial y las diferentes funciones de vecindad que se propusieron en las Secciones 3.2 y 3.3 del capítulo anterior. El objetivo de estas pruebas es el de identificar aquellos componentes que permitan a nuestro RS-MAG encontrar soluciones de calidad y de manera rápida.

4.4.1 Experimentación preliminar para obtener una solución inicial

Las instancias utilizadas en esta experimentación son las que pertenecen al conjunto con óptimo conocido, presentadas en la Sección 4.2.1.

En la Tabla 4.1 se concentran los resultados obtenidos a partir de 31 ejecuciones del Algoritmo BFSi y del método aleatorio presentado en la Sección 3.2. En la primera columna se indica el nombre de la instancia, en la siguiente columna se denota el número de nodos ($|V|$), enseguida se muestra el número de arcos $|E|$, en la cuarta columna se presenta el óptimo conocido (C^*). Las siguientes tres columnas muestran resultados del método aleatorio, se indica el mejor valor del AB alcanzado (C), el AB promedio ($Prom$) y el porcentaje de desviación ($\% \Delta_C$) al límite inferior teórico. Las últimas tres columnas presentan el mismo tipo de resultados para el método BFSi. El porcentaje de desviación al mejor reportado ($\% \Delta_C$) de

cada algoritmo se calculó mediante la Ecuación 4.1.

$$\Delta_C = 1 - \frac{C}{C^*} \quad (4.1)$$

Al final de la tabla podemos observar los promedios de las diferentes columnas. Claramente se puede observar que el método BFSi ofrece resultados mejores que los obtenidos por el método aleatorio. Los resultados del procedimiento aleatorio muestran en promedio un 97.56 % de desviación con respecto a las soluciones óptimas conocidas.

Grafo	V	E	C*	Aleatorio			BFSi		
				C	Prom	% Δ_C	C	Prom	% Δ_C
cycle-90	90	90	44	1	1.00	97.73	43	43.00	2.27
cycle-102	102	102	50	2	1.16	96.00	49	49.00	2.00
cycle-1170	1170	1170	584	1	1.00	99.83	583	583.00	0.17
path-81	81	80	40	2	1.10	95.00	39	38.48	2.50
path-115	115	114	57	2	1.45	96.49	56	55.53	1.75
path-1170	1170	1169	585	2	1.29	99.66	583	583.00	0.34
btree-81	81	80	40	1	1.00	97.50	28	18.38	30.00
btree-115	115	114	57	3	1.13	94.74	33	33.00	42.11
btree-1171	1171	1170	585	1	1.00	99.83	399	289.00	31.79
mesh9x9	81	144	36	1	1.00	97.22	36	36.00	0.00
mesh11x11	121	220	55	1	1.00	98.18	55	55.00	0.00
mesh33x33	1089	2112	528	1	1.00	99.81	528	528.00	0.00
nos4	1089	2112	47	1	1.00	97.87	1	1.00	97.87
sherman4	1089	2112	272	1	1.00	99.63	196	194.76	27.94
dwt_592	1089	2112	295	1	1.00	99.66	1	1.00	99.66
ug3-60	1089	2112	29	1	1.00	96.55	1	1.00	96.55
ug3-80	1089	2112	39	2	1.12	94.87	1	1.00	97.44
ug4-90	1089	2112	44	2	1.09	95.45	1	1.00	97.73
Prom				1.44	1.07	97.56	146.28	139.51	35.01

Tabla 4.1: Comparación entre métodos de inicialización aleatorio y BFSi sobre un conjunto de grafos con óptimo conocido.

En la Figura 4.1 se puede observar con mayor claridad que el método BFSi nos ofrece mejores resultados. La gráfica muestra el porcentaje de desviación al mejor resultado conocido (eje de las ordenadas) para cada instancia utilizada (eje de las abscisas). Con la desviación al mejor resultado conocido se deduce que entre más cercano a cero este valor se encuentra, es mejor la solución. Por lo tanto si alguna línea está en cero, significa que se ha llegado al óptimo conocido. La línea roja representa el método aleatorio y la azul representa el método BFSi. Se puede observar que el método BFSi sobre instancias como los ciclos, caminos y mallas (cycles, path y mesh), tiende a encontrar soluciones con valor del antibandwidth muy cercano al óptimo y el método aleatorio ofrece soluciones de mala calidad. Lo anterior se puede

visualizar de manera sencilla puesto que en dichos casos. La región entre la línea roja y azul es grande. Para el caso de las instancias tipo árbol (btree) la región es más pequeña, por que las soluciones iniciales producidas por BFSi son en promedio más alejadas de los valores óptimos. Para las instancias del conjunto Rome (*ug3-60*, *ug3-80* y *ug4-90*) y Harwell-Boeing (*nos4*, *sherman4* y *dtw_592*) no se logra obtener buenos resultados, solo para la instancia *sherman4* se obtiene una solución más cercana a la óptima. Los dos métodos obtienen soluciones muy alejadas de los valores óptimos por ello en la gráfica se observa que para dichas instancias las líneas roja y azul están muy cerca.

El método BFSi funciona de manera muy eficaz y rápida sobre las instancias tipo malla alcanzado los valores óptimos con el etiquetado generado. Lo anterior debido a que el método aprovecha la estructura del grafo para generar la solución.

Los tiempos de cómputo de ambos métodos de inicialización son realmente bajos (menores a 0.01 segundos) por lo cual se decidió omitirlos de la Tabla 4.1.

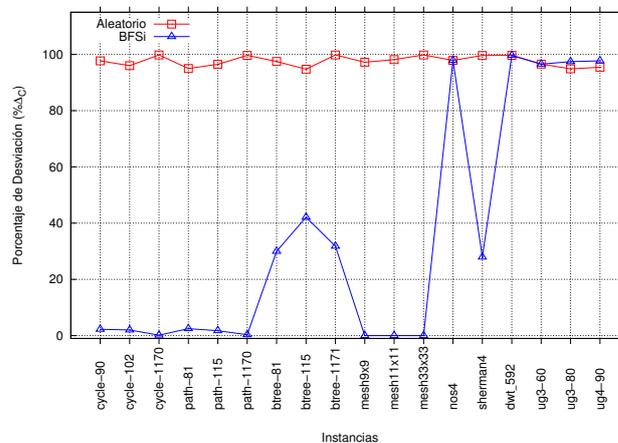


Figura 4.1: Comparación del desempeño promedio entre métodos de inicialización aleatorio y BFSi sobre un conjunto de 18 grafos con óptimo conocido.

Analizando la gráfica anterior se pueden detectar los conjuntos de instancias que son difíciles de encontrar soluciones óptimas, en este caso las pertenecientes a los conjuntos Rome y Harwell-Boeing. Por lo cual existe la necesidad de desarrollar la metaheurística para encontrar soluciones de una calidad superior con respecto a las reportadas en el estado del arte.

4.4.2 Experimentación preliminar sobre diferentes vecindarios

Como parte de la experimentación preliminar se realizó esta prueba para observar el desempeño de las diferentes funciones de vecindad detalladas en el capítulo anterior y algunas combinaciones entre ellas.

Se utilizaron 5 instancias representativas del conjunto Harwell-Boeing y se ejecutó RS-MAG 31 veces. En la Tabla 4.2 se presentan las diferentes combinaciones de las funciones de vecindad y las probabilidades que se aplicaron. Las funciones a combinar son representadas por un par ordenado como se muestra en la Ecuación 3.5. Al primer elemento se le aplica la probabilidad p y al segundo el elemento una probabilidad $p - 1$.

La combinación ganadora es $F_{1,4}$, pero se puede observar que utilizando diferentes probabilidades se obtienen resultados buenos con respecto al resto. Por lo tanto, se consideran las probabilidades 60 con la que se obtuvo 19.63 del AB promedio y la probabilidad 20 que obtuvo 19.53 del AB promedio.

Combinación de Vecindario	Probabilidad p					
	100	80	60	40	20	0
$F_{1,2}$	19.19	7.00	6.30	6.35	6.22	5.83
$F_{1,3}$	19.19	19.46	19.45	19.11	19.37	16.77
$F_{1,4}$	19.19	19.34	19.63	19.47	19.53	18.34
$F_{2,3}$	5.83	6.08	7.12	7.47	7.99	16.77
$F_{2,4}$	5.83	5.80	6.58	6.98	8.37	18.34
$F_{3,4}$	16.77	19.15	19.25	19.20	19.37	18.34

Tabla 4.2: Comparación del desempeño entre diferentes combinaciones de vecindarios para RS-MAG.

En la Figura 4.2 se observa más fácilmente cual vecindario tiene mejor desempeño. En el eje de las ordenadas se muestra el valor promedio del antibandwidth y en el eje de las abscisas las distintas probabilidades utilizadas. Entre mayor sea el AB promedio obtenido es mejor. La línea azul es la mejor en general y representa a la combinación $F_{1,4}$. En la Figura 4.3 se observa de manera más detallada los resultados mediante un acercamiento de la parte superior de la gráfica.

