



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE CIENCIAS

“UN ENFOQUE AL PROBLEMA DE
ASIGNACIÓN DE CANALES UTILIZANDO
TEORÍA DE GRÁFICAS”

T E S I S

QUE PARA OBTENER EL TÍTULO DE:
LICENCIADA EN CIENCIAS DE LA COMPUTACIÓN
P R E S E N T A :
ECHAVESTE ESCOBAR LAURA IRENE

DIRECTORA DE TESIS:
M. I. MARÍA DE LUZ GASCA SOTO



2004
FACULTAD DE CIENCIAS
SECCION ESCOLAR



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

ACT. MAURICIO AGUILAR GONZÁLEZ
Jefe de la División de Estudios Profesionales de la
Facultad de Ciencias
Presente

Comunicamos a usted que hemos revisado el trabajo escrito:
Un Enfoque al problema de Asignación de Canales utilizando
Teoría de Gráficas.

realizado por **Laura Irene Echeveste Escobar.**

con numero de cuenta **95147305**, quien cubrió los créditos de la carrera de:
Ciencias de la Computación.

Dicho trabajo cuenta con nuestro voto aprobatorio.

Atentamente

Director de Tesis

Propietario **M. en I. María de Luz Gasca Soto.**

Propietario **M. en C. Elisa Viso Gurovich.**

Propietario **Dr. Criel Merino López.**

Suplente **M. en C. José de Jesús Galaviz Casas.**

Suplente **Mat. Laura Pastrana Ramírez.**

Lucy Gasca
Elisa Viso

Merino López Criel

José de Jesús Galaviz Casas
Laura Pastrana R.

Consejo Departamental de Matemáticas.

Francisco Hernández Quiroz

Dr. Francisco Hernández Quiroz
Coordinador de la carrera de Ciencias de la Computación.

MATEMÁTICAS

SI LO ESCUCHO, LO OLVIDO.

SI LO VEO, LO RECUERDO.

SI LO HAGO, LO ENTIENDO.

PROVERBIO CHINO.

A MIS PADRES.

Por su apoyo y también por sus regaños.
Que aún con los problemas propios entre padres
e hija, han estado conmigo.
Elena y Juan, gracias por creer en mí.

A MIS HERMANOS.

Erika, Claudia y Juan Carlos, les agradezco haber
sido mis compañeros de juegos, de travesuras y
de algunos desvelos.

IVETH Y KIKELEIN.

Un Agradecimiento especial por dar un
tinte diferente a mi vida.

LUCY.

Te estoy muy agradecida por toda tu paciencia,
por las horas y horas que nos llevó este trabajo,
por lo que aprendí como tu alumna.

ELISA, LAURA, CRIEL Y JOSE.

Gracias por su tiempo, sus consejos y su valiosa
contribución a este trabajo.

A MIS AMIGOS.

No podía faltar un agradecimiento para ustedes, que son
una parte importante de mis alegrías y experiencias.
Gracias por compartir un espacio de su camino conmigo.

Índice.

Lista de Figuras.	V
Lista de Cuadros.	VII
1. Introducción.	1
Antecedentes Teóricos.	5
2. Teoría de Gráficas.	7
3. Algoritmos y Problemas.	13
4. Coloración de Gráficas.	19
4.1. Problema del número cromático	25
4.2. Cotas del número cromático de una gráfica.	30
4.3. Multicoloración de gráficas.	37
4.4. Cotas generales para el número multicromático.	40
Asignación de Canales.	49
5. Asignación de Canales.	51
5.1. Modelación del Problema	52
5.2. Etiquetación a distancia 2	54
5.3. Reducción a etiquetas con valores enteros	55
5.4. Trayectorias y ciclos	57
5.5. Gráficas k -Coloreables.	61

5.6. Complejidad de la $\mathcal{L}(2,1)$ -etiquetación de G .	62
6. Algoritmos.	67
6.1. Asignación para gráficas k -coloreables.	67
6.2. Asignación para gráficas etiquetadas.	72
6.3. Asignación para gráficas bipartitas.	79
6.4. Asignación para ciclos impares.	83
6.5. Asignación para gráficas 3-coloreables.	88
7. Algoritmos Distribuidos	93
7.1. Gráficas Hexagonales.	94
7.2. Algoritmos en línea distribuidos.	96
7.3. Algoritmos.	97
7.3.1. Algoritmo de Préstamo Aritmético (AB).	97
8. Conclusiones.	101
Apéndices	103
A. Algoritmos.	105
A.1. Análisis de Algoritmos	107
A.1.1. Notación asintótica.	110
A.2. Problemas P, NP y NP-Completos.	112
A.2.1. Clasificación de problemas.	114
A.2.2. Problemas NP.	116
B. Sistema GSM	119
B.1. El subsistema radio.	121
B.2. Reducción del nivel de interferencias.	123
C. Lista de símbolos	125
C.1. Vocabulario técnico bilingüe.	128

Bibliografía **129**

Índice alfabético. **133**

Lista de Figuras.

2.1. Ejemplos de gráficas.	8
2.2. Ejemplos de Gráficas: (a) conexa, (b) no conexa	10
2.3. Gráficas Bipartitas.	11
2.4. Ejemplo de un ciclo C_6	11
2.5. Clanes en G	12
4.1. Una 9-coloración válida.	20
4.2. Una Coloración válida y mínima.	21
4.3. Gráfica G , con grado máximo $\Delta = 5$	22
4.4. Ejemplo de la coloración de una gráfica.	22
4.5. Asignación de colores en una gráfica.	23
4.6. Coloración de G	23
4.7. Coloraciones	24
4.8. Una gráfica para transformar un ejemplar 3SAT en uno de PNC.	28
4.9. Ejemplo de Multicoloraciones con $r \cdot k$ colores.	39
4.10. Ejemplo de una 2-multicoloración del ciclo C_5	39
4.11. Multicoloración de una gráfica G que contiene un clan K_4	41
4.12. Ejemplo de la Multicoloración de la Proposición 7	42
4.13. La multicoloración de una gráfica bipartita.	44
4.14. Ejemplo de la multicoloración de un ciclo C_n con n impar.	45
4.15. Ejemplo de una 4-multicoloración de un ciclo de orden 9.	47
5.1. $\mathcal{L}_1(2, 1)$ -etiquetación.	55
5.2. Etiquetación de Trayectorias.	58

5.3. Etiquetaciones de C_n , con base en la Proposición 16	60
6.1. Un ciclo C_5 , con restricciones $c_0 = 1$, $c_1 = 1$	69
6.2. Gráfica 3-Coloreable	69
6.3. Gráfica $G_1 = (V, A, 4, 3, 3)$	74
6.4. Gráfica $G = (V, A, 5, 3, 2, 1, 1)$	75
6.5. Gráfica $C_8 = (V, A, 3, 2)$	75
6.6. Gráfica $G_2 = (V, A, 3, 2)$	77
6.7. Gráfica bipartita.	80
6.8. Gráfica bipartita 3, $G = (V, A, 4, 2)$	80
6.9. $G_1 = C_5 = (V, A, c_0, c_1)$	84
6.10. Gráfica $G = (V, A, 3, 2)$	90
7.1. Ejemplo de gráfica hexagonal.	95
7.2. Asignación Aritmética en una gráfica hexagonal.	95
7.3. La Asignación Aritmética usada en el Algoritmo AB	98
7.4. Ejemplo de una asignación donde $a = 2$ y $b = 1$	99
7.5. Gráfica $G = (V, A, 3, 2)$ con pesos.	100
7.6. Vértice $(0,0)$ y sus vecinos.	100
A.1. Tiempos de ejecución para tamaños de entrada pequeños.	108
A.2. Tiempos de ejecución para tamaños de entrada mayores.	109

Lista de Cuadros.

6.1. ALGORITMO A.	68
6.2. Etiquetación para $G=(V,A,1,1)$	70
6.3. Etiquetación inicial para $G=(V,A,3,1)$	70
6.4. Asignaciones con el ALGORITMO A.	71
6.5. ALGORITMO A'.	73
6.6. Asignación de la gráfica 6.3 con el ALGORITMO A'.	74
6.7. Asignaciones de $G = (V, A, 5, 3, 2, 1, 1)$	76
6.8. Asignaciones de ALGORITMO A' para el ciclo C_8	77
6.9. Asignación de la gráfica 6.6 con A y A'.	78
6.10. ALGORITMO B.	79
6.11. Etiquetación inicial para $G=(V,A,2,1)$	80
6.12. Asignaciones de la gráfica 6.7 con el ALGORITMO A'.	81
6.13. Asignación inicial para la gráfica.	81
6.14. ALGORITMO B con parámetros no recomendados.	82
6.15. ALGORITMO C.	83
6.16. Etiquetación para $C_5 = (V, A, 2, 1)$	84
6.17. Etiquetación inicial para $C_5 = (V, A, 3, 2)$	85
6.18. Asignaciones de la gráfica 6.9 con diferentes algoritmos.	85
6.19. Etiquetación inicial para $C_9 = (V, A, 4, 2)$	86
6.20. Asignaciones de la gráfica 6.19 con diferentes algoritmos.	86
6.21. Asignaciones de la gráfica 6.19 con el ALGORITMO C.	87
6.22. ALGORITMO E.	89
6.23. Asignacion inicial, usada en el ALGORITMO E.	90

6.24. Asignaciones con el ALGORITMO E.	92
A.1. Funciones de índice de crecimiento.	110
A.2. Órdenes de complejidad.	112

Capítulo 1

Introducción.

En los sistemas de comunicación móvil los canales de comunicación, (*communication links*), se han establecido con un número limitado de canales o frecuencias disponibles. El punto a considerar es: ¿cómo deben ser asignados estos canales a los transmisores de manera que los clientes reciban un servicio de alta calidad?

El objetivo principal de este trabajo es introducir el problema de la asignación de canales, con un enfoque de teoría de gráficas. Este problema radica en asignar canales que puedan reutilizarse en áreas suficientemente separadas, evitando interferencias y sin perder la calidad del servicio, el objetivo es usar el menor número de canales.

Inicialmente el problema de la asignación de canales es modelado como un problema de coloración de gráficas y los algoritmos de coloración se usan para intentar resolverlo. Otro enfoque se obtiene al considerar diversas restricciones, tanto técnicas como legales, donde los canales deben ser asignados de manera que la interferencia sea la menor posible. Hoy en día este problema puede ser considerado como un problema de coloración con restricciones adicionales.

Además, debido al incremento en el uso de servicios de comunicación inalámbrica, así como a la correspondiente escasez y alto costo del espectro de ancho de banda, se ha vuelto de suma importancia, para los operadores de redes celulares, maximizar la eficiencia del espectro.

Tal eficiencia puede ser alcanzada mediante el **reuso de canales**, es decir, el uso simultáneo de una misma parte del espectro de radio por canales de comunicación en diferentes puntos de la red. El reuso es restringido por niveles de ruido, resultado de la interferencia entre dichos canales.

Una antena cubre un área de terreno en la cual puede atender llamadas; a esta área se le conoce como celda. Este problema podemos modelarlo con una gráfica $G = (V, A)$, donde los nodos en la gráfica corresponden a cada celda y las aristas representan la interferencia entre celdas vecinas. Asumimos que existe una demanda fija para cada celda y que una asignación de canales para las celdas debe ser encontrada.

Este trabajo está organizado como sigue: El Capítulo 2 presenta material de introducción a la *Teoría de Gráficas*, algunas definiciones básicas, ejemplos y notación usada a lo largo de este trabajo. El Capítulo 3 está enfocado a dar una breve introducción sobre *Algoritmos*, su complejidad y la clasificación de problemas.

El Capítulo 4 presenta el problema del *Número Cromático*, la coloración de los vértices de una gráfica, con un número mínimo de colores. Mostramos que éste es un problema NP-completo. Describimos algunos resultados teóricos con el fin de establecer cotas para el número cromático. Al final del capítulo tratamos el problema de la multicoloración de gráficas, donde se busca colorear los vértices de una gráfica con más de un color. Se define el número cromático y multicromático para algunas familias de gráficas.

El problema de la *Asignación de canales* y la etiquetación a distancia 2 se presenta en el Capítulo 5. En este capítulo se describe cómo se modela la asignación de canales mediante la teoría de gráficas y cómo se considera el número cromático para hacer una asignación eficiente. En los Capítulos 6 y 7 se describen algunos algoritmos para la *Asignación de canales*, tanto generales como para gráficas con restricciones particulares.

Por último, en la parte de Apéndices, se tratan algunos temas, que si bien no se mencionan a fondo, son fundamentales para este trabajo. El Apéndice A es una introducción a los Algoritmos y su análisis, se trata los conceptos de complejidad y notación asintótica. Incluimos también una breve reseña del sistema GSM, así como una lista de símbolos y el vocabulario técnico bilingüe.

Antecedentes Teóricos.

Capítulo 2

Teoría de Gráficas.

Diversas aplicaciones pueden modelarse mediante Teoría de gráficas, por ejemplo, calcular la ruta más rápida para trasladarse de un lugar a otro, o visitar los sitios interesantes en una ciudad, sin ir dos veces al mismo lugar. En el presente capítulo damos definiciones básicas sobre la Teoría de gráficas, así como de los términos y notación usados en capítulos posteriores.

Una **gráfica** $G = (V, A)$ consiste de un conjunto finito V , no vacío, de objetos llamados puntos o vértices, junto con una colección A , de pares, no necesariamente ordenados, de vértices distintos, llamados aristas. Por ejemplo, la Figura 2.1(a) representa la gráfica G cuyos conjuntos son:

- $V(G) = \{a, b, c, d, e\}$, y aristas
- $A(G) = \{(a, b), (a, c), (a, d), (a, e), (b, c), (b, d), (b, e), (c, d), (c, e), (d, e)\}$.

A su vez, la gráfica G_1 en la Figura 2.1(b) es representada por los conjuntos:

- $V(G_1) = \{1, 2, 3, 4, 5, 6, 7\}$,
- $A(G_1) = \{(1, 4), (3, 2), (5, 6), (5, 7)\}$.

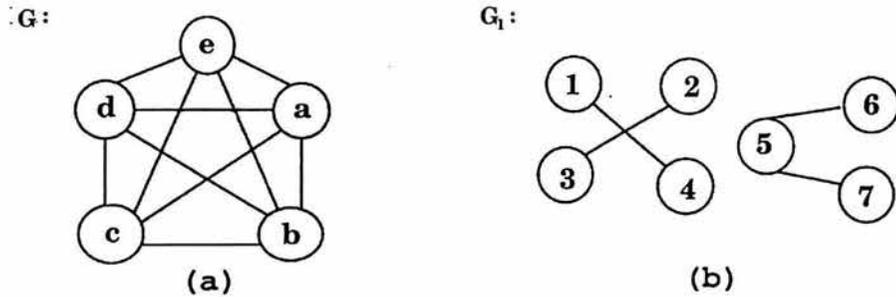


Figura 2.1: Ejemplos de gráficas.

Se dice que la cardinalidad del conjunto $V(G)$ representa el **orden** de G y la del conjunto $A(G)$ es el **tamaño** de G . Por ejemplo, en la Figura 2.1(a) el orden de G es $|V(G)|=5$ y su tamaño es $|A(G)|=10$.

A continuación daremos algunas definiciones que ocuparemos frecuentemente durante el presente trabajo.

Definición 1 *Subgráfica.* Una subgráfica H de una gráfica $G = (V, A)$ consiste de un subconjunto de los vértices de G y un subconjunto de aristas de G , los cuales forman una gráfica, es decir, $H = (V_1, A_1)$ es una gráfica tal que $V_1 \subseteq V$ y $A_1 \subseteq A$.

Definición 2 *Vértices adyacentes.* Si $a = (u, v)$ es una arista de G , diremos que u y v son adyacentes y que a incide en u y a incide en v .

Definición 3 *Aristas adyacentes.* Dos aristas son adyacentes cuando inciden en un mismo punto.

Definición 4 *Grado.* El grado $\delta(u)$ de un vértice $u \in V(G)$ es el número de aristas que inciden en u . Por ejemplo, en la Figura 2.1(b), $\delta(5) = 2$ y $\delta(2) = 1$.

Definición 5 *Grado mínimo.* El grado mínimo de una gráfica $G = (V, A)$, es el número mínimo de aristas que inciden en un vértice v , entre todos los vértices en $V(G)$. Formalmente se define como: $\delta(G) = \min\{\delta(v) \mid v \in V(G)\}$.

Para las gráficas de la Figura 2.1, tenemos que $\delta(G) = 4$; y $\delta(G_1) = 1$;

Definición 6 *Grado máximo.* El grado máximo de una gráfica $G = (V, A)$ se define como: $\Delta(G) = \max\{\delta(u) \mid u \in V(G)\}$

Para las gráficas de la Figura 2.1, se tiene que: $\Delta(G_1) = 2$; y $\Delta(G) = 4$.

Definición 7 *Camino.* Un camino en una gráfica $G = (V, A)$ es una sucesión de vértices y aristas $C = (v_0, a_0, v_1, a_1, \dots, a_{n-1}, v_n)$ donde $a_i = (v_i, v_{i+1})$, $0 \leq i \leq n-1$.

Definición 8 *uv-Camino.* Es un camino del vértice u al vértice v .

Definición 9 *Trayectoria.* Una trayectoria en una gráfica $G = (V, A)$ es un camino que no repite vértices. Si existe una trayectoria T entre los vértices u y v diremos que T es una *uv-trayectoria*.

Definición 10 *Gráfica conexa.* Una gráfica $G = (V, A)$ es conexa si y sólo si para cualquier par de vértices $u, v \in V(G)$ existe un *uv-camino*.

Definición 11 *Longitud de un camino.* Sea C un camino de u a v , $l(C)$, la longitud de C es igual al número de aristas que posee tal camino.

Definición 12 *Distancia.* Sean $G = (V, A)$ una gráfica conexa y $u, v \in V(G)$; la *mínima distancia* en $G = (V, A)$ de u a v la definimos como la longitud de la trayectoria mínima entre u y v : $d_G(u, v) = \min\{l(T) \mid T \text{ es } uv\text{-trayectoria}\}$.

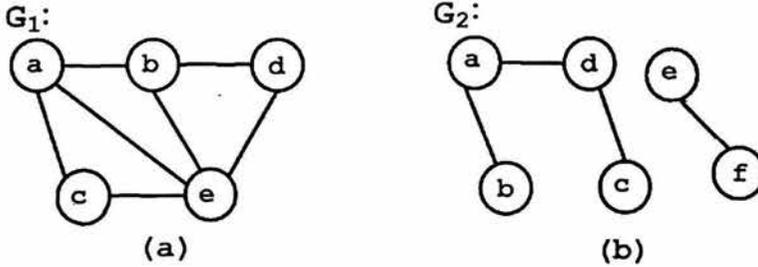


Figura 2.2: Ejemplos de Gráficas: (a) conexas, (b) no conexas

Definición 13 Excentricidad. Sea $G = (V, A)$ una gráfica; la excentricidad de un vértice $v \in V(G)$ se define como: $exc_G(v) = \max\{d_G(v, w) \mid w \in V(G)\}$.

Definición 14 Diámetro. El diámetro de una gráfica $G = (V, A)$ se define como: $diam G = \max\{exc_G(v) \mid v \in V(G)\}$.

Definición 15 Gráfica completa. Una gráfica $G = (V, A)$ de orden n es una *gráfica completa*, si $\forall u, v \in V(G)$, $u \neq v$, existe la arista (u, v) . Tal gráfica se denota con K_n . Un ejemplo de estas gráficas es G en la Figura 2.1(a).

Definición 16 Complemento de una gráfica. El complemento G^c de una gráfica $G = (V, A)$ de orden n , es una gráfica tal que: $V(G^c) = V(G)$ y $A(G^c) = \{a_i \in K_n \mid a_i \notin G\}$.

Definición 17 Gráfica bipartita. Una gráfica $G = (V, A)$, es *bipartita* si existe una partición de $V(G)$ en dos conjuntos V_1 y V_2 , donde $|V_1| = n$, $|V_2| = m$ y $|V(G)| = n+m$, tal que toda arista de G une un vértice de V_1 con un vértice de V_2 .

Es decir, no hay aristas con ambos extremos en V_1 , ni aristas con ambos extremos en V_2 . Adicionalmente, si todo vértice $u_i \in V_1$ es adyacente a todo vértice $v_j \in V_2$, con $1 \leq i \leq n$ y $1 \leq j \leq m$, entonces es una **gráfica bipartita completa**, la cual se denota como $K_{n,m}$.

Por ejemplo, considérese las gráficas de la Figura 2.3: G es una gráfica bipartita y $K_{6,4}$ es una gráfica bipartita completa.

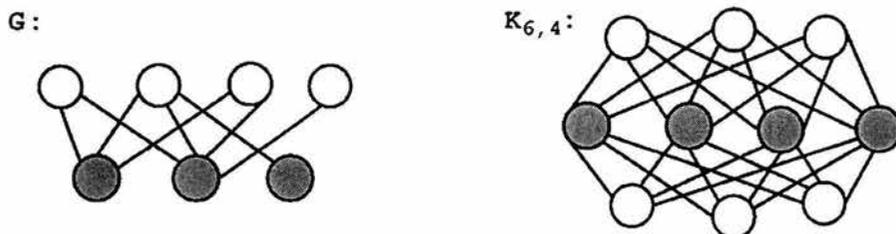


Figura 2.3: Gráficas Bipartitas.

Definición 18 *Ciclo*. Un ciclo de orden n , C_n , $n \geq 3$, es una sucesión de vértices y aristas:

$$C_n = (v_1, a_1, v_2, \dots, a_{n-1}, v_n, a_n, v_1) \text{ tales que:}$$

$$a_i = (v_i, v_{i+1}), \quad 1 \leq i \leq n-1; \quad a_n = (v_n, v_1);$$

$$\text{y } v_i \neq v_j \quad \forall i \neq j \quad 1 \leq i \leq n, 1 \leq j \leq n$$

Dicho de otra manera, C_n es simplemente un camino cerrado de longitud n , que sólo repite el primer vértice. La Figura 2.4 ilustra un ciclo C_6 .

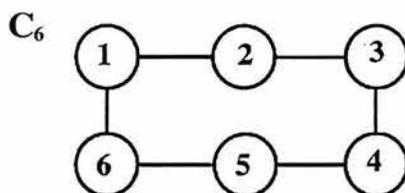


Figura 2.4: Ejemplo de un ciclo C_6 .

Definición 19 *Clan.* Un clan K de una gráfica $G = (V, A)$, es una subgráfica de G , tal que K por sí misma es una gráfica completa.

En la gráfica G , de la Figura 2.5 identificamos los clanes K_2, K_3, K_4 .

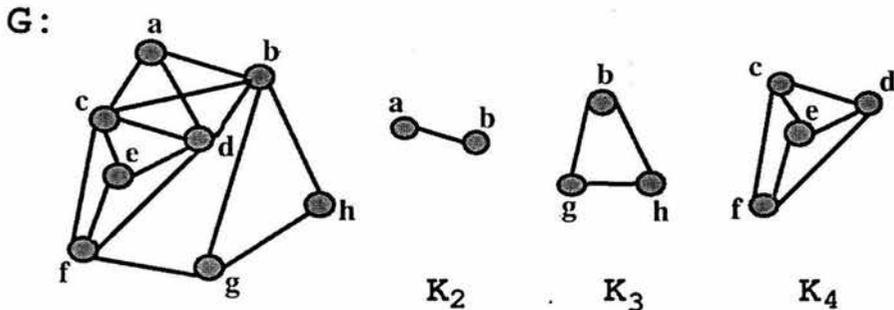


Figura 2.5: Clanes en G .

Definición 20 *Densidad.* Dada una gráfica $G = (V, A)$ de orden n , el número máximo de aristas que puede contener es igual a $n(n-1)/2$. Se define la densidad $\rho(G)$ de G como el cociente entre el número de aristas de G y el máximo número de aristas que puede contener, esto es:

$$\rho(G) = \frac{|A(G)|}{n(n-1)/2} = \frac{2|A(G)|}{n(n-1)}.$$

Capítulo 3

Algoritmos y Problemas.

En este capítulo daremos un panorama general sobre algoritmos, su complejidad y la clasificación de los problemas. En el Apéndice A se presentan con detalle conceptos fundamentales del Análisis de Algoritmos.

En general, un problema es una cuestión a resolver y para solucionarlo es necesario definirlo de manera precisa indicando los parámetros (datos) requeridos por la especificación del problema. A la asignación particular de valores para los parámetros se le denomina *ejemplar* del problema, o *datos de entrada* del problema o simplemente *entrada* del problema.

Un algoritmo es un conjunto bien definido de instrucciones para obtener la solución de un problema, sobre un ejemplar dado, en un número finito de pasos. Un algoritmo tiene las siguientes características:

- I Trabaja a partir de datos, aunque ocasionalmente podría no recibir entradas.
- II Produce como salida un resultado que corresponde a la solución del problema; o bien termina indicando que no existe tal.
- III Secuencia finita de pasos. El algoritmo define una secuencia de pasos cuya ejecución transforma a la entrada en la salida. Cada uno de estos pasos debe estar bien definido.

IV Correctez. El algoritmo debe terminar y garantizar la solución para cualquier ejemplar del problema.

Dado un problema Y podrían existir varios algoritmos que lo resuelvan; para determinar cuál es el mejor de ellos, requerimos establecer reglas de medición sobre el comportamiento de los mismos. La complejidad de un algoritmo mide la cantidad de “esfuerzo computacional” usado por una computadora para resolver un problema usando tal algoritmo. Generalmente, esta medida se refiere al número de operaciones elementales de cómputo, tiempo de ejecución o espacio de almacenamiento.

En este trabajo usaremos el número de operaciones elementales como medida de complejidad. La complejidad de un algoritmo es una función sobre el tamaño y representación de los datos de entrada, es decir, del ejemplar.

Se dice que un algoritmo es eficiente si su función de complejidad resulta ser un polinomio sobre el tamaño de la entrada, digamos n .

Un problema es llamado **tratable** si existe un algoritmo eficiente que lo resuelva. Un problema se denomina **intratable** si es posible establecer que no existe un algoritmo eficiente que lo resuelva. Hay varios problemas para los cuales no se ha podido determinar si son tratables o no.

A la clase de todos los problemas para los cuales existe un algoritmo polinomial que los resuelva se le denomina **Clase P**. Como se indicó anteriormente, existen muchos problemas que no se sabe si son tratables o intratables; entre estos problemas está la clase de los problemas **NP-completos**, pero antes de definirlos con más precisión, estableceremos una serie de conceptos importantes.

Si un problema Π tiene la propiedad de que la respuesta para cada ejemplar del problema es únicamente sí o no, entonces Π es llamado un **problema de decisión**.

Cualquier problema puede ser transformado en un problema de decisión.

Un algoritmo no-determinístico para un problema de decisión Π tiene, básicamente, dos estados: Para un ejemplar dado E de Π , el primer estado adivina o propone una estructura S -solución potencial- y el segundo estado usa el ejemplar E y la solución potencial S para verificar en tiempo polinomial si E induce a una respuesta afirmativa, sí, o negativa, no. A esta última etapa, se le denomina *Estado de Verificación*.

En particular, si Π es un problema de decisión, entonces para todos los ejemplares de Π se debe satisfacer que:

- I Si E induce a la respuesta afirmativa, debe existir una solución potencial tal que cuando E es propuesto, el estado de verificación responda sí;
- II Si E induce a la respuesta negativa, entonces el estado de verificación debe responder no, al evaluar a E .

Si las propiedades I y II se satisfacen, se dice que el algoritmo no-determinístico **resuelve** Π . A la clase de todos los problemas de decisión que pueden ser resueltos por un algoritmo no-determinístico en tiempo polinomial se le llama **clase NP**.

Se dice que un problema Π_1 puede ser **transformado polinomialmente** en un problema Π_2 si existe un algoritmo eficiente A , llamado transformación polinomial, que convierte cada ejemplar E de Π_1 en un ejemplar $A(E)$ de Π_2 tal que: E induce a una respuesta sí para Π_1 si y sólo si $A(E)$ induce una respuesta sí para Π_2 . A tal transformación la denotamos como $\Pi_1 \propto \Pi_2$ y decimos que Π_1 puede ser reducido polinomialmente a Π_2 .

La clase de los problemas **NP-completos** consiste de todos los problemas Π que son NP con la propiedad de que cualquier otro problema X en NP, X puede ser polinomialmente transformado a Π .

En 1971, Cook establece la existencia del primer problema NP-completo[12]: El **problema SAT**. A partir de ahí se han encontrado varios problemas NP-completos, por ejemplo:

■ Problema de la Mochila.

Dado un conjunto S de objetos, tal que para $\forall x \in S$, se tiene asociado un tamaño $t(x)$ y un valor $v(x)$; t es el tamaño máximo que puede llevar la mochila, y $t(Y)$ y $v(Y)$ representan la suma de los tamaños y valores de los elementos del conjunto Y , tal que $Y \subseteq S$.

- Problema de Optimización: Encontrar un conjunto de elementos $Y = \{x_i \mid x_i \in S\}$ de manera que $v(Y)$ sea máximo y cumpla $t(Y) \leq t$.
- Problema de Decisión: Determinar si $\exists Y \subseteq S$ tal que el tamaño total $t(Y)$ sea menor que t y cuyo valor $v(Y)$ sea tal que $v(Y) \leq t$.

■ Bin Packing

Dados un conjunto S de n objetos y para cada objeto $x \in S$ un tamaño $t(x)$ y un número positivo b , buscamos repartir todos los objetos en k contenedores, de manera que la suma del tamaño de los objetos no exceda b .

- Problema de Optimización: Distribuir todos los objetos, de manera que se use el mínimo número de contenedores.
- Problema de Decisión: Dados b y k determinar si el conjunto puede ser dividido en k subconjuntos, tal que la suma de los tamaños sea menor o igual que b .

- Un Clan en una gráfica $G = (V, A)$.
 - Problema de Optimización: Encontrar el clan de mayor orden en una gráfica $G = (V, A)$.
 - Problema de Decisión: Dada una gráfica $G = (V, A)$ y un entero positivo k , determinar si G tiene un clan de tamaño mayor o igual que k .

- El Agente Viajero.

El Problema del Agente Viajero consiste en que dado un conjunto de ciudades, un vendedor debe visitar cada una de ellas y regresar a la ciudad de partida.

 - Problema de Optimización: Obtener un recorrido por todas las ciudades, de manera que el costo sea el mínimo.
 - Problema de Decisión: Dado un número t , hay que determinar si existe un recorrido para el Agente viajero tal que el costo sea menor o igual a t .

- Coloración Mínima.

Dada una gráfica $G = (V, A)$, queremos asignar colores a sus vértices, de manera que vértices vecinos tengan colores diferentes.

 - Problema de Optimización: Asignar un conjunto de colores a los vértices de G , usando el menor número de colores posible.
 - Problema de Decisión: Dados una gráfica $G = (V, A)$ y un entero t , hay que determinar si existe una t -coloración de los vértices de G de manera que vértices adyacentes tengan colores distintos.

Capítulo 4

Coloración de Gráficas.

Aquí presentamos un problema de coloración de gráficas: encontrar el mínimo número de colores para colorear los vértices de una gráfica, de manera que vértices adyacentes tengan colores distintos.

Exponemos algunos resultados teóricos, que si bien no son restrictivos nos permiten establecer cotas. Posteriormente, se desarrolla un modelo para colorear una gráfica con más de un color por vértice, para tratar con el requerimiento de varios canales por vértice o varias frecuencias por área.