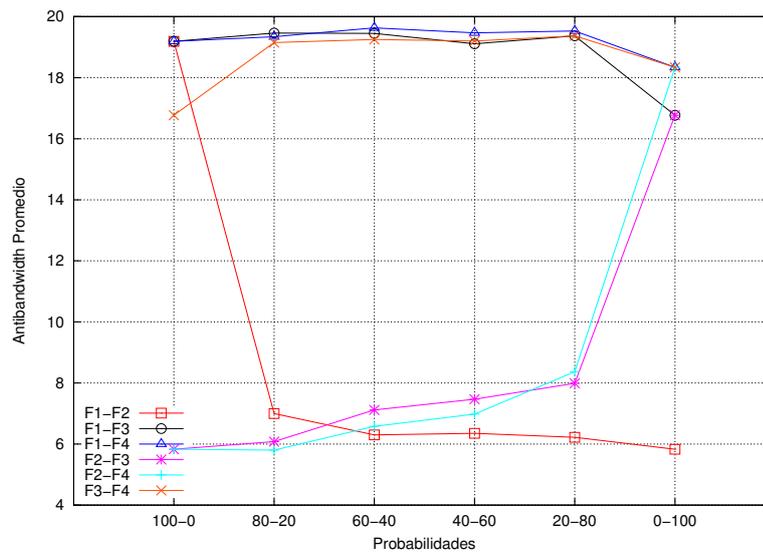


Figura 4.2: Desempeño promedio de diferentes funciones de vecindad combinadas obtenido sobre 5 instancias del conjunto Harwell-Boeing.

4.5 Sintonización de parámetros

La optimización de parámetros es una tarea importante en el diseño de algoritmos. Diferentes métodos han sido propuestos en la literatura para encontrar la combinación más adecuada de valores para los parámetros de entrada (Adenso-Diaz y Laguna, 2006, de Landgraaf *et al.*, 2007, Gunawan *et al.*, 2011). En este trabajo se emplea una metodología de sintonización, previamente reportada en (Gonzalez-Hernandez y Torres-Jimenez, 2010), la cual está basada en pruebas de interacción combinatoria (Cohen *et al.*, 1996). Se decidió utilizar este enfoque por que permite reducir significativamente el número de pruebas (experimentos) necesarias para determinar la mejor combinación de valores para los parámetros de un algoritmo. En lugar de probar de manera exhaustiva todas las combinaciones de valores para los parámetros del algoritmo, solo se analizan las interacciones de t (o menos) parámetros de entrada mediante la creación de pruebas de interacción que incluyen al menos una vez todas las t combinaciones entre los parámetros y sus valores.

Para implementar este enfoque se utilizó un Arreglo de Cobertura con Niveles Mezclados (*Mixed Level Covering Array*) que se define como $MCA(N : t, k, (v_1, v_2, \dots, v_k))$, i.e. una

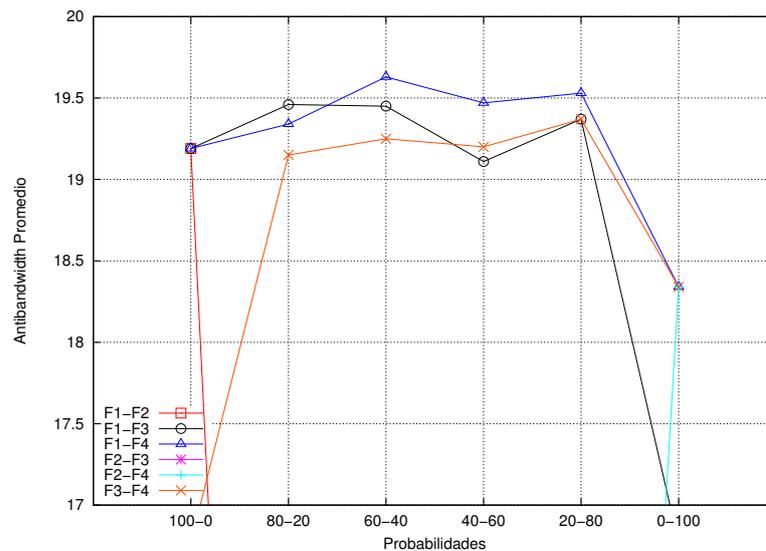


Figura 4.3: Acercamiento superior de la Figura 4.2.

matriz de $N \times k$ con v símbolos, donde $v = \sum_{i=1}^k v_i$, que tiene las siguientes propiedades:

- Cada columna i ($1 \leq i \leq k$) contiene los elementos del conjunto S_i con $|S_i| = v_i$.
- Todas las filas de cada submatriz de tamaño $N \times t$ contienen todas las t -tuplas de valores al menos una vez.

El conjunto v del MCA es expresado en notación exponencial. Donde la base indica el número de valores diferentes que pueden asignarse a un parámetro en particular. Si dos parámetros diferentes pueden adquirir un valor de tres posibles, entonces 3^2 se expresaría en el MCA.

Para RS-MAG se identificaron $k = 6$ parámetros de entrada: método de inicialización de la solución, funciones de vecindad, longitud de la cadena de Markov (L), tasa de aceptación, criterio de paro, esquema de enfriamiento. Con base en la experimentación preliminar se seleccionaron v posibles valores para cada parámetro de entrada (ver Tabla 4.3).

En la Tabla 4.4 se muestra la matriz transpuesta del MCA resultante, donde cada columna representa un caso de prueba. Se ha construido el arreglo de cobertura más pequeño posible $MCA(63; 3, 6, (2^3, 3^2, 7^1))$ usando el AM reportado por Rodríguez-Tello y Torres-Jimenez (2010). En este CA es muy fácil asignar valores que representen los parámetros de entrada.

No.	Caso	Tiempo	Prom AB	Desv
22	000226	0.54	30.01	0.36
37	000224	9.56	30.01	0.32
6	000123	0.58	29.95	0.23
31	100225	1.42	29.92	0.17
8	000013	2.78	29.88	0.21
36	111126	0.60	29.86	0.15
32	001121	0.74	29.83	0.07
63	011125	2.25	29.81	0.04
24	101025	0.46	29.80	0.00
40	001122	0.47	29.80	0.00

Tabla 4.6: 10 mejores casos ejecutados por RS-MAG sobre grafos del conjunto Rome.

4.6 Experimentación final

Para la experimentación final se ejecutó 31 veces RS-MAG sobre cada grafo de los diferentes conjuntos de instancias presentados en las Secciones 4.2.2 a 4.2.4.

Las Tablas 4.7, 4.8 y 4.9 muestran los resultados obtenidos en la experimentación final. En sus primeras tres columnas se indica el nombre del grafo, el número de nodos y por último el número de arcos. Enseguida se denota el límite inferior teórico (C^*) conocido por construcción el cual equivale al óptimo. Las siguientes columnas presentan los resultados alcanzados por los algoritmos reportados en la literatura. Primero se indica el mejor AB alcanzado (C), el AB promedio ($Prom.$), tiempo promedio (T) y la desviación al mejor valor del AB conocido ($\% \Delta_C$). Las columnas restantes presentan la información de RS-MAG, y se utilizan las mismas columnas antes mencionadas agregando la desviación estándar ($Desv.$). Al final de cada tabla se muestran algunos promedios que son de ayuda para el análisis de la información.

Las Figuras 4.4, 4.6, 4.7 y 4.8 muestran la desviación al mejor conocido en términos de porcentaje (eje de las ordenadas) de las instancias probadas (eje de las abscisas) para cada conjunto de instancias utilizadas.

También se utilizó un tipo de gráfica llamada *tiempo a la mejor solución* (*time to target*) (Figuras 4.5 y 4.9). Este tipo de representación de información es utilizada en otras investigaciones (Aiex *et al.*, 2002). Para mostrar la probabilidad de que un algoritmo encuentre una buena solución (eje de las ordenadas) en un rango de tiempo de ejecución (eje de las abscisas). En dichas gráficas la línea verde representa el comportamiento teórico. Las cruces

rojas representan los tiempos en que cada ejecución tardó en encontrar la mejor solución.

4.6.1 Resultados sobre instancias del conjunto Rome

El algoritmo MAAMP propuesto por Bansal y Srivastava (2011) es contra el cual se realizan las comparaciones para este conjunto de prueba.

En la Tabla 4.7 podemos observar que en las 30 instancias que conforman el conjunto de prueba Rome, se superan los resultados publicados. El mayor porcentaje de desviación al límite teórico mediante RS-MAG es de 28.57% mientras que con el algoritmo MAAMP es de 36.73%. Entre menor sea la desviación son mejores las soluciones encontradas. En el caso del grafo *ug1-80* utilizando RS-MAG la desviación al mejor AB es de 10.26%, es el valor más bajo de todos los casos.

En términos generales, el algoritmo MAAMP presenta una desviación promedio en todo el conjunto de instancias de 25.04% mientras que RS-MAG muestra 20.85% como lo podemos observar en la última fila de la Tabla 4.7. RS-MAG no solo presenta soluciones con mejor calidad, además el tiempo de cómputo consumido es menor. MAAMP utiliza en promedio 257.79 segundos, en contraste con RS-MAG que consume en promedio solo 3.39 segundos.

En la Figura 4.4 se observa la gráfica de las desviaciones al mejor AB descrita en el inicio de este capítulo. El algoritmo MAAMP obtiene solo 4 instancias con una desviación menor al 20.00%, es decir, el 13.33% de las instancias. En contraste, para RS-MAG en 10 casos es menor, lo que es equivalente al 33.33% de los grafos de este conjunto.

La Figura 4.5 muestra como en 22 ejecuciones de las 31 (73.33% de las corridas) RS-MAG encuentra la mejor solución en menos de 2.5 segundos para la instancia *ug1-80*. Cabe mencionar que se observaron resultados similares para el resto de los grafos de este conjunto.

Grafo	V	E	C*	MAAMP				RS-MAG				
				C	Avg.	T	% Δ_C	C	Avg.	Dev.	T	% Δ_C
ug1-50		64		18	17.40	65.28	25.00	19	19.00	0.00	0.09	20.83
ug2-50		66		18	16.80	33.51	25.00	20	19.10	0.30	0.10	16.67
ug3-50	50	63	24	17	16.80	52.04	29.17	19	18.23	0.42	0.05	20.83
ug4-50		70		19	18.40	59.55	20.83	20	19.77	0.42	8.73	16.67
ug5-50		62		20	19.80	83.44	16.67	21	21.00	0.00	1.44	12.50
ug1-60		79		21	20.20	155.98	27.59	23	22.06	0.25	0.12	20.69
ug2-60		81		23	21.80	21.80	20.69	24	23.97	0.18	77.23	17.24
ug3-60	60	84	29	22	20.20	116.12	24.14	23	23.00	0.00	1.48	20.69
ug4-60		79		22	21.20	235.20	24.14	23	23.00	0.00	0.23	20.69
ug5-60		80		21	20.80	70.20	27.59	22	22.00	0.00	0.16	24.14
ug1-70		98		26	23.80	354.52	23.53	27	27.00	0.00	6.06	20.59
ug2-70		82		28	15.80	54.20	17.65	29	29.00	0.00	0.25	14.71
ug3-70	70	82	34	27	26.20	116.31	20.59	28	28.00	0.00	0.21	17.65
ug4-70		97		23	22.60	74.50	32.35	25	24.16	0.37	0.11	26.47
ug5-70		88		26	25.50	323.45	23.53	27	27.00	0.00	0.32	20.59
ug1-80		92		34	33.60	61.07	12.82	35	35.00	0.00	1.93	10.26
ug2-80		93		31	30.20	226.90	20.51	32	32.00	0.00	0.33	17.95
ug3-80	80	95	39	28	27.20	138.68	28.21	29	29.00	0.00	0.09	25.64
ug4-80		101		27	25.20	582.50	30.77	29	28.19	0.40	0.08	25.64
ug5-80		94		27	25.80	190.52	30.77	29	28.48	0.50	0.07	25.64
ug1-90		102		32	29.80	150.40	27.27	33	33.00	0.00	0.09	25.00
ug2-90		114		35	34.60	469.62	20.45	36	36.00	0.00	1.17	18.18
ug3-90	90	108	44	31	29.80	316.29	29.55	33	32.23	0.42	0.07	25.00
ug4-90		99		34	33.30	381.20	22.73	35	35.00	0.00	0.18	20.45
ug5-90		104		34	31.80	815.47	22.73	35	35.00	0.00	0.25	20.45
ug1-100		114		34	32.20	499.20	30.61	38	35.71	1.14	0.08	22.45
ug2-100		114		40	38.40	715.90	18.37	41	41.00	0.00	0.64	16.33
ug3-100	100	116	49	33	31.60	256.40	32.65	36	34.35	0.60	0.08	26.53
ug4-100		122		35	33.80	419.80	28.57	36	36.00	0.00	0.12	26.53
ug5-100		125		31	30.60	693.65	36.73	35	32.55	1.01	0.05	28.57
				27.23	25.84	257.79	25.04	28.73	28.33	0.20	3.39	20.85

Tabla 4.7: Comparativa de desempeño entre MAAMP y RS-MAG sobre 30 grafos del conjunto Rome.

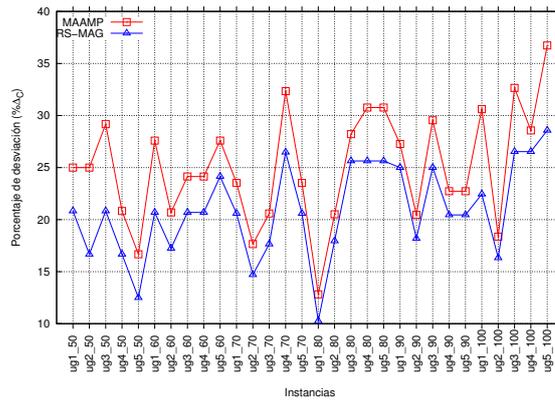


Figura 4.4: Comparación del desempeño entre MAAMP y RS-MAG con respecto al límite inferior teórico sobre 30 grafos del conjunto Rome.

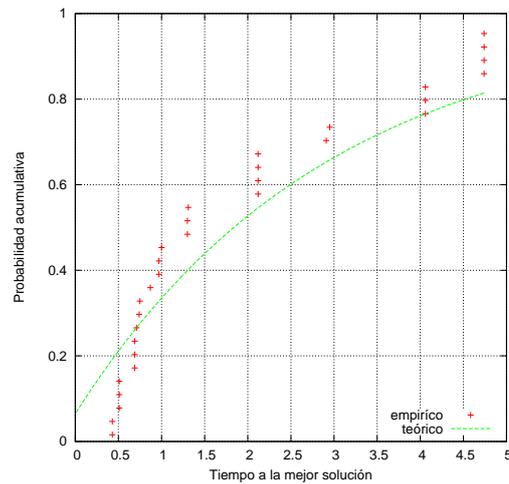


Figura 4.5: Gráfica de *Tiempo a la mejor solución* para la instancia ug1-80.

4.6.2 Resultados de instancias tipo malla

GRASP+evPR es el algoritmo contra el cual se realizan las comparaciones, propuesto por Duarte *et al.* (2011).

En el caso particular de instancias de grafos tipo malla se obtienen resultados óptimos con el método de inicialización que se utiliza en el RS-MAG. En algunos casos se muestra una desviación estándar mayor a cero, esto debido a que el proceso de inicialización genera una estructura en niveles y dependiendo en que nivel comience a etiquetar puede llegar al óptimo

o por otro lado quedarse a una unidad.

En la Tabla 4.8 se presentan los resultados obtenidos de RS-MAG sobre las instancias tipo malla. Con los datos que presenta se puede llegar a la conclusión de que RS-MAG ofrece mejores soluciones en las 24 instancias de tipo malla, teniendo un promedio de los mejores valores del AB de 273.96 contra un 268.00 del obtenido por GRASP+evPR.

Grafo	V	E	C*	GRASP+evPR			RS-MAG				
				C	T	% Δ_C	C	Prom.	Desv.	T	% Δ_C
mesh9x9	81	144	36	36	2.54	0.00	36	36.00	0.00	0.01	0.00
mesh50x2	100	148	49	48	3.85	2.04	49	49.00	0.00	0.01	0.00
mesh25x4	100	171	48	46	4.44	4.17	48	48.00	0.00	0.01	0.00
mesh20x5	100	175	48	46	4.39	4.17	48	47.75	0.43	0.01	0.00
mesh10x10	100	180	45	45	4.33	0.00	45	45.00	0.00	0.01	0.00
mesh34x3	102	167	50	48	4.62	4.00	50	49.63	0.48	0.01	0.00
mesh17x6	102	181	48	47	4.66	2.08	48	48.00	0.00	0.01	0.00
mesh13x8	104	187	48	47	4.55	2.08	48	48.00	0.00	0.01	0.00
mesh15x7	105	188	49	48	5.01	2.04	49	49.00	0.00	0.01	0.00
mesh12x9	108	195	50	49	5.08	2.00	50	49.41	0.49	0.01	0.00
mesh11x11	121	220	55	55	6.95	0.00	55	55.00	0.00	0.01	0.00
mesh12x12	144	264	66	66	10.66	0.00	66	66.00	0.00	0.01	0.00
mesh130x7	910	1683	452	442	1816.21	2.21	452	451.43	0.50	0.01	0.00
mesh120x8	960	1792	476	468	1968.86	1.68	476	476.00	0.00	0.01	0.00
mesh110x9	990	1861	491	482	2146.27	1.83	491	490.50	0.50	0.01	0.00
mesh100x10	1000	1890	495	485	2227.91	2.02	495	495.00	0.00	0.01	0.00
mesh50x20	1000	1930	490	474	1809.99	3.27	490	490.00	0.00	0.01	0.00
mesh40x25	1000	1935	488	480	1866.68	1.64	488	487.25	0.43	0.01	0.00
mesh60x17	1020	1963	502	481	2030.33	4.18	502	501.31	0.46	0.01	0.00
mesh34x30	1020	1976	495	491	1751.98	0.81	495	495.00	0.00	0.01	0.00
mesh80x13	1040	1987	514	500	2609.57	2.72	514	513.69	0.46	0.01	0.00
mesh70x15	1050	2015	518	502	2826.23	3.09	518	517.56	0.50	0.01	0.00
mesh90x12	1080	2058	534	522	2829.10	2.25	534	534.00	0.00	0.01	0.00
mesh33x33	1089	2112	528	524	1968.86	0.76	528	528.00	0.00	0.01	0.00
Avg.				268.00	1079.71	2.04	273.96	273.77	0.18	0.01	

Tabla 4.8: Comparación del desempeño entre GRASP+evPR y RS-MAG sobre 24 grafos tipo malla.