Definición 21 *n-Coloración.* Sean $G = (V, A)$ una gráfica y $\mathcal{C} = \{c_1, c_2, \dots, c_n\}$ un conjunto de colores. Una **coloración** de los vértices de G , es una función $f : V(G) \rightarrow \mathcal{C}$.

Si n colores son utilizados, entonces nos referiremos a la coloración como una **n- coloración**.

Definición 22 *Coloración válida.* Una coloración para los vértices de G es *válida* si cualesquiera dos vértices adyacentes tienen asignado un color distinto.

La coloración válida de una gráfica induce una partición en el conjunto de vértices; a cada conjunto de la partición le llamamos **clase de color**. Por ejemplo, la gráfica en la Figura 4.1 representa una 9-coloración válida, donde el número en cada uno de los vértices representa la clase de color a la que pertenece el vértice; representaremos cada color con un número entero.

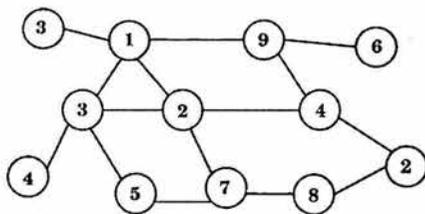


Figura 4.1: Una 9-coloración válida.

Diremos que una gráfica es n -*coloreable* si admite una n -coloración válida.

Decimos que un conjunto de vértices $S \subseteq V(G)$ es *independiente* si no hay aristas en G , con ambos extremos en S . Cada clase de color de una coloración válida es un conjunto independiente.

Definición 23 Número cromático. El mínimo número de colores asignados a los vértices de una gráfica G para obtener una coloración válida se le llama número cromático de G , y se denota por $\chi(G)$.

Observemos que la coloración de la gráfica en la Figura 4.1 usa nueve colores; sin embargo, como se observa en la Figura 4.2, son suficientes tres colores para obtener una coloración válida. Observemos que la gráfica tiene un triángulo el cual no puede ser coloreado con menos de tres colores. Por lo tanto, el número cromático resulta ser $\chi(G) = 3$.

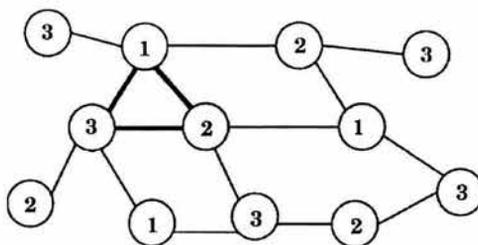


Figura 4.2: Una Coloración válida y mínima.

Resultado 1 *Es fácil observar que:*

- La gráfica completa de n vértices se colorea con n colores: $\chi(K_n) = n$.
- Para toda gráfica bipartita: $\chi(G) = 2$
- Para todo ciclo de longitud par se requieren dos colores: $\chi(C_n) = 2$;
Para los ciclos de longitud impar se necesitan tres colores: $\chi(C_n) = 3$.

Dada una gráfica $G = (V, A)$ su número cromático siempre es menor o igual al número de vértices. Además, una gráfica que contiene un clan de orden r , es decir, una subgráfica K_r , tiene un número cromático mayor o igual a r . Estas cotas pueden ajustarse en función del grado de los vértices de G .

Teorema 1 *Toda gráfica $G = (V, A)$ con grado máximo Δ es a lo más $(\Delta + 1)$ -coloreable, es decir $\chi(G) \leq 1 + \Delta(G)$.*

Revisemos un par de ejemplos, antes de dar la demostración del teorema. La gráfica $G = (V, A)$ en la Figura 4.3, tiene como grado $\Delta = \Delta(G) = 5$ y orden $n = 6$. Los vértices a y b tienen grado $\delta(a) = \delta(b) = 5$, es decir, son los de mayor grado en la gráfica.

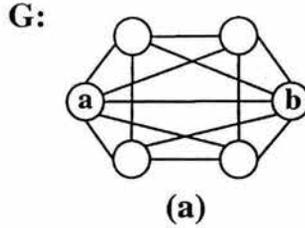


Figura 4.3: Gráfica G , con grado máximo $\Delta = 5$

El Teorema 1 señala que serán suficientes 6 colores para colorear G . En este caso el orden también es 6, entonces basta con asignar un color a cada vértice de los $(\Delta + 1)$ colores disponibles, tal como se ve en la Figura 4.4.

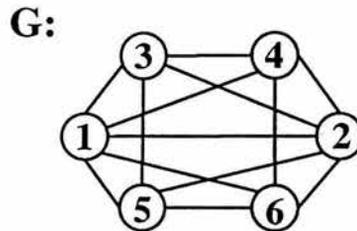


Figura 4.4: Ejemplo de la coloración de una gráfica.

Como segundo ejemplo, consideremos la gráfica $G = (V, A)$ de la Figura 4.5(a), el grado de G es $\Delta(G) = \Delta = 5$ y v es el vértice de mayor grado en la gráfica. De acuerdo al Teorema 1, tenemos 6 colores disponibles para colorear G .

En primer término, podemos asignar el color 1 al vértice v y ya que cualquier $u \in V(G)$ tiene grado $\delta(u) \leq 5$, podemos distribuir los otros 5 colores disponibles entre todos los vértices adyacentes a v , tal como se muestra en la Figura 4.5(b).

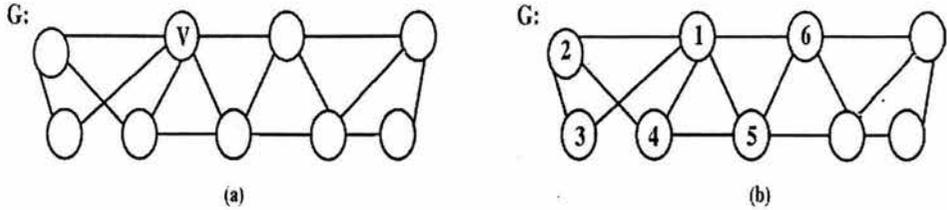


Figura 4.5: Asignación de colores en una gráfica.

Finalmente nos fijamos en los vértices que aún no han sido coloreados y le asignamos un color que no tenga alguno de sus vecinos. En la Figura 4.6 podemos ver la gráfica G , completamente coloreada con seis colores, o sea $\Delta(G) + 1$.

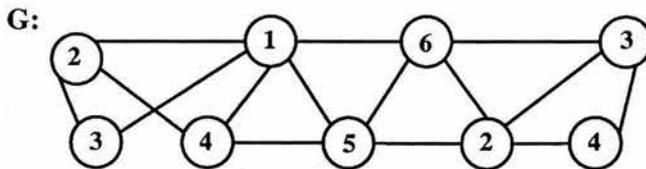


Figura 4.6: Coloración de G .

Demostración del Teorema 1:

Por inducción sobre el número de vértices de G .

Para $n = \Delta + 1$. Este es el caso más sencillo, ya que a cada vértice le asignamos un color diferente de los $\Delta + 1$ disponibles.

Hipótesis de inducción. Toda gráfica G de orden $n - 1$ y grado máximo Δ es $(\Delta + 1)$ -coloreable.

Sean G una gráfica de orden n y $v \in V(G)$ y sea $G_1 = G - v$. G_1 es de orden $n - 1$ y a lo más de grado máximo Δ . Por Hipótesis de Inducción G_1 tiene una $(\Delta + 1)$ -coloración. Por otra parte v es adyacente como máximo a Δ vértices, por lo tanto podemos asignarle un color, que no tengan sus vecinos, de los $\Delta + 1$ colores disponibles, de manera que obtenemos para G una $(\Delta + 1)$ -coloración válida.

■

Esta cota puede mejorarse, cuando G es conexa, no completa y de grado máximo $\Delta(G) \geq 3$. En este caso podemos decir que G es Δ -colorable. Por ejemplo, consideremos las gráficas en la Figura 4.7, donde el número en cada vértice representa el color asignado. Observemos que G_1 y G_2 son gráficas que contienen un clan K_4 , por lo que $\chi(G) \geq 4$, además $\Delta(G_1) = 5$ y $\Delta(G_2) = 4$.

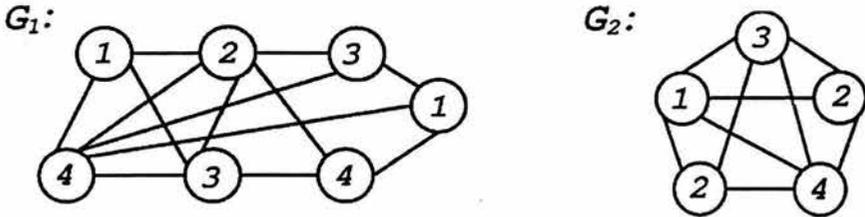


Figura 4.7: Coloraciones

Ya que hemos dado una 4-coloración, tenemos que su número cromático es $\chi(G_1) = 4$ y $\chi(G_2) = 4$, con lo que existe una mejora a la cota del Teorema 1.

Deducimos entonces que estas cotas no son restrictivas en general; por ejemplo el grado máximo de la gráfica bipartita completa $K_{n,m}$ es $\Delta(G) = \max\{n, m\}$, que puede ser arbitrariamente grande, sin embargo su número cromático siempre es 2.

Analicemos un poco más el problema de determinar el número cromático $\chi(G)$ para una gráfica dada.

4.1. Problema del número cromático

En esta sección se describe el problema del número cromático para una gráfica cualquiera y se demuestra que es un problema NP-completo. Este problema es fundamental para el desarrollo de este trabajo.

Problema: Dada una gráfica $G = (V, A)$ determinar el mínimo número de colores necesarios y suficientes para alcanzar una coloración válida de la gráfica G . Es decir se pretende determinar $\chi(G)$.

Transformamos este problema de optimización al siguiente problema de decisión:

Problema PNC: Dada una gráfica $G = (V, A)$ y un entero positivo b , ¿Existe una coloración válida para G con a lo más b colores?

De esta manera, el planteamiento formal del problema queda como:

Ejemplar: Sean $G = (V, A)$ una gráfica de orden n y un entero b , tal que $2 \leq b \leq n$.

Pregunta: ¿Es $\chi(G) \leq b$?

Mostraremos que el problema PNC es NP-completo mediante una transformación del problema del problema 3SAT a PNC. El problema 3SAT consiste en determinar si una expresión lógica en su forma normal conjuntiva se satisface, teniendo como restricción que cada cláusula tiene tres variables.

Teorema 2 *El problema PNC es NP-completo.*

Primero demostramos que PNC está en NP, exhibiendo el siguiente algoritmo no determinístico polinomial:

ALGORITMO NO DETERMINÍSTICO PARA EL PROBLEMA
DEL NÚMERO CROMÁTICO DE UN GRÁFICA.

Ejemplar: Sea G una gráfica, $G = (V, A)$ y b un entero positivo, $2 \leq b \leq n \leq m$.

1. Sea $C = \{1, 2, \dots, m\}$ una gama de colores. Asignamos aleatoriamente a cada vértice de $V(G)$ un color de C .
2. Verificamos si la coloración dada a los vértices en V es válida y usa a lo más b colores.

Es claro que la verificación puede realizarse en tiempo polinomial, por lo cual el algoritmo descrito es no determinístico polinomial para el problema PNC. Por lo tanto, PNC está en NP. ■

A continuación describimos detalladamente la transformación $3SAT \propto PNC$.

Realizamos la transformación de ejemplares de la siguiente forma: Sea E_s un ejemplar del 3SAT con k cláusulas C_1, C_2, \dots, C_k , cada una de las cuales contiene a lo más 3 literales y en total se tienen n variables x_1, x_2, \dots, x_n . Construiremos en tiempo polinomial una gráfica G que tiene una $(n + 1)$ -coloración, si y sólo si las cláusulas se satisfacen.

Si $n \leq 4$ será necesario verificar a lo más 16 asignaciones posibles de valores de verdad para las variables, para determinar si las cláusulas se satisfacen. Esto es factible hacerlo, incluso manualmente. Asumimos entonces que $n > 4$.

Construimos la gráfica G con orden $3n + k$. Primero definimos los vértices como:

$$V(G) = \{x_1, x_2, \dots, x_n\} \cup \{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n\} \cup \{y_1, y_2, \dots, y_n\} \cup \{c_1, c_2, \dots, c_k\},$$

donde \bar{x}_i es la negación de x_i y cada c_i corresponde a la cláusula C_i .

Definimos el conjunto de aristas de la manera siguiente:

- Para cada i , $1 \leq i \leq n$, (x_i, \bar{x}_i) es una arista de G .
- Cada vértice y_i , $1 \leq i \leq n$, está unido a los vértices y_j , x_j y \bar{x}_j , para $j \neq i$ y $1 \leq j \leq n$.
- El vértice x_i , $1 \leq i \leq n$, está unido al vértice c_j , $1 \leq j \leq k$, siempre que x_i no sea una de las literales en la cláusula C_j .
- De manera correspondiente, unimos cada vértice \bar{x}_i , $1 \leq i \leq n$, a cada c_j , $i \leq j \leq k$, si \bar{x}_i no aparece en C_j .

Ilustramos, en la Figura 4.8, esta construcción para la gráfica G de orden 15, con $n = 4$, $k = 3$, $C_1 = \{x_1, x_3\}$, $C_2 = \{x_1, \bar{x}_2, \bar{x}_3\}$, $C_3 = \{x_2, x_4\}$.

Ahora demostraremos que la gráfica G tiene una $(n + 1)$ -coloración válida si y sólo si las cláusulas se satisfacen. Ya que la subgráfica inducida por y_1, y_2, \dots, y_n es completa, al menos se necesitan n colores para colorear los vértices de G . Sin pérdida de generalidad, asumimos que y_i tiene asignado el color i para cada $1 \leq i \leq n$.

El vértice y_i es adyacente a cada par de vértices x_j , \bar{x}_j con $j \neq i$, de esta manera si el color i se usa para colorear algún vértice en el conjunto $S = \{x_1, x_2, \dots, x_n\} \cup \{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n\}$ entonces puede usar el color de x_i o el de \bar{x}_i , pero no ambos, porque son adyacentes.

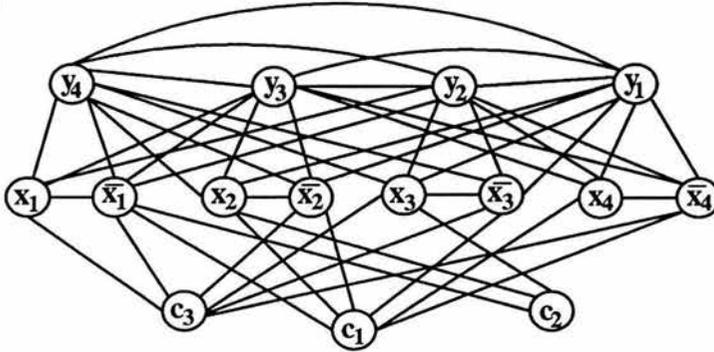
G:

Figura 4.8: Una gráfica para transformar un ejemplar 3SAT en uno de PNC.

Por tanto, necesitamos otro color para colorear todos los vértices de S , es decir requerimos del color $n + 1$. De aquí que $\chi(G) \geq n + 1$.

Ahora bien, si tenemos la gráfica coloreada con $n + 1$ colores, entonces para la i -ésima variable, $1 \leq i \leq n$, usamos el color i o el color $(n + 1)$: si x_i se colorea con el color i , a su complemento se le da el color $(n + 1)$ y viceversa.

Supongamos que $\chi(G) = n + 1$. Mostraremos que en este caso las cláusulas se satisfacen. Consideremos una $(n + 1)$ -coloración de G . Podemos asumir que y_i tiene el color i , donde $1 \leq i \leq n$. Ya que cada C_i contiene a lo más 3 literales y $n > 4$, se sigue que el vértice c_j es adyacente con x_l y \bar{x}_l para alguna l , $1 \leq l \leq n$.