Observando las columnas que denotan el porcentaje de desviación al límite inferior teórico ($\% \Delta_C$) de ambos algoritmos se demuestra que RS-MAG no solo obtiene mejores soluciones que GRASP+evPR, además son las soluciones óptimas. En todos los casos para el algoritmo RS-MAG se obtiene un porcentaje de desviación al límite inferior teórico ($\% \Delta_C$) igual a cero. GRASP+evPR consume demasiado tiempo de cómputo, en promedio 1079.71 segundos, mientras RS-MAG consume 0.18 segundos en promedio. Se observa que para las 12 instancias con mayor número de nodos, GRASP+evPR tiende a elevar los tiempos de cómputo. Pasa de 10.66 segundos de la instancia *mesh12x12* a 1816.21 segundos de la instancia *mesh130x7*. RS-MAG es rápido debido al método de inicialización basado en BFSi, el cual está descrito en

la Sección 3.2.

En la Figura 4.6 se puede observar que en todos los casos RS-MAG ofrece mejores soluciones que el algoritmo GRASP+evPR. Lo anterior exceptuando a las instancias *mesh9x9*, *mesh10x10*, *mesh11x11* y *mesh12x12* en donde ambos algoritmos encuentran la mejor solución conocida.

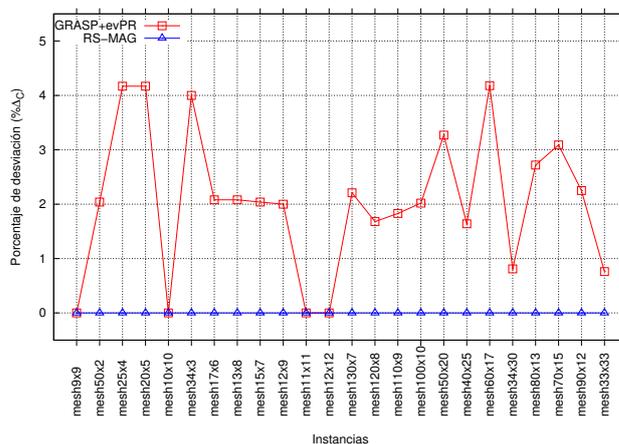


Figura 4.6: Comparación del desempeño entre GRASP+evPR y RS-MAG con respecto al límite inferior teórico sobre 24 grafos tipo malla.

4.6.3 Resultados de instancias Harwell-Boeing

El algoritmo contra el cual se realizan las comparaciones es el publicado por Duarte *et al.* (2011), llamado GRASP+evPR.

Se observa que en 10 casos se igualan los resultados previamente reportados por Duarte *et al.* (2011). Con respecto a los resultados obtenidos sobre la instancia *impcol_d* se logró mejorar los resultados obteniendo una desviación estándar de 0.56. Otro de los resultados interesantes es el caso de la instancia *685_bus*, RS-MAG iguala el resultado anteriormente reportado en un tiempo de 23.10 segundos en promedio contra 594.70 segundos del algoritmo GRASP+evPR.

En términos generales, RS-MAG se encuentra a solo 3.35% de diferencia entre el promedio de porcentaje de desviación al límite inferior teórico que presenta GRASP+evPR.

La Figura 4.7 presenta las 12 instancias más pequeñas del conjunto Harwell-Boeing. Se observa claramente que en las instancias *ash85*, *nos4* y *dtw_234* no se logra llegar a los

Grafo	V	E	C*	GRASP+evPR			RS-MAG				
				C	T	% Δ_C	C	Prom.	Desv.	T	% Δ_C
pores_1	30	103	8	6	0.98	25.00	6	6.00	0.00	0.74	25.00
ibm32	32	90	9	9	0.93	0.00	9	8.42	0.49	89.41	0.00
bcpwr01	39	46	17	17	0.63	0.00	17	16.81	0.40	147.64	0.00
bcsstk01	48	176	11	8	2.91	27.27	8	8.00	0.00	478.91	27.27
bcpwr02	49	59	22	21	1.06	4.55	21	20.74	0.44	45.31	4.55
curtis54	54	124	13	12	2.17	7.69	13	12.10	0.30	164.09	0.00
will57	57	127	14	13	2.37	7.14	13	13.00	0.00	64.81	7.14
impcol_b	59	281	14	8	6.10	42.86	8	8.00	0.00	451.84	42.86
ash85	85	219	27	22	7.47	18.52	21	20.71	0.45	614.93	22.22
nos4	100	247	47	35	10.16	25.53	33	31.81	0.64	464.69	29.79
dwt_234	117	162	58	51	8.43	12.07	50	47.29	1.37	86.04	13.79
bcpwr03	118	179	57	39	8.62	31.58	39	39.00	0.00	19.28	31.58
bcsstk06	420	3720	210	33	1184.72	84.29	6	3.10	1.00	1862.19	97.14
bcsstk07	420	3720	210	33	1160.86	84.29	5	3.10	0.89	1639.13	97.62
impcol_d	425	1267	212	105	409.19	50.47	106	105.52	0.56	363.31	50.00
can_445	445	1682	221	85	663.90	61.54	76	72.23	2.11	2160.66	65.61
494_bus	494	586	247	228	375.61	7.69	220	218.84	1.30	1438.73	10.93
dwt_503	503	2762	250	58	1045.85	76.80	25	24.06	0.62	2145.73	90.00
sherman4	546	1341	272	261	1568.87	4.04	241	238.84	1.39	2078.78	11.40
dwt_592	592	2256	295	112	1321.90	62.03	95	88.00	3.42	2218.73	67.80
662_bus	662	906	331	220	662.06	33.53	220	219.19	0.40	145.39	33.53
nos6	675	1290	337	328	2303.03	2.67	299	296.84	0.88	2393.58	11.28
685_bus	685	1282	342	136	594.70	60.23	136	136.00	0.00	23.10	60.23
can_715	715	2975	357	121	2259.50	66.11	84	80.10	2.68	2360.69	76.47
Avg.				81.71	566.75	33.16	72.96	71.57	0.81	894.07	36.51

Tabla 4.9: Comparativa del desempeño entre GRASP+evPR y RS-MAG sobre 24 grafos Harwell-Boenig.

resultados reportados. Sin embargo, se mejora el resultado de la instancia *curtis54*. En el resto de las instancias se logra igualar los resultados que se han reportado en la literatura. Los resultados obtenidos por RS-MAG que no se lograron mejorar, están muy cercanos a los valores reportados. Lo anterior es notorio en la gráfica, por la cercanía entre las dos líneas.

La Figura 4.8 muestra las 12 instancias más grandes del conjunto Harwell-Boeing. Se observa que en dos casos se igualan los resultados publicados previamente. RS-MAG ofrece mejor resultado en el caso de la instancia *impcol_d*, en la gráfica no se aprecia tan claramente esto, sin embargo, En la Tabla 4.9 se puede observar una mejora de RS-MAG de 0.47% (50.47% vs. 50.00%).

En la Figura 4.9 se puede observar que en 27 casos de las 31 corridas(87.09%), RS-MAG encuentra una buena solución antes de los 30 segundos. Solo en el 9.67% de las 31 corridas se observa una dispersión considerable con respecto al resto de las ejecuciones. En esta gráfica se observa también que el comportamiento en las práctica está muy apegado a la estimación teórica.