Dado que x_l o \bar{x}_l deben recibir el color $n + 1$, tenemos que c_j no puede ser coloreado con $n + 1$ para cada j , $1 \leq j \leq n$. Entonces, cada c_j debe ser coloreado con uno de los colores $\{1, 2, 3, \dots, n\}$.

Asignemos el valor de verdad F al vértice x_i o \bar{x}_i que es coloreado con el color $n + 1$; mientras al otro vértice del par le asignamos el valor de verdad V . Entonces cada c_j debe ser coloreado con el mismo color dado a alguna literal que esté en C_j y que tenga un valor de verdad V . Con esta asignación de valores de verdad a x_1, x_2, \dots, x_n , las cláusulas se satisfacen.

■

Supongamos ahora que las cláusulas se satisfacen. Para cada c_i seleccionamos una literal $z_{i,j}$ de C_i , donde $z_{i,j}$ es x_{i_j} o \bar{x}_{i_j} , tal que el valor de verdad para $z_{i,j}$ es V . Coloreamos c_i y $z_{i,j}$ con el color i_j , y coloreamos a $\bar{z}_{i,j}$ con el color $n + 1$. Si cada par x_i, \bar{x}_i permanece sin colorear, entonces se pone el color i a x_i y el color $n + 1$ a \bar{x}_i .

Finalmente, hemos construido una $(n + 1)$ -coloración válida de G , esto es $\chi(G) = n + 1$.

■

Ya que G tiene a los más $\binom{n}{2} + 2n^2 - n + k(2n - 1)$ aristas, la transformación se realiza en tiempo polinomial.

■

Ahora podemos asegurar que determinar el número cromático de una gráfica es un problema NP-completo; ya que no siempre resultará fácil calcularlo, veamos como establecer cotas para aproximarnos al número óptimo de colores usados.

4.2. Cotas del número cromático de una gráfica.

Sea $G = (V, A)$ una gráfica bipartita de orden n . Sabemos que existe una partición de $V(G)$ en dos subconjuntos V_1 y V_2 , donde $V_1 \neq \emptyset$ y $V_2 \neq \emptyset$. Sean n_1 y n_2 los órdenes de V_1 y V_2 respectivamente, por definición $n = n_1 + n_2$. De esta forma, una gráfica bipartita completa tiene un número de aristas igual a $n_1(n - n_1)$ donde $1 \leq n_1 \leq (n - 1)$. El valor de n_1 que maximiza el número de aristas resultante es $n_1 = n/2$ si n es par y $n_1 = (n - 1)/2$ si n es impar. En este caso, si definimos $M_{max}(n, 2)$ como el número máximo de aristas que puede tener una gráfica bipartita de orden n , tenemos que:

$$M_{max}(n, 2) = \begin{cases} \left(\frac{n}{2}\right)^2 & \text{si } n \text{ es par} \\ \left(\frac{n-1}{2}\right)\left(\frac{n+1}{2}\right) & \text{si } n \text{ es impar} \end{cases} \quad (4.1)$$

De esta manera, deducimos que toda gráfica de orden n con un número de aristas superior a $(n/2)^2$ no es bipartita. Esta condición puede traducirse en términos de la densidad de la gráfica de modo que para toda gráfica G de orden n y densidad ρ se tiene que G no es bipartita si:

$$\begin{aligned} \rho \cdot \frac{n(n-1)}{2} &> \left(\frac{n}{2}\right)^2 \\ \Leftrightarrow \rho \cdot \frac{n(n-1)}{2} &> \frac{n \cdot n}{2 \cdot 2} \\ \Leftrightarrow \rho \cdot (n-1) &> \frac{n}{2} \\ \Leftrightarrow \rho &> \frac{n}{2} \cdot \frac{1}{n-1} \end{aligned}$$

Es decir G no es bipartita si su densidad ρ es tal que:

$$\rho > \frac{n}{2(n-1)}.$$

Así, en el caso en que el orden de la gráfica sea arbitrariamente grande, cualquier gráfica con una densidad superior a $\frac{1}{2}$ no es bipartita.

Generalizando la expresión anterior, definimos $M_{max}(n, k)$ como el número máximo de aristas que puede tener una gráfica k -coloreable o k -partita de orden n . Considérese una gráfica $G = (V, A)$, k -coloreable de orden n . En este caso es posible partir $V(G)$ en k subconjuntos disjuntos de vértices V_i de orden n_i , de tal manera que:

$$\sum_{i=0}^{k-1} n_i = n \quad \text{y} \quad n_i \geq 1, \quad \forall i.$$

A continuación calculamos el número de aristas de una gráfica k -partita completa en el que cada uno de los subconjuntos disjuntos V_i tiene n_i vértices. Para ello, en vez de sumar restaremos aristas.

Una gráfica k -partita y completa de orden n contiene las $n(n-1)/2$ aristas de K_n menos todas las aristas correspondientes a vértices de un mismo subconjunto V_i . De esta forma, el número total de aristas de una gráfica completa k -partita de orden n es:

$$A(n, k) = \frac{n(n-1)}{2} - \sum_{i=0}^{k-1} \frac{n_i(n_i-1)}{2}$$

Ahora buscamos los valores n_i que maximizan esta función. Dado que el primer término de la ecuación sólo depende de n , se minimizará el segundo término. Obtenemos el siguiente planteamiento:

$$\begin{aligned}
& \text{minimizar } \sum_{i=0}^{k-1} n_i(n_i - 1)/2 \\
& \text{sujeto a} \\
& \sum_{i=0}^{k-1} n_i = n \\
& n_i \in \mathbb{Z} \\
& n_i \geq 1 \quad \forall i.
\end{aligned} \tag{4.2}$$

La partición V_i de vértices que optimiza las expresiones anteriores debe ser tal que todas las particiones V_i tengan el mismo orden o a lo sumo, órdenes separados por una unidad. De este modo, los n_i óptimos son:

$$\begin{aligned}
n_i &= \left\lfloor \frac{n}{k} \right\rfloor & 0 \leq i \leq (k-1 - n \bmod k) \\
n_i &= \left\lceil \frac{n}{k} \right\rceil & (k - n \bmod k) \leq i \leq k-1
\end{aligned} \tag{4.3}$$

Ejemplo 1 Definir la gráfica 7-partita $G = (V, A)$ de orden 100 de tamaño máximo. Dado que el cociente $100/7 = 14$ y el residuo es 2, deberían separarse los vértices de G en dos subconjuntos de orden 15 y cinco de orden 14. Esta es la única partición de G tal que ningún par de subconjuntos V_i tiene tamaños de diferencia mayor a 1.

Una vez conocida la partición óptima puede calcularse el número $M_{\max}(n, k)$ de aristas de una gráfica k -partita de orden n . Con el objeto de simplificar las operaciones denominamos como C al cociente de la división n/k y R al residuo, es decir, $C = \lfloor \frac{n}{k} \rfloor$ y $R = n \bmod k$. Entonces,

$$M_{\max}(n, k) = \frac{n(n-1)}{2} - \sum_{i=0}^{k-1} \frac{n_i(n_i-1)}{2}$$

$$\begin{aligned}
&= \frac{n(n-1)}{2} - \frac{1}{2}(k-R)C(C-1) - \frac{1}{2}R(C+1)C \\
&= \frac{n(n-1)}{2} - \frac{1}{2}C(k(C-1) + 2R)
\end{aligned}$$

Aunque la demostración sólo tiene sentido para $n/k \notin \mathbb{Z}$, el resultado es el mismo cuando la división es entera, en este caso $R = 0$ y desaparece el tercer sumando de la expresión. Considerando los valores de C y R se obtiene la expresión:

$$M_{\max}(n, k) = \frac{n(n-1)}{2} - \frac{1}{2} \left\lfloor \frac{n}{k} \right\rfloor \left(k \left(\left\lfloor \frac{n}{k} \right\rfloor - 1 \right) + 2(n \bmod k) \right) \quad (4.4)$$

◻

Sustituyendo en esta expresión el valor de k como $k = 2$ se vuelve a los valores calculados en la Ecuación 4.1.

Teorema 3 Sea $G = (V, A)$ una gráfica de orden n y tamaño M . Si existe un entero positivo k tal que M cumple la siguiente relación:

$$M > \frac{n(n-1)}{2} - \frac{1}{2} \left\lfloor \frac{n}{k} \right\rfloor \left(k \left(\left\lfloor \frac{n}{k} \right\rfloor - 1 \right) + 2(n \bmod k) \right)$$

entonces el número cromático de G es $\chi(G) > k$.

A partir de la Ecuación 4.4 puede demostrarse que toda gráfica k -partita de orden n y tamaño M , tal que:

$$M = \frac{n(n-1)}{2} - \frac{1}{2} \left\lfloor \frac{n}{k} \right\rfloor \left(k \left(\left\lfloor \frac{n}{k} \right\rfloor - 1 \right) + 2(n \bmod k) \right)$$

es una gráfica k -partita completa. En este caso, dada una k -coloración de la gráfica, ningún conjunto de vértices del mismo color puede superar en más de una unidad la cardinalidad de otro conjunto.

Ejemplo 2 Sea $G = (V, A)$ una gráfica de orden 100. Deseamos colorear a G con 5 colores. En este caso $M_{max}(100, 5) = 4000$ y de este modo si el tamaño de G es mayor que 4000, entonces G no es 5-coloreable. En términos de la densidad se obtiene que para valores superiores a 0.81 no existen gráficas 5-coloreables de orden 100.

El Teorema 3 puede expresarse en función de la densidad ρ de la gráfica. Según la definición de la densidad el tamaño de una gráfica puede escribirse como:

$$M = \rho \cdot \frac{n(n-1)}{2} \quad (4.5)$$

Por otro lado, hemos visto que si el tamaño de una gráfica G es superior a $M_{max}(n, k)$ entonces G no es k -coloreable de modo que $\chi(G) > k$. Sustituyendo la Ecuación 4.5 en las expresiones del Teorema 3 podemos obtener el valor de la densidad a partir del cual una gráfica de orden n no puede ser k -coloreable.

$$\begin{aligned} \rho \frac{n(n-1)}{2} &> \frac{n(n-1)}{2} - \frac{1}{2} \left\lfloor \frac{n}{k} \right\rfloor \left(k \left(\left\lfloor \frac{n}{k} \right\rfloor - 1 \right) + 2(n \bmod k) \right) \\ \Rightarrow n(n-1)(1-\rho) &< \left\lfloor \frac{n}{k} \right\rfloor \left(k \left(\left\lfloor \frac{n}{k} \right\rfloor - 1 \right) + 2(n \bmod k) \right) \end{aligned} \quad (4.6)$$

Aplicemos las siguientes desigualdades:

$$\frac{n}{k} - 1 < \left\lfloor \frac{n}{k} \right\rfloor \leq \frac{n}{k} \quad (4.7)$$

$$0 \leq n \bmod k \leq (k-1) \quad (4.8)$$

En este caso puede escribirse:

$$\left\lfloor \frac{n}{k} \right\rfloor \left(k \left(\left\lfloor \frac{n}{k} \right\rfloor - 1 \right) + 2(n \bmod k) \right) > \left(\frac{n}{k} - 1 \right) k \left(\frac{n}{k} - 2 \right)$$

De este modo, si se retoma la expresión 4.6 tenemos que una gráfica que cumpla la siguiente desigualdad:

$$n(n-1)(1-\rho) \leq \left(\frac{n}{k} - 1 \right) k \left(\frac{n}{k} - 2 \right)$$

tendrá un número cromático mayor a k . Despejando la densidad se obtiene el siguiente resultado.

Proposición 4 *Sea $G = (V, A)$ una gráfica de orden n y densidad ρ . Si existe un entero positivo k tal que*

$$\rho \geq 1 - \frac{k}{n(n-1)} \left(\frac{n}{k} - 1 \right) \left(\frac{n}{k} - 2 \right), \quad (4.9)$$

entonces el número cromático de G es mayor o igual a k .

Por ejemplo, toda gráfica G de orden 200 y densidad superior a 0.68 tiene número cromático $\chi(G) > 3$.

Podríamos haber calculado la Ecuación 4.4 en otros términos, obteniendo en este caso cotas de la densidad, distintas de la Ecuación 4.9. De este modo, si en vez de restar las posibles aristas entre elementos de los mismos subconjuntos V_i , se hubiesen sumado las aristas entre vértices del mismo color, las expresiones habrían cambiado formalmente.

Exponemos a continuación la expresión inicial de dos nuevos cálculos de $M_{max}(n, k)$; sólo hemos efectuado las sumas de manera distinta, como resultado de simplificar la expresión y la cota final calculada. Empleamos otra vez las variables C y R y las desigualdades 4.7 y 4.8. Ahora obtenemos:

$$\begin{aligned}
 M_{max}(n, k) &= \frac{R(C+1)}{2} [(R-1)(C+1) + C(k+R)] \\
 &\quad + \frac{(k-R)C}{2} [R(C+1) + (k-R-1)C], \\
 M_{max}(n, k) &= \frac{1}{2} [C(k-1)(2R+kC) + R(R-1)].
 \end{aligned} \tag{4.10}$$

Si $\rho(G) \geq (2k+n-2)\frac{k-1}{k(n-1)} + \frac{(k-1)(k-2)}{n(n-1)}$ entonces $\chi(G) > k$

También pueden obtenerse las siguientes expresiones.

$$\begin{aligned}
 M_{max}(n, k) &= \frac{1}{2} [(n-C)(k-R)C + (n-C-1)R(C+1)] \\
 M_{max}(n, k) &= \frac{1}{2} [Ck(n-C) + R(n-2C-1)].
 \end{aligned} \tag{4.11}$$

Si $\rho(G) \geq \frac{n}{n-1}\frac{k-1}{k} + \frac{k-1}{n} + \frac{1}{n-1}$ entonces $\chi(G) > k$

Considerando que $n = kC + R$ puede comprobarse que las 3 expresiones obtenidas para $M_{max}(n, k)$ son equivalentes. De esta manera resultan tres cotas distintas de la densidad de una gráfica k - coloreable de orden n . Puede demostrarse que la cota

4.11 es menos restrictiva que la cota 4.9. Por otro lado, para valores $k = 2$ y $n \geq 5$ la cota 4.10 es la más restrictiva de las tres. En este caso, la expresión de la Proposición 4 puede ser sustituida por la expresión 4.10. En el resto de los casos la primera cota es la más ajustada; además podemos calcular cotas más restrictivas a partir de las expresiones $M_{max}(n, k)$, [17].

4.3. Multicoloración de gráficas.

Hemos mencionado que el problema de la asignación de frecuencias puede ser tratado en términos de la teoría de gráficas como un problema de coloración, y de esta forma resolverlo mediante el empleo de técnicas algorítmicas. Por otra parte, en las redes celulares de telefonía móvil es habitual trabajar con más de una frecuencia por celda debido a los requerimientos de tráfico. Esto implica, en términos de la teoría de gráficas, colorear con más de un color (frecuencia) cada vértice de la gráfica resultante.

En estos casos se ha llevado a cabo, en primer término, una adaptación del problema, asignando a cada celda una serie de vértices, tantos como frecuencias tiene asignadas la celda, con el objeto de resolver el problema como si se tratara de una coloración simple. Las adyacencias de los vértices en la gráfica construida se describen así en función de las interferencias entre celdas, de modo que dos vértices son adyacentes si pertenecen a celdas interferentes, siendo el peso de la arista que los une igual a la distancia que deben guardar las frecuencias asignadas en cada celda.

En esta sección desarrollamos un modelo teórico que, combinado con técnicas algorítmicas, permite colorear una gráfica con más de un color por vértice sin necesidad de transformar la gráfica. Por simplicidad, presentamos las técnicas en el caso más sencillo, donde todos los vértices requieren un número de colores igual a 2. Los resultados y métodos descritos son extrapolables para un número mayor de colores e incluso para el caso genérico en que cada vértice lleve asociado un número variable de colores.