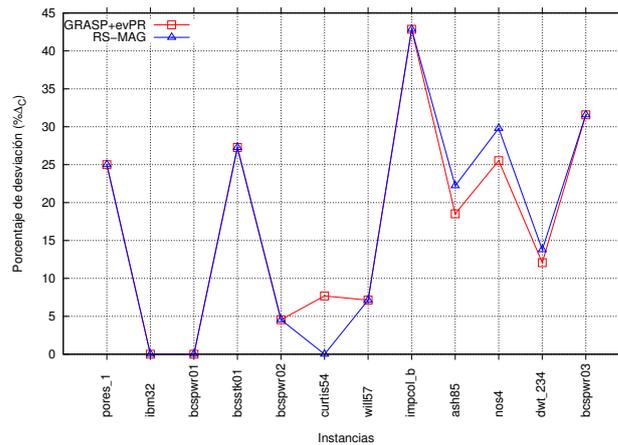


Figura 4.7: Comparación del desempeño entre GRASP+evPR y RS-MAG sobre 12 grafos pequeños Harwell-Boeing

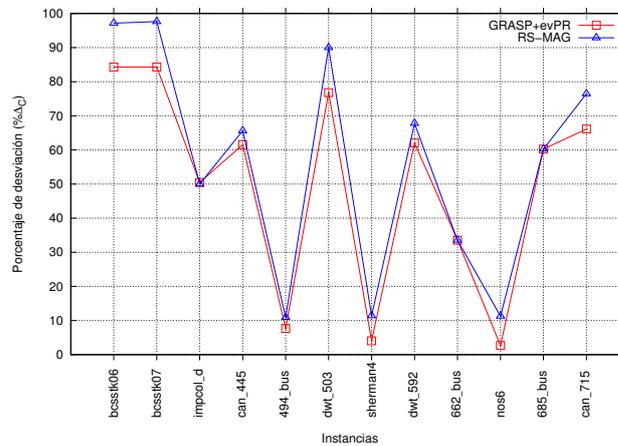


Figura 4.8: Comparación del desempeño entre GRASP+evPR y RS-MAG sobre 12 grafos grandes Harwell-Boeing.

Resumen del capítulo

En este capítulo se presentaron los diferentes conjuntos de instancias utilizados. Además se mostraron las pruebas preliminares realizadas sobre grafos con óptimo conocido para definir los posibles valores de los parámetros de entrada de RS-MAG. Se indicó como se eligieron las funciones de vecindad y que método de solución inicial se utilizó.

Una vez definidos los posibles valores de entrada se mostró como se obtuvo una correcta combinación de parámetros para que tuviera un mejor desempeño el algoritmo. Lo anterior mediante la utilización de un arreglo de cobertura que indica las pruebas necesarias con el

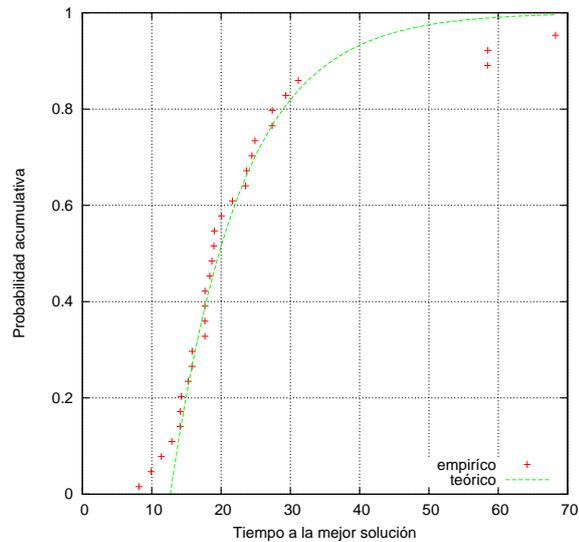


Figura 4.9: Gráfica *Tiempo a la mejor solución* para la instancia 685_bus.

objetivo de no realizar de manera exhaustiva todas las posibles combinaciones de parámetros. La sintonización está inspirada en las pruebas de interacción combinatoria (Cohen *et al.*, 1996). También se presentaron los resultados de la experimentación final para los conjuntos de instancias de grafos tipo malla, los grafos de la colección Rome así como del conjunto Harwell-Boeing. Para el análisis de los resultados, nos apoyamos de diferentes tipos de gráficas con el propósito de hacer notar que método es mejor dependiendo de las características de cada instancia.

En el siguiente capítulo se plantea un caso de estudio del mundo real con el objetivo de ofrecer un panorama de las posibles aplicaciones que se le puede dar a RS-MAG.

5

Aplicación práctica del problema de MAG

Introducción

El objetivo de este capítulo es hacer notar que resolviendo una instancia del problema de MAG se pueden solucionar problemas prácticos que se presentan en el mundo real. En el caso particular de la asignación de frecuencias se busca encontrar la menor interferencia posible durante las transmisiones.

A continuación se presentan los conceptos básicos de la teoría de radiotransmisión. Lo anterior con el fin de comprender mejor el caso de estudio de asignación de frecuencias presentado en este capítulo.

5.1 Introducción a la radiotransmisión

Las ondas de radio son fáciles de generar, viajan grandes distancias y penetran edificios sin problemas. Por ello, las ondas de radio son muy utilizadas en el área de comunicaciones. Las ondas de radio son omnidireccionales, es decir, viajan a todas direcciones a partir de la fuente que las emite. Por lo anterior, no es necesario que el emisor y receptor estén alineados (Tanenbaum, 2003).

Las propiedades de las ondas de radio pueden variar dependiendo de la cantidad de oscilaciones por segundo, es decir, su frecuencia. Las ondas que viajan a baja frecuencia atraviesan bien casi cualquier obstáculo, el inconveniente es que la potencia se reduce drásticamente a medida que se alejan del emisor. A frecuencias altas, las ondas de radio tienden a viajar en línea recta y rebotar con los obstáculos (Tanenbaum, 2003).

Uno de los problemas a los que se enfrenta este medio de comunicación, es la interferencia. La interferencia es la superposición de dos o más ondas en un punto, a este fenómeno se le conoce como el principio de la superposición. Existen diversos estudios sobre el uso de la radio frecuencia para la transmisión de información (Hall, 1980, Rappaport y Sandhu, 1994).

5.2 Caso de estudio

El problema de asignación de frecuencias (Hale, 1980) se puede modelar mediante un grafo donde los nodos representan un conjunto de n emisoras. En este grafo las arcos conectan emisoras que pueden interferir entre ellas y el etiquetado de los nodos indica las frecuencias asignadas a cada emisora. La distancia entre frecuencias corresponde a la mínima diferencia en valor absoluto entre etiquetas de nodos adyacentes. Por lo tanto para resolver el problema de asignación de frecuencias se debe encontrar un etiquetado φ^* que maximice la mínima diferencia absoluta entre etiquetas de nodos adyacentes (ver Ecuación 2.2).

La empresa de autoservicio llamada OXXO, requiere establecer comunicación entre sus diferentes sucursales y algunas de sus bodegas distribuidoras. Uno de los problemas a los que se enfrenta es a la interferencia entre frecuencias de las antenas emisoras. Para tratar de solucionar dicho problema la empresa contrata personal para que realice una configuración adecuada de las frecuencias. Sin embargo, el realizar cálculos y desplegar la nueva configuración requiere de tiempo. Mientras se encuentra una buena solución para mejorar el sistema de radiofrecuencia existen muchos retrasos de abastecimiento debido a la mala comunicación.

En la Figura 5.1 se muestra el mapa de Cd. Victoria, Tamaulipas¹. Se pueden observar puntos en color rojo que indican la presencia de antenas emisoras.

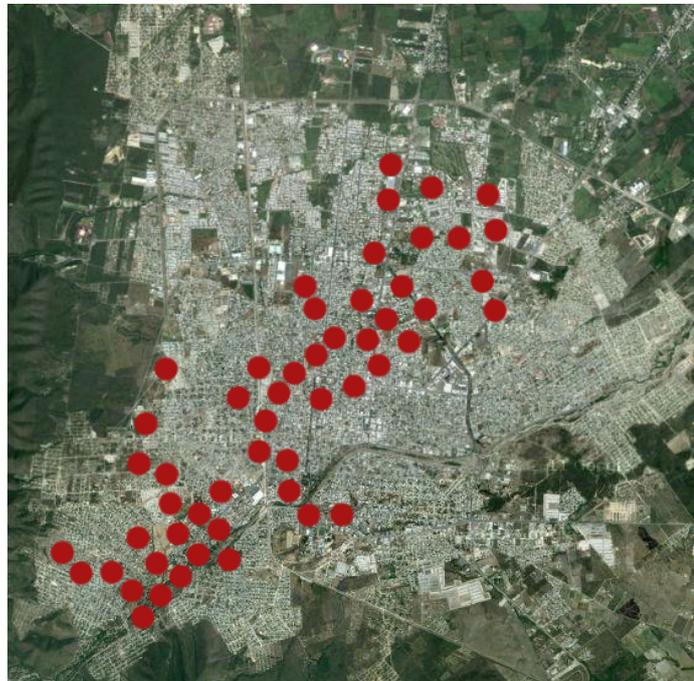


Figura 5.1: Mapa de Cd. Victoria, Tamaulipas.

En la Figura 5.2 se presenta el grafo que se modela a partir de las ubicaciones de las antenas (Figura 5.1). Cada arco denota entre que nodos existe interferencia. Resolver este problema es equivalente a encontrar solución al problema de MAG. La Figura 5.2 muestra un etiquetado φ producido manualmente con un valor de $AB = 1$. Es una de las peores soluciones que se pueden obtener, ya que se busca maximizar el valor del AB .

¹Imagen tomada de Google Maps 2011.

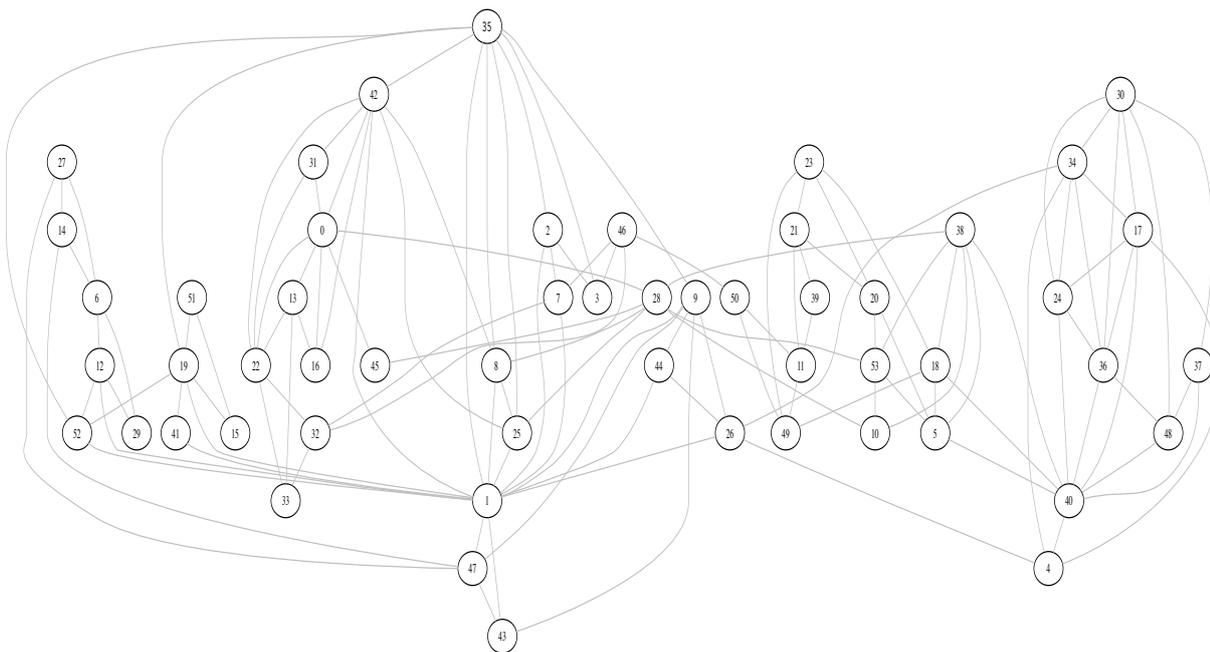
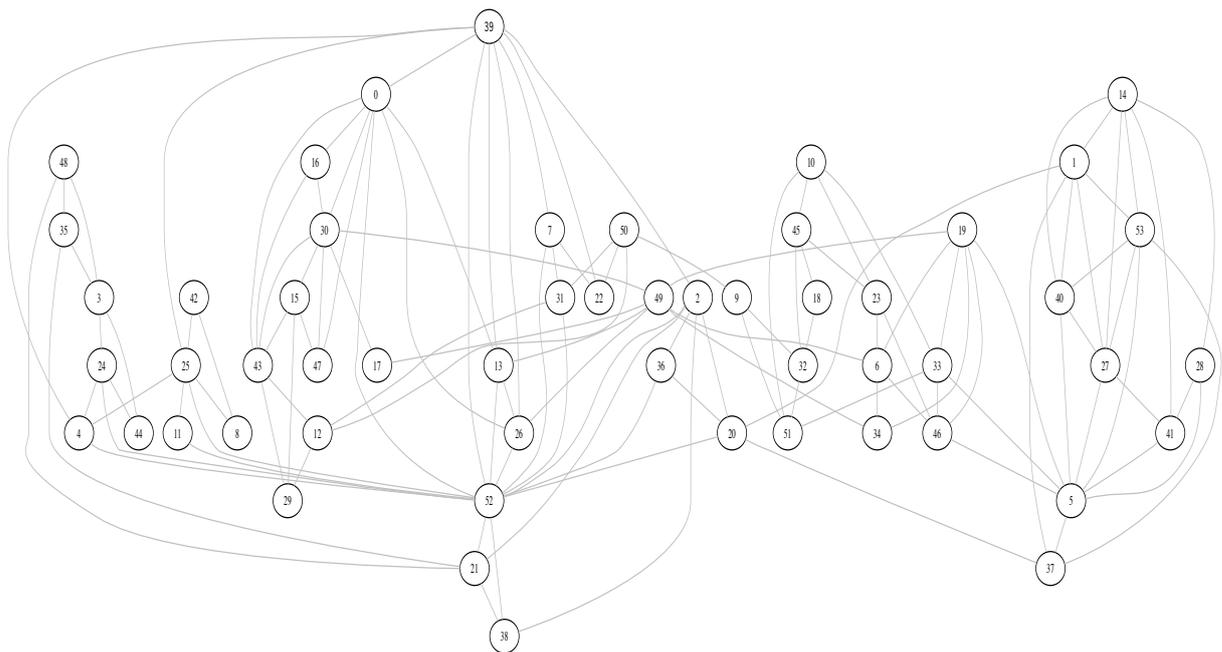


Figura 5.2: Grafo con $AB(G, \varphi) = 1$.

Para ofrecer una solución de calidad al problema de las interferencias, se ejecutó RS-MAG 31 veces. La instancia utilizada fue el grafo generado a partir de las ubicaciones de las antenas donde los arcos representan las interferencias entre ellas. La mejor solución se encontró en un tiempo promedio de 164 segundos. El AB promedio de todas las ejecuciones fue de 12.10 y presentó una desviación estándar de 0.30. Los parámetros utilizados por RS-MAG son los mejores obtenidos en la etapa de sintonización para las instancias del conjunto Harwell-Boeing vistos en la Tabla 4.5.

La Figura 5.3 muestra el grafo con un etiquetado φ^* producido por el algoritmo RS-MAG que se presenta en este trabajo de tesis. El etiquetado tiene un valor de AB de 13. Dicho etiquetado es mucho mejor ya que reduce considerablemente la interferencia entre emisoras vecinas.

Por ejemplo, mientras que con el etiquetado φ que se muestra en la Figura 5.2 los nodos ubicados en la parte superior derecha con las etiquetas 30 y 34 tienen una diferencia de 4. Los mismos nodos con las etiquetas generadas por RS-MAG (Figura 5.3) tienen ahora una diferencia de 13. Algo similar sucede con el resto de las emisoras.

Figura 5.3: Grafo con $AB(G, \varphi) = 13$.

Resumen del capítulo

En este capítulo se mostró una introducción a la radiotransmisión para posteriormente plantear el problema de interferencia entre antenas emisoras. Se presentó el grafo generado a partir de las ubicaciones de las antenas emisoras en Cd. Victoria de la cadena de tiendas de autoservicio OXXO.

Se contrastaron dos etiquetados diferentes. Un etiquetado calculado de forma manual y un etiquetado producido por RS-MAG. Se observó que RS-MAG proporciona una mejor solución en menos tiempo que calculando la solución de forma manual. Al ahorrar tiempo y encontrar una buena asignación de etiquetas, el beneficio se traduce en más ventas. Las sucursales y bodegas distribuidoras siempre estarán en constante comunicación y no habrá problemas de abastecimiento de mercancía.

En el siguiente capítulo se presentan las conclusiones de este trabajo de tesis. Analizando los resultados obtenidos de RS-MAG sobre los diferentes conjuntos de prueba. Además se abordan algunas posibilidades de trabajo futuro.

6

Conclusiones y trabajo futuro

En este trabajo de tesis se estudiaron los diferentes métodos existentes que ofrecen solución al problema de MAG. La metaheurística que hemos seleccionado para implementar y dar solución al problema de MAG es la de recocido simulado, la cual hemos nombrado RS-MAG. Se mostraron los componentes principales de nuestro algoritmo: método para generar la solución inicial, función de vecindad, criterio de aceptación de movimientos, inicialización de la temperatura, criterio para definir la longitud de la cadena de Markov, esquema de enfriamiento y la condición de paro. Además se utilizó una metodología basada en pruebas de interacción combinatoria que utiliza arreglos de cobertura para sintonizar todos los parámetros de entrada de RS-MAG y así lograr obtener un mejor desempeño de nuestro algoritmo.

Una vez concluida esta investigación se realizó un análisis sobre los resultados obtenidos. Enseguida se desglosó el análisis por cada conjunto de instancias con los que se trabajó. Posteriormente se describió el posible trabajo futuro con el fin de mejorar RS-MAG y seguir contribuyendo al estado del arte del problema de MAG.