Definición 24 *Multicoloración.* Sea una gráfica $G = (V, A)$; se define una r -multicoloración de G , con k colores, como una asignación de colores a los vértices de G de forma que cada vértice tenga asignados r colores diferentes y dos vértices adyacentes no tengan algún color en común.

Nótese la diferencia entre la notación k -coloración y la expresión r -multicoloración con k colores. La primera es reservada para la coloración simple en que cada vértice tiene asignado un color único, escogido de un conjunto de cardinalidad k , y en la segunda cada vértice tiene asignados r colores distintos escogidos de un conjunto de cardinal k . Recordemos también que los colores son representados con enteros.

Definición 25 *El número r -multicromático, $\chi_r(G)$, o más genéricamente el número multicromático de una gráfica G , se define como la cardinalidad del mínimo conjunto de colores necesarios para r -multicolorar G .*

Proposición 5 *Sea $G = (V, A)$ una gráfica. Si $\chi(G) = k \implies \chi_r(G) \leq r \cdot k$*

Demostración: Podemos r -multicolorar la gráfica ejecutando r veces una coloración con k colores: De 1 a k la primera coloración, de $k + 1$ a $2k$ la segunda coloración y así sucesivamente.

■

Por ejemplo, veamos la Figura 4.9. La gráfica G_1 tiene número cromático $\chi(G) = 5$. Queremos asignar 2 colores diferentes a cada vértice, por la Proposición 5, sabemos que tendremos que ocupar a lo más 10 colores, esto es $\chi_r(G) \leq r \cdot k$. La gráfica G_2 tiene número cromático $\chi(G) = 3$, al hacer la 3-multicoloración para G_2 comprobamos que es necesario usar $9 = 3 \cdot 3$ colores.

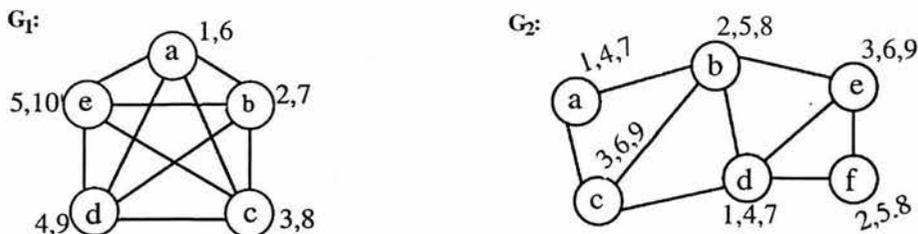


Figura 4.9: Ejemplo de Multicoloraciones con $r \cdot k$ colores.

Es sencillo mostrar la existencia de gráficas para las cuales $\chi_r(G) < r \cdot k$. En general, cualquier ciclo C_n con $n > 3$ e impar, que tiene número cromático $\chi(C_n) = 3$, puede ser 2-multicoloreado con 5 colores. Por ejemplo, el ciclo de orden 5, mostrado en la Figura 4.10 ilustra este caso.

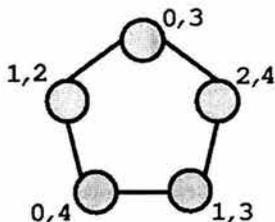


Figura 4.10: Ejemplo de una 2-multicoloración del ciclo C_5 .

Por otra parte, si se tiene una r -multicoloración con $\chi_r(G)$ colores, se puede conseguir una $(r + 1)$ -multicoloración añadiendo a cada vértice un nuevo color, escogido entre $\chi_r(G)$ y $\chi_r(G) + k$ y distribuido según una coloración simple de $\chi(G) = k$ colores.

De este modo,

$$\begin{aligned} \chi(G) &= k; \\ \chi_2(G) &\leq 2k, & \chi_2(G) &= 2k - \alpha_2, & \alpha_2 &\in \mathbb{Z}, \alpha_2 \geq 0; \\ \chi_3(G) &\leq 3k - \alpha_2, & \chi_3(G) &= 3k - \alpha_2 - \alpha_3, & \alpha_3 &\in \mathbb{Z}, \alpha_3 \geq 0; \end{aligned}$$

y en general,

$$\chi_r(G) \leq \chi_{r-1}(G) + k, \quad \chi_r(G) = rk - \sum_{i=2}^r \alpha_i, \quad \alpha_i \in \mathbb{Z}, \quad \alpha_i \geq 0. \quad (4.12)$$

4.4. Cotas generales para el número multicromático.

Presentamos a continuación algunos resultados simples sobre las cotas del número multicromático de una gráfica G .

Proposición 6 *Sea una gráfica $G = (V, A)$ tal que contiene un clan de orden c . En este caso se cumple que,*

$$\chi_r(G) \geq r \cdot c. \quad (4.13)$$

Demostración: Inmediata dado que ninguna de las gráficas r coloreadas podrá utilizar menos de c colores distintos en cada caso. ■

En la Figura 4.11(a), la gráfica G , tiene un clan K de orden 4, donde $V(K) = \{d, e, f, g\}$; hemos dado una 4-coloración válida, por tanto $\chi(G) = 4$. Asignando 2 colores diferentes a cada vértice, tenemos la asignación en la Figura 4.11(b).

Proposición 7 *Dada una gráfica $G = (V, A)$ de orden n , tal que s es el orden del conjunto independiente máximo de G ; se tiene que*

$$\chi_r(G) \geq \left\lceil \frac{r \cdot n}{s} \right\rceil. \quad (4.14)$$

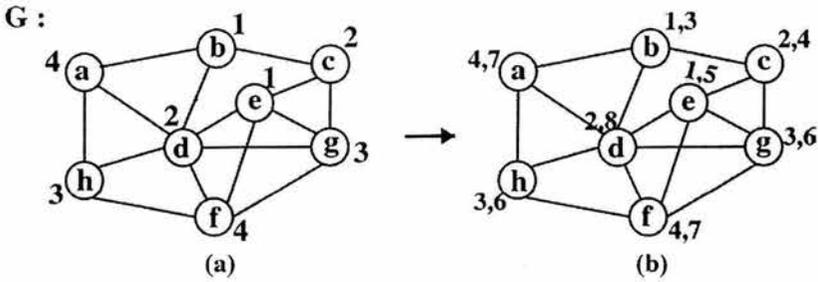


Figura 4.11: Multicoloración de una gráfica G que contiene un clan K_4 .

Demostración: Dado un color j cualquiera, éste puede ser asignado como máximo a s vértices diferentes. Por otra parte, si queremos multicolorar G con r colores en cada vértice tendremos un total de $n \cdot r$ colores asignados a lo largo de toda la gráfica, contando las repeticiones.

Por lo tanto, se obtiene la cota propuesta. ■

En la gráfica G_1 , de la Figura 4.12, el orden del conjunto máximo independiente es 4, el orden de G_1 es $n = 10$; digamos que queremos asignar dos colores por vértice, de acuerdo a la Proposición 7:

$$\chi_r(G) \geq \left\lceil \frac{2 \cdot 10}{4} \right\rceil = \left\lceil \frac{20}{4} \right\rceil = 5,$$

observamos que el máximo color usado es 6 y, en efecto, $6 > 5$.

Proposición 8 Dada una gráfica $G = (V, A)$ conexa y no completa de grado máximo Δ , $\Delta \geq 3$, se cumple que,

$$\chi_r(G) \leq r \cdot \Delta.$$

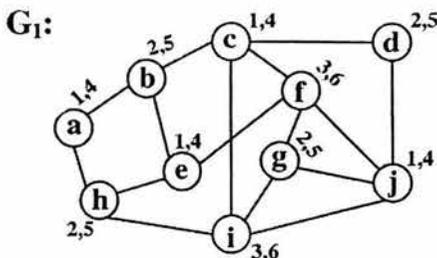


Figura 4.12: Ejemplo de la Multicoloración de la Proposición 7

Demostración: El Teorema de Brooks[16] asegura que si G es una gráfica conexa no completa y no es un ciclo impar, entonces su número cromático cumple $\chi(G) \leq \Delta(G)$. Entonces dada una gráfica $G = (V, A)$, conexa, no completa y con grado máximo $\Delta \geq 3$, podemos considerar el Teorema de Brooks y la Proposición 5 y obtenemos $\chi_r(G) \leq r \cdot \chi(G) \leq r \cdot \Delta$.

■

Regresemos a la gráfica G de la Figura 4.12, el grado máximo de G es $\Delta(G) = 4$. Entonces, tenemos que $\chi_r(G) \leq 2 \cdot \chi(G) \leq r \cdot 4 = 8$. Por lo tanto, la asignación propuesta también cumple la Proposición 8.

Proposición 9 Dada una gráfica $G = (V, A)$ de número r -multicromático $\chi_r(G)$ se cumple que,

$$\chi_r(G) \leq \min\{\chi_i(G) + \chi_j(G) \mid i + j = r\} \leq r \cdot \chi(G).$$

Demostración: La primera desigualdad se deduce del hecho de que para obtener una multicoloración con r colores pueden incorporarse dos multicoloraciones con i y j colores por vértice respectivamente, si $i + j = r$.

Para la segunda desigualdad, $\min\{\chi_i(G) + \chi_j(G) \mid i + j = r\} \leq r \cdot \chi(G)$, consideremos que

$$\chi_i(G) = i \cdot \chi(G) - \Sigma\alpha_i$$

por la ecuación 4.12. Entonces:

$$\begin{aligned} \chi_i(G) + \chi_j(G) &= i \cdot \chi(G) + j \cdot \chi(G) - \Sigma\alpha_i - \Sigma\alpha_j \\ \Leftrightarrow \chi_i(G) + \chi_j(G) &\leq i \cdot \chi(G) + j \cdot \chi(G) = (i + j)\chi(G) \end{aligned}$$

como $(i + j) = r$, entonces:

$$\chi_i(G) + \chi_j(G) \leq r \cdot \chi(G) .$$

■

Con todo lo anterior, podemos deducir fácilmente:

Resultado 2 *El número r -multicromático de una gráfica completa de orden n es $n \cdot r$.*

Calculamos, a continuación, el número cromático de algunas familias de gráficas.

Proposición 10 *Sea $G = (V, A)$ una gráfica bipartita y $A \neq \emptyset$. Se cumple:*

$$\chi_r(G) = 2r.$$

Demostración: Dada una gráfica bipartita $G = (V, A)$ tal que $A(G) \neq \emptyset$ existe al menos una multicoloración de G con $2r$ colores, superponiendo r veces la primera coloración con 2 colores, por lo tanto $\chi_r(G) \leq 2r$. Por otra parte, de la Proposición 6 se tiene que $\chi_r(G) \geq 2r$ ya que toda gráfica bipartita no nula contiene un clan de orden 2. De esta forma $\chi_r(G) = 2r$.

■

Un ejemplo de la proposición anterior, con $r = 3$, es la gráfica G_2 en la Figura 4.13. Podemos observar que la coloración para este tipo de gráficas es muy sencilla, ya que se repite para cada vértice de la partición.

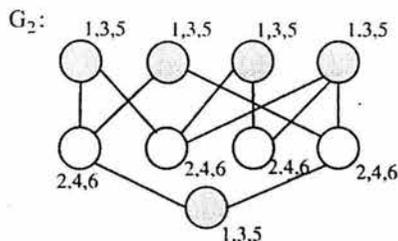


Figura 4.13: La multicoloración de una gráfica bipartita.

Proposición 11 *Sea C_n un ciclo de orden n . Entonces*

$$\chi_r(C_n) = \begin{cases} 2r & \text{si } n \text{ es par} \\ 2r + \lceil \frac{2r}{n-1} \rceil & \text{si } n \text{ es impar} \end{cases}$$

Demostración: En el caso en que el orden n sea par, el ciclo C_n es bipartito, el número cromático se deduce entonces de la Proposición 10. Por otra parte, para colorear un ciclo con un número impar de vértices se necesitan como mínimo 3 colores. Ahora bien, para 2-multicolorear esta gráfica sólo se necesitan 5 colores, dado que se puede aprovechar uno de los colores c de la primera coloración, suponiendo que c sólo se ha empleado una vez en la primera asignación en un vértice no adyacente y colorear los vértices restantes según el esquema de la primera coloración, con dos nuevos colores como en la Figura 4.14.

Este esquema podría repetirse para el tercer y cuarto coloreado, aprovechando siempre el color c de la primera coloración y así sucesivamente hasta que no sea posible reutilizar el color c .

Como el conjunto independiente máximo de C_n tiene orden $(n-1)/2$, el color c puede repetirse un número de veces igual a $(n-1)/2$, pudiéndose reutilizar en total $(n-1)/2 - 1$ ocasiones. De este modo:

$$\chi_1(C_n) = 3 \quad \chi_2(C_n) = 5 \quad \chi_3(C_n) = 7$$

hasta

$$\chi_{\frac{n-1}{2}}(C_n) = 3 + 2 \left(\frac{n-1}{2} - 1 \right) = n.$$

■

Queda por determinar el caso cuando $r > \frac{n-1}{2}$. Antes de realizar la demostración, ejemplificamos este caso. Observemos la Figura 4.14, que ilustra la proposición anterior. Los números representan el color asignado al vértice en cada coloración. Para la siguiente coloración, $r = 2$, se necesitarían entonces 3 nuevos colores, dado que no podría utilizarse de nuevo el color c , ni ninguno de los anteriores, repitiéndose en este punto el mismo proceso anterior. Así en cada incremento de colores, es decir cada vez que pasemos de una r -multicoloración a una $(r + 1)$ -multicoloración, se necesitarán dos nuevos colores, excepto cuando se pasa de r donde el incremento es de 3 unidades.

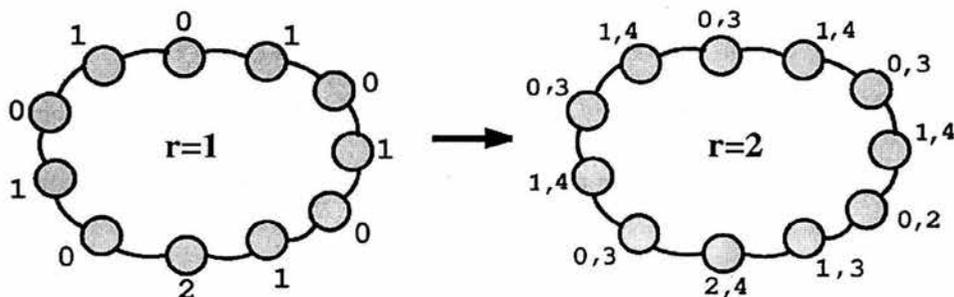


Figura 4.14: Ejemplo de la multicoloración de un ciclo C_n con n impar.

Retomando la demostración, se deduce la siguiente cota para el número multicromático.

$$\chi_r(C_n) \leq 2r + \left\lceil \frac{r}{(n-1)/2} \right\rceil = 2r + \left\lceil \frac{2r}{n-1} \right\rceil$$

Teniendo en cuenta que para un ciclo \mathcal{C}_n impar, el orden del conjunto independiente máximo es $s = (n - 1)/2$, entonces según la Proposición 7 puede obtenerse la siguiente cota inferior del número cromático de \mathcal{C}_n con n impar.

$$\chi_r(\mathcal{C}_n) \geq \left\lceil \frac{rn}{(n-1)/2} \right\rceil = \left\lceil \frac{2rn}{n-1} \right\rceil$$

De este modo, combinando las dos cotas obtenidas se tiene que:

$$\left\lceil \frac{2rn}{n-1} \right\rceil \leq \chi_r(\mathcal{C}_n) \leq 2r + \left\lceil \frac{2r}{n-1} \right\rceil.$$

Ahora bien, dado que $r \in \mathbb{Z}$, podemos reescribir:

$$\left\lceil \frac{2rn}{n-1} \right\rceil = \left\lceil \frac{2r(n-1) + 2r}{n-1} \right\rceil = \left\lceil 2r + \frac{2r}{n-1} \right\rceil = 2r + \left\lceil \frac{2r}{n-1} \right\rceil.$$

quedando finalmente,

$$\chi_r(\mathcal{C}_n) = 2r + \left\lceil \frac{2r}{n-1} \right\rceil = \left\lceil \frac{2rn}{n-1} \right\rceil. \quad (4.15)$$

■

La Figura 4.15 muestra el esquema de multicoloración descrito en la demostración anterior para el caso de un ciclo C_9 . En este caso se cumplen las siguientes desigualdades:

$$\begin{array}{lll} \chi_1(C_9) \leq 3 & \chi_5(C_9) \leq 12 & \chi_9(C_9) \leq 21 \\ \chi_2(C_9) \leq 5 & \chi_6(C_9) \leq 14 & \dots \\ \chi_3(C_9) \leq 7 & \chi_7(C_9) \leq 16 & \\ \chi_4(C_9) \leq 9 & \chi_8(C_9) \leq 18 & \end{array}$$

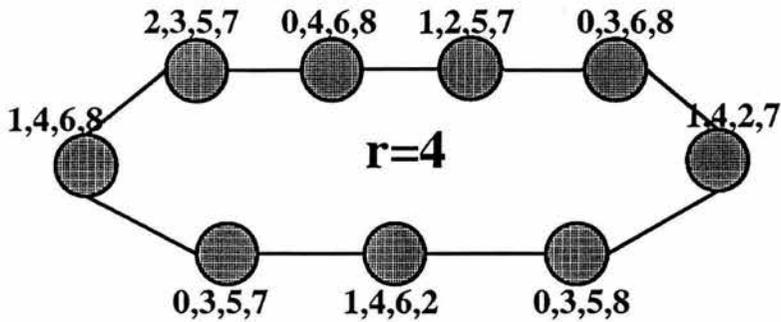


Figura 4.15: Ejemplo de una 4-multicoloración de un ciclo de orden 9.

Asignación de Canales.

Capítulo 5

Asignación de Canales.

Aunque la mayoría de los algoritmos para la asignación de canales revisados en la literatura consideran un modelo estático del problema, se ha probado que muchos de estos pueden ser adaptados fácilmente para solucionar el modelo dinámico [9]. Por otro lado, la demanda máxima posible para los canales en la celda y la asignación fija de canales basada en esta demanda debe ejecutarse tan bien como en la situación dinámica.

En el modelo de gráfica descrito en la introducción, la demanda es representada mediante un entero positivo $w(v)$ asociado a cada nodo v , de la gráfica. Una asignación de valores enteros a los nodos de una gráfica de manera que cumpla ciertas condiciones o restricciones, es referida como una **etiquetación de la gráfica**. La coloración de una gráfica puede ser vista como un caso especial de etiquetación, que satisface la condición de que las etiquetas de nodos adyacentes deben ser distintas.

El marco de la etiquetación de gráficas otorga la posibilidad de incorporar las restricciones de la separación de canales. Se representa estas restricciones mediante una secuencia no decreciente de parámetros enteros positivos: $c_0 \geq c_1 \geq \dots \geq c_k$.

En este modelo suponemos que la distancia de la gráfica relaciona a la distancia entre celdas, requerimos que canales asignados a los nodos a distancia i uno del otro, deben tener una separación de al menos c_i . Así, la restricción c_0 representa la

separación entre canales asignados a la misma celda y es referida como **restricción co-sitio** (*co-site constraint*). Mientras que las restricciones entre celdas diferentes son llamadas **restricciones inter-sitio** (*inter-site constraints*). Frecuentemente, c_0 es grande comparada con las otras restricciones, y el resto, la mayoría de las veces, toman valores pequeños, especialmente 1 ó 2.

En otros términos, una restricción co-sitio indica que los canales asignados a las celdas correspondientes deben ser distintos. Una restricción inter-sitio 2, llamada **restricción de adyacencia de canal**, codifica los requerimientos para los canales asignados a un par de celdas e indica que no puede ser contiguo uno con otro en el espectro de radio.

5.1. Modelación del Problema

Asumimos que para cada celda se da una demanda fija de canales. Se requiere encontrar una asignación que asocie exactamente el número de canales solicitados a cada celda.

Definición 26 Gráfica con restricciones. Una gráfica con restricciones $G = (V, A, c_0, c_1, \dots, c_k)$ es una gráfica $G = (V, A)$ con parámetros enteros positivos c_0, c_1, \dots, c_k , que representan las restricciones entre las etiquetas de $V(G)$; además $c_0 \geq c_1 \geq \dots \geq c_k$.

Estos parámetros representan el espacio del canal pre-establecido para los canales asignados al mismo vértice o a vértices distintos; es decir, c_i representa la restricción que deben satisfacer los vértices a distancia i , o bien c_i representa la diferencia que debe existir entre las etiquetas asignadas a nodos a distancia i . Por consistencia, la restricción entre un par de nodos cuya distancia sea al menos la distancia de reuso se define como cero.

Definición 27 Gráfica con pesos y restricciones. Una gráfica con pesos y restricciones es el par (G, \mathbf{w}) , donde G es una gráfica con restricciones y \mathbf{w} es un vector de pesos positivos para los vértices de G .

El componente de \mathbf{w} correspondiente al nodo u se denota por $w(u)$, se conoce como el **peso del vértice u** y representa el número de llamadas que serán atendidas por el vértice. Usaremos w_{max} para denotar $\max\{w(u) \mid v \in V(G)\}$ y w_{min} para denotar el correspondiente peso mínimo $\min\{w(u) \mid v \in V(G)\}$. Además, para cualquier conjunto $S \subseteq V$, usaremos $w(S)$ para denotar la suma de los pesos de todos los vértices en S .

Una definición formal de la asignación de canales puede ser dada en el contexto de nuestro modelo en Teoría de Gráficas:

Definición 28 Asignación de canales. Una asignación de canales para una gráfica con pesos y restricciones (G, \mathbf{w}) , donde $G = (V, A, c_0, \dots, c_k)$, es una asignación f de conjuntos de enteros no negativos (que representan los canales) a los vértices de G , que satisface las siguientes condiciones:

$$I) |f(u)| = w(u), \quad \forall u \in V.$$

$$II) i \in f(u) \text{ y } j \in f(v) \Rightarrow |i - j| \geq c_l \quad \forall u, v \in V \text{ tales que } d_G(u, v) = l.$$

El ancho de banda utilizado por una asignación de canales está representado mediante su **amplitud**, (*span*). La amplitud $\sigma(f)$ de una asignación f , en una gráfica con restricciones y pesos, es la diferencia entre los canales mayor y menor asignados por f . La amplitud de una gráfica con restricciones y pesos (G, \mathbf{w}) se denota por $\sigma(G, \mathbf{w})$, y representa la amplitud mínima de cualquier asignación de canales para la gráfica con pesos y restricciones (G, \mathbf{w}) .

5.2. Etiquetación a distancia 2

En esta sección revisaremos la etiquetación de gráficas, considerando los vértices a distancia 2. Antecedentes de esta clase de etiquetación son la T -coloración de gráficas, material que ha sido ampliamente estudiado en la literatura [25, 36, 21, 20, 23]. En el artículo *T-colorings of graphs and the channel assignment problem* [24], Roberts propuso el problema de la asignación eficiente de frecuencias de radio para transmitir a diferentes localidades, usando enteros no negativos para representar las frecuencias.

El material aquí presentado está basado en el artículo de Jerrold R. Griggs y Roger K. Yeh [6], donde se propone un problema análogo para una gráfica $G = (V, A)$ y se busca caracterizar la etiquetación de una gráfica mediante números enteros.

Definición 29 $\mathcal{L}_d(2, 1)$ -etiquetación de G . Dado un número real $d > 0$, una $\mathcal{L}_d(2, 1)$ -etiquetación de G es una función, evaluada en \mathbb{R}^+ ,

$$f : V(G) \mapsto [0, \infty)$$

tal que para cualesquiera vértices u y v , se tiene que:

$$\text{i) Si } d_G(u, v) = 1 \Rightarrow |f(u) - f(v)| \geq 2d ;$$

$$\text{ii) Si } d_G(u, v) = 2 \Rightarrow |f(u) - f(v)| \geq d.$$

El número $\lambda(G, d)$ de una $\mathcal{L}_d(2, 1)$ -etiquetación de G es el número m más pequeño de manera que $G = (V, A)$ tiene una $\mathcal{L}_d(2, 1)$ -etiquetación de G donde ninguna etiqueta es mayor que m . Si f es una $\mathcal{L}_d(2, 1)$ -etiquetación de G , entonces se dice que $f \in \mathcal{L}_d(2, 1)(G)$.

Sea $G = (V, A)$ una gráfica y $f \in \mathcal{L}_d(2, 1)(G)$. Se define la norma $\|f(G)\|$ como:

$$\|f(G)\| = \max\{f(v) \mid v \in V(G)\}.$$

Entonces $\lambda(G, d) = \min \|f(G)\|$, donde el mínimo se toma entre todas las etiquetaciones $f \in \mathcal{L}_d(2, 1)(G)$.

A continuación ejemplificamos estas definiciones. Consideremos la gráfica G de la Figura 5.1(a). Las Figuras 5.1(b) y 5.1(c) muestran diferentes $\mathcal{L}_1(2, 1)$ -etiquetaciones para G . Las etiquetas dadas en las Figuras 5.1(c) y 5.1(d) son óptimas y entonces $\lambda(G, 1) = 5$. Por otra parte, $\|f_0(G)\| = 10$ y $\|f_1(G)\| = 5$

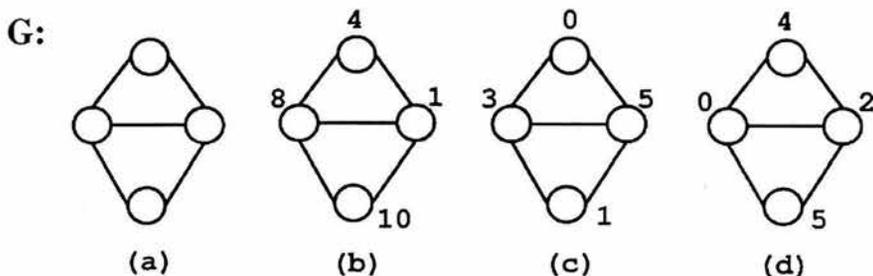


Figura 5.1: $\mathcal{L}_1(2, 1)$ -etiquetación.

5.3. Reducción a etiquetas con valores enteros

Primero caracterizaremos a $\lambda(G, d)$ en términos de $\lambda(G, 1)$ y posteriormente demostraremos que determinar $\lambda(G, 1)$ es suficiente para estudiar el caso cuando la etiquetación es con valores enteros.

Lema 12 Para toda gráfica $G = (V, A)$ y entero d , se cumple que $\lambda(G, d) = d \cdot \lambda(G, 1)$.

Demostración: Demostraremos este lema mediante dos afirmaciones:

1. Se cumple que $\lambda(G, d) \geq d \cdot \lambda(G, 1)$.

Sea $f \in \mathcal{L}_d(2,1)(G)$. Definimos $f_1(u) = f(u)/d$ para todo $u \in V(G)$. Se sigue que $f_1 \in \mathcal{L}_1(2,1)(G)$. Esto implica que $\|f(G)\|/d = \|f_1(G)\| \geq \lambda(G,1)$.

Por lo tanto, $d \cdot \lambda(G,1)$ es una cota inferior para $\|f(G)\|$ y la suposición es verdadera. ■

2. Se cumple $\lambda(G,d) \leq d \cdot \lambda(G,1)$.

Sea $f \in \mathcal{L}_1(2,1)(G)$. Definimos $f_1(u) = f(u) \cdot d$ para todo $u \in V(G)$. Por tanto, $f_1 \in \mathcal{L}_d(2,1)(G)$. Esto implica que $\|f(G)\| \cdot d = \|f_1(G)\| \geq \lambda(G,1)$. ■

Por tanto, el Lema 12 se cumple. ■

Lema 13 Sean $x \cdot y \geq 0$, $d > 0$ y $k \in \mathbb{Z}^+$. Si $|x - y| \geq k \cdot d$, entonces $|x' - y'| \geq k \cdot d$, donde $x' = \lfloor x/d \rfloor d$ y $y' = \lfloor y/d \rfloor d$.

Los lemas anteriores implican el siguiente teorema:

Teorema 14 Dada una gráfica G , existe una $f \in \mathcal{L}_1(2,1)(G)$ con valores enteros y $\|f(G)\| = \lambda(G,1)$.

Este resultado indica que siempre podemos encontrar una etiquetación con valores enteros, donde es posible obtener el valor mínimo.

Para una generalización veamos que $\lambda(G, d)$ se alcanza por alguna f en $\mathcal{L}_d(2, 1)(G)$ cuyos valores son todos múltiplos de d . Es decir, $f = d \cdot f'$ donde $f' \in \mathcal{L}_1(2, 1)(G)$, evaluada en los enteros, por el Lemà 12. Por tanto, es suficiente estudiar el caso donde $d = 1$ y considerar sólo $f \in \mathcal{L}_1(2, 1)(G)$ con valores enteros.

5.4. Trayectorias y ciclos

En esta sección presentamos resultados que caracterizan la $\mathcal{L}(2, 1)$ -etiquetación de G cuando la gráfica es una trayectoria o un ciclo.

Trabajemos primero la $\mathcal{L}(2, 1)$ -etiquetación de G en una gráfica sencilla, como una trayectoria.

Proposición 15 *Sea P_n una trayectoria de n vértices. Entonces*

- I) $\lambda(P_2) = 2$;
- II) $\lambda(P_3) = \lambda(P_4) = 3$;
- III) $\lambda(P_n) = 4$ para $n \geq 5$.

Ahora ejemplificamos la Proposición anterior. En cada una de las trayectorias de la Figura 5.2 el número debajo de cada vértice representa el color asignado al vértice con etiqueta v_i .

En la Figura 5.2(a) definimos una etiquetación f válida, usando el par de etiquetas 0 y 2; observemos que la Proposición 15 se cumple para la trayectoria P_2 y se seguiría cumpliendo aún cuando se elijan las etiquetas (1 y 3) ó (0 y 3) aunque si es esta última entonces la amplitud $\sigma(f) = 3$. Por tanto, usamos f con una amplitud mínima.

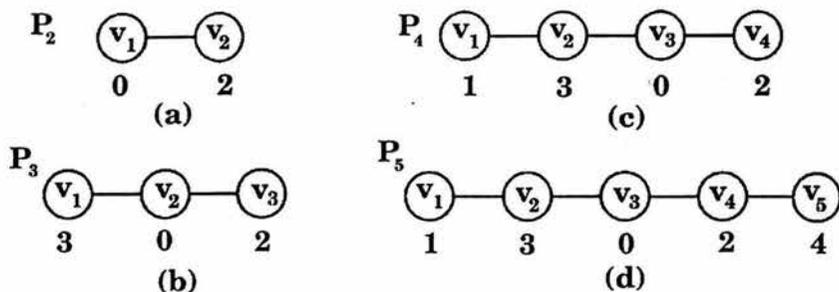


Figura 5.2: Etiketación de Trayectorias.

Para P_3 copiamos la etiquetación propuesta para P_2 y ya que el objetivo es conservar la amplitud mínima, de acuerdo a las restricciones de la $\mathcal{L}(2, 1)$ -etiquetación, la siguiente etiqueta disponible es 3. Procedemos de la misma manera para P_4 , considerando que la etiqueta 1 está disponible, tenemos que se cumple que $\lambda(P_3) = \lambda(P_4) = 3$.