6.1 Conclusiones

Con los resultados obtenidos, se concluye que el recocido simulado implementado en esta tesis es un método muy competitivo con respecto a los algoritmos reportados en la literatura. Por lo anterior esta tesis aporta al estado del arte del problema de MAG un nuevo algoritmo para su resolución. En la Figura 6.1 se observan todos los resultados concentrados en una gráfica circular. Se realizaron las pruebas con un total de 78 instancias: 24 mallas, 24 instancias del conjunto Harwell-Boeing y 30 del conjunto Rome. Es fácil visualizar que en 52 instancias (66.67%) RS-MAG supera las mejores soluciones previamente conocidas. En 14 grafos se lograron igualar estos resultados lo que equivale al 17.95%. Solo en el caso de las 12 instancias restantes (15.38%) no se lograron igualar los mejores resultados reportados en la literatura.

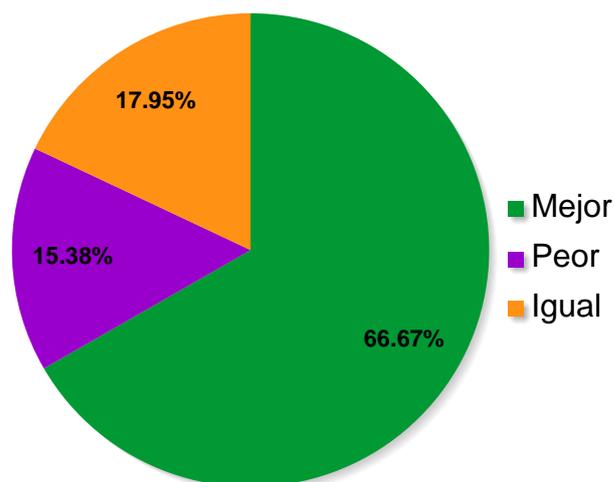


Figura 6.1: Resultados obtenidos por RS-MAG sobre el total de instancias estudiadas.

En la Figura 6.2 se presenta una gráfica donde el eje de las ordenadas representa el número de instancias. El eje de las abscisas muestra los diferentes conjuntos de pruebas utilizados en la experimentación final. A su vez cada conjunto está dividido en tres partes que corresponde a los resultados con mejor calidad, igual y de peor calidad respectivamente.

A continuación se analizan a detalle los resultados de cada uno de estos conjuntos de pruebas.

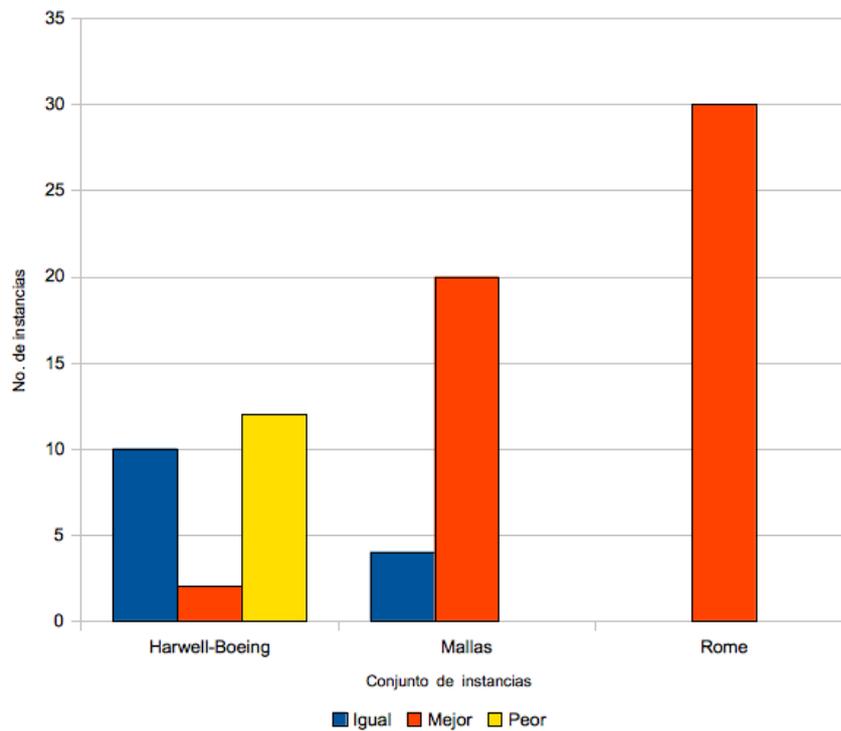


Figura 6.2: Resultados obtenidos por RS-MAG sobre los tres conjuntos de instancias analizadas.

6.1.1 Conjunto Rome

Como puede observarse en la gráfica de barras de la Figura 6.2 para el caso de las instancias del conjunto Rome, RS-MAG logró mejorar consistentemente los resultados (100% de las 30 instancias) obtenidos por Bansal y Srivastava (2011) con su algoritmo MAAMP.

Aunque los resultados alcanzados por RS-MAG son mejores con respecto a los producidos por MAAMP, aún existe la posibilidad de mejorarlos debido a que todavía no se llega a los límites teóricos para esas 30 instancias.

6.1.2 Conjunto de grafos tipo malla

Los resultados que se obtuvieron de las ejecuciones de RS-MAG sobre el conjunto de instancias de tipo malla fueron satisfactorios tanto en calidad de solución como en tiempo de cómputo.

En la gráfica de barras de la Figura 6.2 se observa que para el caso de las 24 instancias

tipo malla los resultados generados por el algoritmo GRASP+evPR son igualados por RS-MAG en 4 casos(16.67%). Cabe mencionar que en estos 4 grafos los dos algoritmos obtienen las soluciones óptimas. Por otra parte en las otras 20 instancias de este conjunto (83.33%) los resultados alcanzados por RS-MAG fueron mejores a los reportados por Duarte *et al.* (2011) con su algoritmo GRASP+evPR.

6.1.3 Conjunto Harwell-Boeing

En nuestra investigación observamos que las instancias del conjunto Harwell-Boeing son más complicadas de resolver. Lo anterior se hace evidente por los resultados que se reportan previamente con el algoritmo GRASP+evPR de Duarte *et al.* (2011). Dichos resultados están muy lejos de las soluciones óptimas teóricas y en particular para las 12 instancias de mayor tamaño. A pesar de esto, con la ejecución de RS-MAG se lograron mejorar algunos resultados publicados sobre instancias de este conjunto.

En la Figura 6.2 se presenta una gráfica de barras donde se muestran los resultados obtenidos por RS-MAG sobre las 24 instancias del conjunto Harwell-Boeing. El 8.33% de los resultados son mejores que los reportados por GRASP+evPR (Duarte *et al.*, 2011). En 10 casos (41.67%) los resultados obtenidos por el algoritmo de referencia GRASP+evPR son igualados por RS-MAG son igualados mientras que en los 12 casos restantes (50.00%) nuestro algoritmo no logra igualar estos resultados.

Nosotros creemos que el motivo por el cual no se obtienen mejores resultados sobre los 12 grafos más grandes, es la densidad de arcos así como el tamaño y estructura. Dos grafos pueden tener la misma densidad de arcos pero pueden tener una estructura diferente. Es posible que ciertos métodos ofrezcan mejores soluciones para un cierto tipo de grafo y para otros no.

6.2 Trabajo futuro

Una vez concluido el trabajo de tesis podemos proponer algunas posibilidades de trabajo futuro. Para que RS-MAG pueda ser mejorado es necesario realizar un análisis adicional de sus componentes clave.

- **Solución inicial:** Se propone realizar un análisis sobre las instancias y sacar provecho de su estructura para diseñar un método diferente a los presentados en este trabajo de tesis. Implementar diversos tipos de recorridos como por ejemplo búsqueda primero en profundidad.
- **Funciones de vecindad:** Estudiar los vecindarios propuestos para mejorarlos. Además de proponer nuevas funciones y diferentes maneras de combinarlas. Sacando provecho de las características de los grafos, como la densidad de arcos y la estructura
- **Esquemas de enfriamiento:** Implementar y experimentar con esquemas de enfriamiento diferentes a los utilizados como: el esquema propuesto por Lundy y Mees (1986), enfriamiento de Cauchy (Szu y Hartley, 1987), por mencionar algunos. Además de implementar esquemas de enfriamiento adaptativos, en donde se tome en cuenta el comportamiento del algoritmo durante la ejecución. Lo anterior con el propósito de realizar un estudio comparativo y observar la influencia que ejerce sobre las soluciones finales
- **Longitud de Cadena de Markov:** Implementar un método que provea longitudes de la cadena de Markov dinámicas, tomando en cuenta los movimientos aceptados y rechazados durante la ejecución
- **Criterio de paro:** Estudiar diferentes condiciones de paro, en especial aquellas que toman en cuenta el comportamiento de los movimientos aceptados durante la experimentación

Bibliografía

- B. Adenso-Diaz y M. Laguna. Fine-tuning of algorithms using fractional experimental designs and local search. *Operations Research*, 54:99–114, 2006.
- R. M. Aiex, M. G. C. Resende y C. C. Ribeiro. Probability distribution of solution time in grasp: An experimental investigation. *Journal of Heuristics*, 8, 2002.
- A. Albrecht y Chak-Kuen Wong. On logarithmic simulated annealing. In *Theoretical Computer Science: Exploring New Frontiers of Theoretical Informatics*, volume 1872 of *Lecture Notes in Computer Science*, pages 301–314. Springer Berlin / Heidelberg, 2000.
- R. Bansal y K. Srivastava. Memetic algorithm for the antibandwidth maximization problem. *Journal of Heuristics*, 17:39–60, 2011.
- C. Blum y A. Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, 35(3):268–308, September 2003.
- R. E. Burkard, H. Donnani, Y. Lin y G. Rote. The obnoxious center problem on a tree. *SIAM Journal on Computing*, 14(4):498–509, 2001.
- T. Calamoneri, A. Missini, L. Török y I. Vrt'ó. Antibandwidth of complete k-ary trees. *Electronic Notes in Discrete Mathematics*, 24:259–266, 2006.
- P. Cappanera. A survey on obnoxious facility location problems. Technical report, Uni. di Pisa, 1999.
- M. F. Cardoso, R. L. Salcedo y S. F. de Azevedo. Nonequilibrium simulated annealing: A faster approach to combinatorial minimization. *Industrial & Engineering Chemistry Research*, 33(8):1908–1918, 1994.
- V. Cerny. A thermodynamic approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45(1):41–51, 1985.

- D. M. Cohen, S. R. Dalal, J. Parelius y G. C. Patton. The combinatorial design approach to automatic test generation. *IEEE Software*, 13:83–88, 1996.
- Ch. Darwin. *On the Origin of Species*. John Murray (London), 1859.
- W.A. de Landgraaf, A.E. Eiben y V. Nannen. Parameter calibration using meta-algorithms. In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pages 71 –78, 2007.
- S. Dobrev, R. Královic, D. Pardubská, L. Török y I. Vrt'ó. Antibandwidth and cyclic antibandwidth of hamming graphs. *Electronic Notes in Discrete Mathematics*, 34:295–300, 2009.
- M. Dorigo. *Optimization, Learning and Natural Algorithm*. PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, 1992.
- A. Duarte, R. Martí, M. G. C. Resende y R. M. A. Silva. GRASP with path relinking heuristics for the antibandwidth problem. *Networks*, 58(3):171–189, 2011.
- M. A. Saleh Elmohamed, G. Fox y P. Coddington. A comparison of annealing techniques for academic course scheduling. In *Lecture Notes in Computer Science*, pages 92–112. Springer-Verlag, 1998.
- T. A. Feo y M. G. C. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6:109–133, 1995.
- M. Fielding. Simulated annealing with an optimal fixed temperature. *SIAM Journal on Optimization*, 11(2):289–307, 2000.
- B. L. Fox y G. W. Heine. Simulated annealing with overrides. Technical report, University of Colorado, 1993.
- M. R. Garey y D. S. Johnson. *Computers and Intractability: A guide to the theory of NP-Completeness*. W.H. Freeman and Company, New York, 1979.
- F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, 13(5):533–549, 1986.

- L. Gonzalez-Hernandez y J. Torres-Jimenez. MiTS: A new approach of tabu search for constructing mixed covering arrays. *Lecture Notes in Artificial Intelligence*, 6438:382–392, 2010.
- A. Gunawan, H. C. Lau y L. Lindawati. Fine-tuning algorithm parameters using the design of experiments. In *Lecture Notes in Computer Science*, volume In press, 2011.
- W. K. Hale. Frequency assignment: Theory and applications. *Proceedings of the IEEE*, 68 (12):1497–1514, December 1980.
- M. P. M. Hall. Effects of the troposphere on radio communication. *Stevenage Herts England Peter Peregrinus Ltd IEE Electromagnetic Waves Series*, 8, 1980.
- D. Henderson, A. Johnson y J. Sheldon. Chapter 10 the theory and practice of simulated annealing. *Theory and Practice*, 57:287–319, 2003.
- L. Herault. Rescaled simulated annealing. In *Neural Networks Proceedings, 1998. IEEE World Congress on Computational Intelligence. The 1998 IEEE International Joint Conference on*, volume 2, pages 1239 –1244 vol.2, may 1998.
- J. Holland. *Adaptation in natural and artificial systems*. The University of Michigan Press, 1975.
- D. S. Johnson, C. R. Aragon, L. A. McGeoch y C. Schevon. Optimization by simulated annealing: an experimental evaluation. part i, graph partitioning. *Operations Research*, 37: 865–892, October 1989.
- S. Kirkpatrick, C. Gelatt y M. Vecchi. Optimization by simulated annealing. *Science*, 220: 671–680, May 1983.
- W. Krauth. *Statistical mechanics: Algorithms and computations*. Oxford University Press, 2006.
- P. J. M. Laarhoven y E. H. L. Aarts. *Simulated annealing: Theory and applications*. Kluwer Academic Publishers, 1987.
- J. Leung, O. Vornberger y J. Witthoff. On some variants of the bandwidth minimization problem. *SIAM Journal on Computing*, 13(3):650–667, 1984.

- M. Lundy y A. Mees. Convergence of an annealing algorithm. *Mathematical Programming*, 34:111–124, 1986.
- F. Melício, J. P. Caldeira y A. C. Rosa. Timetabling implementation aspects by simulated annealing. In *Proceedings of the IEEE- ICSSSE'98*, Beijing, China, 1998.
- N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller y E. Teller. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6): 1087–1092, 1953.
- Z. Miller y D. Pritikin. On the separation number of a graph. *Networks*, 19:651–666, 1989.
- P. Moscato. An introduction to population approaches for optimization and hierarchical objective functions: A discussion on the role of tabu search. *Annals of Operations Research*, 41:85–121, 1993. ISSN 0254-5330.
- P. Moscato. Memetic algorithms: A short introduction. In *New Ideas in Optimization*, pages 219–234. McGraw-Hill, London, 1999.
- Y. Nourani y B. Andresen. A comparison of simulated annealing cooling strategies. *Journal of Physics A: Mathematical and General*, 31(41):8373, 1998.
- R.H.J.M. Otten y L.P.P.P. van Ginneken. Stop criteria in simulated annealing. In *Computer Design: VLSI in Computers and Processors, 1988. ICCD '88., Proceedings of the 1988 IEEE International Conference on*, pages 549 –552, oct 1988.
- H. Papadimitriou y K. Steiglitz. *Combinatorial optimization: Algorithms and complexity*. Dover Publications, 1998.
- T.S. Rappaport y S. Sandhu. Radio-wave propagation for emerging wireless personal-communication systems. *Antennas and Propagation Magazine, IEEE*, 36(5):14 –24, 1994.
- A. Raspaud, H. Schröder, O. Sykora, L. Török y I. Vrt'o. Antibandwidth and cyclic antibandwidth of meshes and hypercubes. *Discrete Mathematics*, 309(11):3541–3552, June 2009. doi: 10.1016/j.disc.2007.12.058.

- Real Academia Española. *Ortografía de la lengua española*. Espasa, 2010.
- F. S. Roberts. New directions in graph theory. *Annals of Discrete Mathematics*, 55:13–44, 1993.
- E. Rodríguez-Tello y J. Torres-Jimenez. Memetic algorithms for constructing binary covering arrays of strength three. *Lecture Notes in Computer Science*, 5975:86–97, 2010.
- E. Rodríguez-Tello, J. K. Hao y J. Torres-Jimenez. An effective two-stage simulated annealing algorithm for the minimum linear arrangement problem. *European Journal of Operational Research*, 185(3):1319–1335, 2008.
- F. Romeo y A. Sangiovanni-Vincentelli. A theoretical framework for simulated annealing. *Algorithmica*, 6:302–345, 1991. ISSN 0178-4617.
- Y. Sankar. Ultra-fast automatic placement for FPGAs. Master's thesis, University of Toronto, 1999.
- H. Szu y R. Hartley. Fast simulated annealing. *Physics Letters A*, 122(3-4):157 – 162, 1987.
- E. Talbi. *Metaheuristics: From design to implementation*. Jhon Wiley & Sons, Inc., 2009.
- A. S. Tanenbaum. *Redes de computadoras*. Prentice Hall, 2003.
- L. Török. Antibandwidth of three-dimensional meshes. *Electronic Notes in Discrete Mathematics*, 28:161–167, March 2007.
- J. Torres-Jimenez y E. Rodríguez-Tello. Simulated annealing for constructing binary covering arrays of variable strength. In *Proceedings of the CEC 2010*, pages 4102–4109, Barcelona, Spain, 2010.
- X. Wang, X. Wu y S. Dimitrescu. On explicit formulas for bandwidth and antibandwidth of hypercubes. *Discrete Applied Mathematics*, 157(8):1947–1952, 2009.
- W. Yao, Z. Ju y L. Xiaoxu. Dual bandwidth of some special trees. *Zhengzhou University Natural Science*, 2003. ISSN 35.

- L. Yixun y Y. Jinjiang. The dual bandwidth problem for graphs. *Journal of Zhengzhou University*, 35(1):1–5, March 2003.