Por último, observemos la Figura 5.2(d). Con la etiquetación definida podemos generalizar que para trayectorias P_n cuando $n \geq 5$ es posible establecer etiquetaciones periódicas, ya que P_n tendrá como subgráfica a P_5 la cual ya podemos etiquetar cumpliendo $\lambda(P_5) = 4$.

Si unimos los vértices primero y último de una trayectoria P_n , $n \geq 3$, entonces tenemos un ciclo C_n . Ahora determinemos su número de etiquetación, tomando en cuenta las etiquetaciones propuestas para las trayectorias.

Proposición 16 Sea C_n un ciclo de longitud n , entonces $\lambda(C_n) = 4$, para cualquier n , $n \geq 3$.

Supongamos que $n \geq 5$. Para toda $n \geq 5$, C_n debe contener una trayectoria P_5 como subgráfica. De aquí que $\lambda(C_n) \geq \lambda(P_5) = 4$, por la Proposición 15.

Ahora demosremos que $\lambda(C_n) \leq 4, n \geq 5$. Para ello es suficiente demostrar que existe una $\mathcal{L}_d(2, 1)$ -etiquetación de G f , tal que $|f(C_n)| = 4$.

Sean v_0, \dots, v_{n-1} vértices de C_n tales que $d_G(v_i, v_{i+1}) = 1, 0 \leq i \leq n-2$ y $(v_0, v_{n-1}) \in A(C_n)$.

Se propone la siguiente etiquetación:

1. Si $n \equiv 0 \pmod{3}$, entonces definimos

$$f(v_i) = \begin{cases} 0 & \text{Si } i \equiv 0 \pmod{3}; \\ 2 & \text{Si } i \equiv 1 \pmod{3}; \\ 4 & \text{Si } i \equiv 2 \pmod{3}. \end{cases}$$

2. Si $n \equiv 1 \pmod{3}$, entonces redefinimos la asignación f anterior en v_{n-4}, \dots, v_{n-1} como:

$$f(v_i) = \begin{cases} 0 & \text{Si } i = n-4; \\ 3 & \text{Si } i = n-3; \\ 1 & \text{Si } i = n-2; \\ 4 & \text{Si } i = n-1. \end{cases}$$

3. Si $n \equiv 2 \pmod{3}$, entonces redefinimos la asignación f en v_{n-2} y v_{n-1} dada en 1 como:

$$f(v_i) = \begin{cases} 1 & \text{Si } i = n-2; \\ 3 & \text{Si } i = n-1. \end{cases}$$

La etiquetación propuesta anteriormente se ilustra en las gráficas de la Figura 5.3. En (a) se presenta una asignación para C_6 , es decir, cuando $n \equiv 0 \pmod 3$; En (b) se proporciona una asignación para C_{10} , esto es cuando $n \equiv 1 \pmod 3$. Finalmente, en (c) realizamos una asignación para C_{11} , es decir, cuando $n \equiv 2 \pmod 3$. Obsérvese que tienen similitud con una etiquetación periódica de trayectorias, en este caso con P_3 , ajustando sólo la “parte final” del ciclo, de acuerdo a la numeración de los vértices.

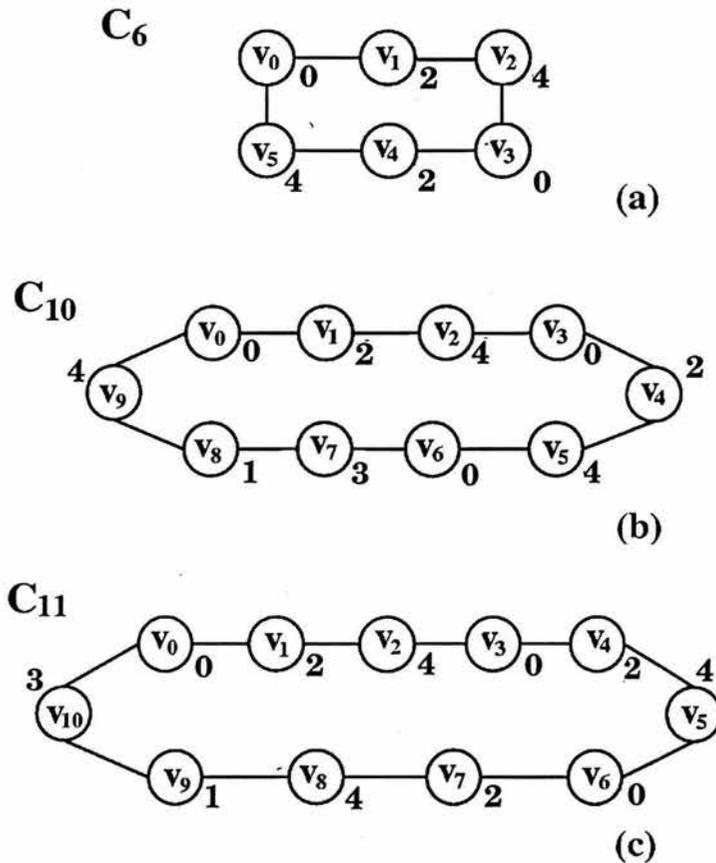


Figura 5.3: Etiquetaciones de C_n , con base en la Proposición 16

5.5. Gráficas k -Coloreables.

Ahora revisemos gráficas con un número cromático $\chi(G)$ específico.

Teorema 17 Sea $G = (V, A)$ una gráfica con $\chi(G) = k$ y $|V(G)| = v$.

Entonces $\lambda(G) \leq v + k - 2$.

Demostración: Dado que $\chi(G) = k$, podemos particionar a G en k conjuntos

G_1, \dots, G_k donde $|V(G_i)| = s_i$ y cada G_i es un conjunto independiente.

Sea $V_i = V(G_i) = \{v_{i1}, v_{i2}, \dots, v_{is_i}\}$, donde $1 \leq i \leq k$.

Ahora consideremos la etiquetación f definida por $f(v_{1j}) = j - 1$ con $1 \leq j \leq s_1$.

$$f(v_{ij}) = \sum_{t=1}^{i-1} s_t + i + j - 2$$

para $1 \leq j \leq s_i$ y $2 \leq i \leq k$.

De aquí verificamos que f está en $\mathcal{L}(2, 1)(G)$

Por lo tanto, $\lambda(G) \leq \|f(G)\| = s + k - 2$.

■

Corolario 18 Sea $G=(V,A)$ una gráfica k -partita completa con $|V(G)| = s$. Entonces $\lambda(G) = s + k - 2$.

Demostración:

- a) Dado que G es una gráfica k -partita, $\chi(G) = k$. Por el Teorema 17, $\lambda(G) \leq s + k - 2$.

- b) Por otra parte, la distancia entre cualesquiera dos vértices en G es a lo más 2, entonces las etiquetas deben ser distintas.

Además etiquetas consecutivas no pueden ser usadas en vértices de partes diferentes y hay k componentes, entonces:

$$\lambda(G) \geq s + k - 2.$$

De esta manera $\lambda(G) = v + k - 2$.

■

5.6. Complejidad del problema de la $\mathcal{L}(2, 1)$ -etiquetación de G .

Como sabemos, el problema de encontrar el número cromático de una gráfica es un problema NP-completo. Dado que el problema de la $\mathcal{L}(2, 1)$ -etiquetación de G es similar, veamos si este problema también es NP-completo.

Consideremos el siguiente planteamiento del problema de la $\mathcal{L}(2, 1)$ -etiquetación:

Ejemplar: Sea $G = (V, A)$, una gráfica con diámetro 2.

Pregunta: ¿Es $\lambda(G,) \leq |V|$?

Teorema 19 *El problema de la $\mathcal{L}(2, 1)$ -etiquetación de G es NP-completo.*

Antes de presentar la demostración de este resultado, presentamos un par de lemas que nos faciliten la demostración.

Lema 20 *Las siguientes afirmaciones son equivalentes:*

1. *Existe una función inyectiva $f : V(G) \rightarrow [0, |V| - 1]$ tal que $|f(u) - f(v)| \geq 2$ para toda $(u, v) \in A(G)$.*
2. *G^c contiene una trayectoria Hamiltoniana, es decir, una trayectoria que pasa por todos los vértices de G .*

Demostración:

(1) \Rightarrow (2). Sea f una función inyectiva definida en $V(G)$ que satisface las condiciones en 1. Ya que f es inyectiva, f^{-1} existe. Ordenemos los vértices en V como sigue:

$$v_i = f^{-1}(i), \quad 0 \leq i \leq |V| - 1.$$

Entonces v_i es adyacente a v_{i-1} en G^c para $0 \leq i \leq |V| - 1$. Por tanto, la trayectoria $\{v_0, v_1, \dots, v_{|V|-1}\}$ es una trayectoria Hamiltoniana en G^c .

(2) \Rightarrow (1) Sea $P = \{v_0, v_1, \dots, v_{|V|-1}\}$ una trayectoria Hamiltoniana en G^c . Definamos una función $f : V(G) \rightarrow [0, |V| - 1]$ donde $f(v_i) = i$, $0 \leq i \leq |V| - 1$.

Dado que P es Hamiltoniana, v_i es único, por tanto f es inyectiva. Sea $(u, v) \in A(G)$. Entonces $f(u) = f(v_i) = i$ y $f(w) = f(v_j) = j$, para alguna i, j con $|i - j| \geq 2$, ya que u no es adyacente a w en G^c .

Entonces f es la función inyectiva que necesitamos. ■

Lema 21 Sean $G_1 = (V_1, A_1)$ una gráfica de diámetro 2 y $G = (V, A)$ una gráfica, tales que:

- $V_1 = V \cup \{u\}$,
- $A_1 = A \cup \{(u, v) \mid \text{para todo } v \in V\}$.

Entonces existe una función inyectiva $f : V(G) \rightarrow [0, |V| - 1]$, tal que $|f(u) - f(v)| \geq 2$ para cada $(u, v) \in A(G)$ si y sólo si $\lambda(G_1) \leq |V_1|$.

Demostración:

Supongamos que hay una función inyectiva f , definida sobre V , que satisface la condición anterior. Definamos $g(v) = f(v)$, $\forall v \in V$ y $g(u) = |V| + 1 = |V_1|$. También $g \in \mathcal{L}(2, 1)(G_1)$ y $\|g(G_1)\| = |V| + 1 = |V_1|$.
De aquí, $\lambda(G_1) \leq |V_1|$.

Por otra parte, supongamos que se cumple $\lambda(G_1) \leq |V_1|$, es decir, existe una función g en $\mathcal{L}(2, 1)(G_1)$, tal que: $\|g(G_1)\| \leq |V| + 1$. Ahora supongamos que $g(u) \neq 0$ o bien $g(u) \neq |V| + 1$. Por las condiciones de $\mathcal{L}(2, 1)(G_1)$, no existe $v \in V$ tal que $g(v) \neq g(u) + 1$ o $g(v) \neq g(u) - 1$. Esto nos lleva a usar $|V| + 3$ números para etiquetar a V_1 .

Esto es una contradicción, porque $\|g(G_1)\| \leq |V| + 1$ y todas las etiquetas son distintas. Por tanto, $g(u) = 0$ o bien $g(u) = |V| + 1$.

Si $g(u) = |V| + 1$, entonces restringir g a V nos da la función inyectiva f que buscamos. Por otro lado, si $g(u) = 0$, entonces restringimos $g - 2$ a V para obtener la función inyectiva f .

Con lo que queda demostrado el Lema 21. ■

Demostración del Teorema 19.

Para demostrar que el problema de la $\mathcal{L}(2, 1)$ -etiquetación es NP-completo, revisemos el siguiente problema de decisión.

Ejemplar: Una gráfica $G = (V, A)$

Pregunta: ¿Existe una función inyectiva $f : V \rightarrow [0, |V| - 1]$
tal que $|f(u) - f(v)| \geq 2$ para cualquier $(u, v) \in A(G)$?

Para contestar esta pregunta, requerimos el Lema 20. De acuerdo a este lema, la característica de NP-completo de la etiquetación inyectiva a distancia 2, que denotaremos como IED , surge como consecuencia de la condición de NP-completo del problema de la trayectoria Hamiltoniana, el cual citamos a continuación:

Ejemplar: Sea $G = (V, A)$.

Pregunta: ¿Existe una trayectoria Hamiltoniana en G ?

Observemos que la $\mathcal{L}(2, 1)$ -etiquetación de G está en NP. Una gráfica $G = (V, A)$ puede transformarse en un ejemplar en tiempo del orden $O(|V| + |A|)$, y puede verificarse en tiempo polinomial que G tiene diámetro 2, esto es una etiquetación f está en $\mathcal{L}(2, 1)$ -etiquetación de G y es tal que $\|f(G)\| \leq |V|$.

Falta demostrar que el problema de la $\mathcal{L}(2, 1)$ -etiquetación es NP-completo; lo haremos mediante la transformación de IED a una $\mathcal{L}(2, 1)$ -etiquetación.

Sea $G = (V, A)$ cualquier gráfica en los ejemplares de IED .
Construimos una gráfica $G' = (V', A')$ como sigue:

Añadimos un vértice u a $V'(G)$, de manera que sea adyacente a todo vértice en $V(G)$ es decir, $G' = (V', A')$ donde $V' = V \cup \{u\}$ y $A' = A \cup \{ (u, v) \mid \text{para todo } v \in V \}$. Entonces $|V'| = |V| + 1$ y $\text{diam } G' = 2$.

La condición de NP-completo de la $\mathcal{L}(2,1)$ -etiquetación de G es consecuencia de la NP-completez de IED y del Lema 21.

Con lo que queda demostrado que la $\mathcal{L}(2,1)$ -etiquetación de G es un problema NP-completo.

■

Capítulo 6

Algoritmos para la Asignación de Canales.

En este capítulo hacemos una revisión de algoritmos para la asignación de canales con restricciones generales. Algunos de estos algoritmos son adaptaciones de algoritmos de multicoloración de gráficas, mientras otros están basados en la etiquetación de gráficas. La mayoría del trabajo hecho es para el caso donde sólo se conocen las restricciones co-sitio, c_0 , y a distancia uno, c_1 .

Además, al igual que en la multicoloración, una coloración base de una gráfica $G = (V, A)$ con un color por vértice puede ser usada para generar una coloración para un problema de asignación de canales y pesos que tiene a G como gráfica subyacente.

6.1. Algoritmo de Asignación para gráficas k -coloreables.

Con este algoritmo a cada vértice pueden serle asignados sus canales de manera independiente del resto de la red, es decir, éste es un algoritmo completamente distribuido; la única información necesaria para que cada nodo sea capaz de calcular su asignación, es su color base.

ALGORITMO A.

Para gráficas con $\chi(G) = k$.

PRECONDICIONES:

$G = (V, A, c_0, c_1)$ una gráfica con restricciones.

w un vector de pesos arbitrario.

$f_1 : V \mapsto \{0, 1, \dots, k-1\}$ una coloración base de $V(G)$ que satisface c_0, c_1 .

ASIGNACION.

$s = \text{máx}\{c_0, kc_1\}$.

para cada *vertice* $v \in V(G)$ **hacer**

para cada $i \in \{0, 1, \dots, w(v) - 1\}$ **hacer**

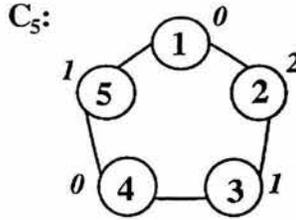
$f(v) = f_1(v) + is$

Cuadro 6.1: ALGORITMO A.

A es un algoritmo sencillo para multicolorar gráficas, ya que sólo basta calcular un parámetro, s , que tiene un valor mayor o igual al número cromático de la gráfica y representa el valor que será sumado a la etiquetación inicial para obtener el siguiente color del vértice.

Veamos cómo se hace la etiquetación; tomemos como ejemplo el vértice 1 de la gráfica C_5 en la Figura 6.1, por sencillez digamos que $c_0 = c_1 = 1$.

La etiquetación inicial de 1 es $f_1(1) = 0$, y el número cromático de C_5 es 3. Por tanto $s = \text{máx}\{1, 3 \cdot 1\} = 3$. Digamos que $w(1) = 2$, de esta manera tenemos que: $f(1) = 0, 3$.

Figura 6.1: Un ciclo C_5 , con restricciones $c_0 = 1$, $c_1 = 1$

Hacemos lo mismo con los otros vértices y resulta la etiquetación en la siguiente tabla, para C_5 .

v	1	2	3	4	5
$w(v)$	2	2	2	2	2
$f_1(v)$	0	2	1	0	1
$f(v)$	0, 3	2,5	1, 4	0,3	1,4

Para este ejemplo, la amplitud de la etiquetación es $\sigma(f) = 5$.

Consideremos ahora la gráfica $G = (V, A)$, en 6.2.

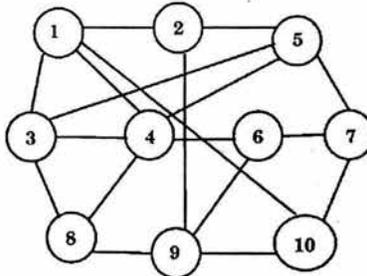


Figura 6.2: Gráfica 3-Coloreable

De acuerdo a las Proposiciones 5 y 8, de la sección de multicoloración, si queremos 3-multicolorar la gráfica G ocuparemos al menos 9 colores y como máximo 15 colores. Revisemos la asignación del ALGORITMO A.

v	1	2	3	4	5	6	7	8	9	10
$w(v)$	3	3	3	3	3	3	3	3	3	3
$f_1(v)$	0	1	1	2	0	0	2	0	2	1
$f(v)$	0, 3, 6	1,4, 7	1, 4, 7	2, 5, 8	0, 3, 6	0, 3, 6	2, 5, 8	0, 3, 6	2, 5, 8	1, 4, 7

Cuadro 6.2: Etiquetación para $G=(V,A,1,1)$

Observemos que la etiquetación anterior se ajusta a las cotas propuestas. Ahora veamos cómo se ejecuta el Algoritmo A cuando las demandas para cada vértice son distintas, modificando también las restricciones. A continuación se muestra la tabla con la información de etiquetación inicial y restricciones para G .

v	1	2	3	4	5	6	7	8	9	10
$w(v)$	10	9	9	7	7	6	6	4	4	3
$f_1(v)$	0	1	1	2	0	0	2	0	2	1

Cuadro 6.3: Etiquetación inicial para $G=(V,A,3,1)$

En este caso, no es posible considerar una r -multicoloración general, ya que existen más restricciones y las demandas son diferentes para cada vértice. De cualquier manera podemos establecer cotas; dado que G es una gráfica conexa, no completa, tiene grado $\Delta(G) = 5$, con demanda máxima $w_{max} = 10$; además contiene un conjunto independiente S de tamaño 4.

De acuerdo a las Proposiciones 7 y 8, la multicoloración de la grafica G ocupará al menos 15 colores y como máximo 30. Veamos la etiquetación resultante:

V	$f(v)$
1	0, 3, 6, 9, 12, 15, 18, 21, 24, 27
2	1, 4, 7, 10, 13, 16, 19, 22, 25
3	1, 4, 7, 10, 13, 16, 19, 22, 25
4	2, 5, 8, 11, 14, 17, 20
5	0, 3, 6, 9, 12
6	0, 3, 6, 9, 12, 15
7	2, 5, 8, 11, 14, 17
8	0, 3, 6, 9
9	2, 5, 8, 11
10	1, 4, 7

Cuadro 6.4: Asignaciones con el ALGORITMO A.

Para la gráfica 6.2 obtenemos una asignación con $\sigma(f) = 27$, que se ajusta a las cotas.

Particularmente, la coloración base usada en el ALGORITMO A puede verse como una etiquetación que satisface la restricción $c_1 = 1$. Podemos formular una versión de este algoritmo basada en una etiquetación de la gráfica.

6.2. Algoritmo de Asignación para gráficas etiquetadas.

El método de repetir una asignación de un canal por vértice es denominado como **Asignación Fija**, puesto que cada vértice tiene un conjunto fijo de canales disponibles para su asignación. Un tipo de etiquetación que otorga etiquetas regulares y periódicas para mallas ya fue definida por Jan van den Heuvel y otros en [4] y es llamada **etiquetación mediante progresión aritmética**

Definición 30 *Etiquetación mediante progresión aritmética.* Una etiquetación f de una malla t -dimensional es una etiquetación mediante progresión aritmética si existen enteros no negativos a_1, \dots, a_t y n tales que para cada vértice v con coordenadas (m_1, \dots, m_t) , $f(v) = a_1 m_1 + \dots + a_t m_t \bmod m$. Al parámetro n se le llama la **amplitud cíclica** (*cyclic span*) de la etiquetación.

Definición 31 *Etiquetación óptima.* Una etiquetación mediante progresión aritmética es considerada **óptima** para un conjunto de restricciones, si su amplitud cíclica es lo más pequeña posible.

Dada f una etiquetación mediante progresión aritmética, $f(\mathbf{m}_1, \mathbf{m}_2)$ denota el valor de la etiquetación del vértice con coordenadas (m_1, m_2) . Las etiquetas mediante progresión aritmética son fáciles de definir y con el algoritmo A' pueden ser usadas para encontrar algoritmos de asignación de canales. Más aún, su regularidad puede ser útil para diseñar métodos que den mejores asignaciones de canal para pesos no uniformes.

A' también es un algoritmo completamente distribuido, ya que una vez que se conoce la amplitud cíclica, la única información necesaria para cada vértice es el valor de su etiquetación.

ALGORITMO A'.

Basado en una etiquetación de la gráfica.

PRECONDICIONES:

$G = (V, A, c_0, c_1, \dots, c_k)$ una gráfica con restricciones.

w un vector de pesos arbitrario.

$f_1 : V \mapsto \mathbb{N}$ una etiquetación dada, que satisface las constantes c_1, \dots, c_k .

M la *amplitud cíclica* de f .

ASIGNACIÓN.

$s = \text{máx}\{c_0, M\}$.

para cada $v \in V(G)$ **hacer**

para todo $i \in \{0, 1, \dots, w(v) - 1\}$ **hacer**

$$f(v) = f_1(v) + is$$

Cuadro 6.5: ALGORITMO A'.

El número que se suma a la etiquetación inicial se calcula considerando la amplitud cíclica de la gráfica, por lo que podemos esperar que sea más ajustado que el cálculo en el ALGORITMO A.

Consideremos la gráfica 6.3, con restricciones $c_0 = 4$, $c_1 = 3$, $c_2 = 3$. Esta gráfica con la etiquetación propuesta tiene un amplitud cíclica $M = 16$, entonces $s = 16$.

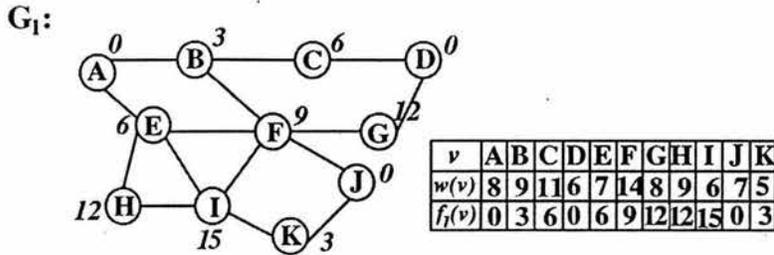


Figura 6.3: Gráfica $G_1 = (V, A, 4, 3, 3)$.

Una vez que conocemos el parámetro s basta con sumar múltiplos de s a la etiquetación inicial; de esta manera tenemos la etiquetación en el Cuadro 6.6, para la gráfica G_1 .

v	$f(v)$
A	0, 16, 32, 48, 64, 80, 96, 112
B	3, 19, 35, 51, 67, 83, 99, 115, 131
C	6, 22, 38, 54, 70, 86, 102, 118, 134, 150, 166
D	0, 16, 32, 48, 64, 80
E	6, 22, 38, 54, 70, 86, 102
F	9, 25, 41, 57, 73, 89, 105, 121, 137, 153, 169, 185, 201, 217
G	12, 28, 44, 60, 76, 92, 108, 124
H	12, 28, 44, 60, 76, 92, 108, 124, 140
I	15, 31, 47, 63, 79, 95
J	0, 16, 32, 48, 64, 80, 96
K	3, 19, 35, 51, 67

Cuadro 6.6: Asignación de la gráfica 6.3 con el ALGORITMO A'.

También podemos considerar una gráfica G con un número mayor de restricciones, siempre y cuando el diámetro de G sea mayor o igual al número de restricciones. Examinemos la gráfica $G = (V, A, 5, 3, 2, 2, 1, 1)$, en la Figura 6.4. G tiene diámetro $D = 5$, por ello es válido considerar las restricciones hasta c_5 .

G:

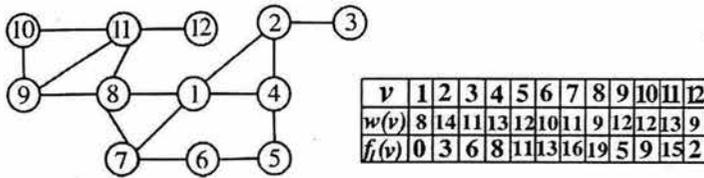


Figura 6.4: Gráfica $G = (V, A, 5, 3, 2, 2, 1, 1)$.

La mayor etiqueta asignada por f_1 es $f_1(8) = 19$; la amplitud cíclica de G es $M = 22$; con respecto al primer ejemplo, la amplitud de f_1 es mayor, debido a que existen más restricciones. Sólo hay que sumar múltiplos de 22 a la etiquetación $f(v)$ hasta cubrir la demanda. Veamos las asignaciones en el Cuadro 6.7.

En una gráfica particular tal como un ciclo C_n , es posible considerar hasta $\lfloor n/2 \rfloor$ restricciones, también sabemos que $\chi(C_n)$ es 2 ó 3. Sea el ciclo C_8 , con las restricciones $c_0 = 4, c_1 = 3, c_2 = 3, c_3 = 2, c_4 = 1$, en la Figura 6.5.

C_8 :

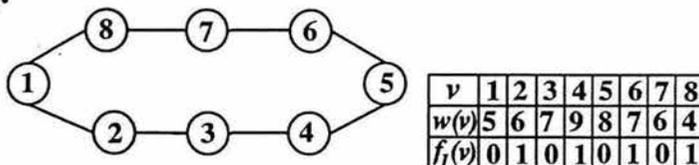


Figura 6.5: Gráfica $C_8 = (V, A, 3, 2)$.

v	$f(v)$
1	0, 22, 44, 66, 88, 110, 132, 154
2	3, 25, 47, 69, 91, 113, 135, 157, 179, 201, 223, 245, 267, 289
3	6, 28, 50, 72, 94, 116, 138, 160, 182, 204, 226
4	8, 30, 52, 74, 96, 118, 140, 162, 184, 206, 228, 250, 272
5	11, 33, 55, 77, 99, 121, 143, 165, 187, 209, 231, 253
6	13, 35, 57, 79, 101, 123, 145, 167, 189, 211
7	16, 38, 60, 82, 104, 126, 148, 170, 192, 214, 236
8	19, 41, 63, 85, 107, 129, 151, 173, 195
9	5, 27, 49, 71, 93, 115, 137, 159, 181, 203, 225, 247
10	9, 31, 53, 75, 97, 119, 141, 163, 185, 207, 229, 251
11	15, 37, 59, 81, 103, 125, 147, 169, 191, 213, 235, 257, 279
12	2, 24, 46, 68, 90, 112, 134, 156, 178

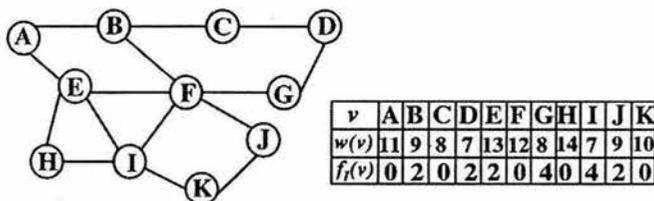
Cuadro 6.7: Asignaciones de $G = (V, A, 5, 3, 2, 1, 1)$.

Para esta gráfica la amplitud inicial es $\sigma(C_8) = 20$; para M , la amplitud cíclica, tenemos $M = 13$; finalmente, como se puede ver en el Cuadro 6.8, la amplitud de f es $\sigma(C_8)=169$.

En este momento cabe preguntar, ¿realmente A' representa una mejora al ALGORITMO A ? en primer lugar con A' podemos considerar un número mayor de restricciones; pero por otra parte en el ALGORITMO A' no es suficiente conocer el número cromático de la gráfica, sino que también es necesario conocer la etiquetación inicial para calcular su amplitud cíclica.

Fijémonos entonces en cómo se comportan estos dos algoritmos con la misma gráfica con restricciones.

v	$f(v)$
1	0, 13, 26, 39, 52, 65, 78, 91, 104, 117, 130, 143, 156, 169
2	3, 16, 29, 42, 55, 68, 81, 94, 107, 120, 133, 146
3	6, 19, 32, 45, 58, 71, 84, 97, 110, 123, 136, 149, 162
4	9, 22, 35, 48, 61, 74, 87, 100, 113, 126, 139, 152, 165
5	1, 14, 27, 40, 53, 66, 79, 92, 105, 118, 131
6	4, 17, 30, 43, 56, 69, 82, 95
7	7, 20, 33, 46, 59, 72, 85, 98, 111
8	10, 23, 36, 49, 62, 75, 88, 101, 114, 127

Cuadro 6.8: Asignaciones de ALGORITMO A' para el ciclo C_8 . G_2 :Figura 6.6: Gráfica $G_2 = (V, A, 3, 2)$.

La gráfica G_2 , en la Figura 6.6 con restricciones $c_0 = 3$ y $c_1 = 2$, con la etiquetación dada, tiene una amplitud cíclica $M = 5$ y número cromático $\chi(G) = 3$. Por tanto, en la ejecución del ALGORITMO A, $s = 6$, y en A' $s = 5$, y dado que A y A' tienen una ejecución similar, es de esperarse que la amplitud de la etiquetación de A' sea menor que la amplitud de A. Revisemos ambas etiquetaciones en el Cuadro 6.9.

De esta manera, si la amplitud cíclica es menor que $\max\{c_0, \chi(G) \cdot c_1\}$, la asignación que produce el ALGORITMO A' tendrá una amplitud menor.

v	A	A'
A	0, 6, 12, 18, 24, 30, 36, 42	0, 5, 10, 15, 20, 25, 30, 35
B	2, 8, 14, 20, 26, 32, 38, 44, 50	2, 7, 12, 17, 22, 27, 32, 37, 42
C	0, 6, 12, 18, 24, 30, 36 42, 48, 54, 60	0, 5, 10, 15, 20, 25, 30 35, 40, 45, 50
D	2, 8, 14, 20, 26, 32	2, 7, 12, 17, 22, 27
E	2, 8, 14, 20, 26, 32, 38	2, 7, 12, 17, 22, 27, 32
F	0, 6, 12, 18, 24, 30, 36, 42, 48, 54, 60, 66, 72, 78	0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65
G	4, 10, 16, 22, 28, 34, 40, 46	4, 9, 14, 19, 24, 29, 34, 39
H	0, 6, 12, 18, 24, 30, 36, 42, 48	0, 5, 10, 15, 20, 25, 30, 35, 40
I	4, 10, 16, 22, 28, 34	4, 9, 14, 19, 24, 29
J	2, 8, 14, 20, 26, 32, 38	2, 7, 12, 17, 22, 27, 32
K	0, 6, 12, 18, 24	0, 5, 10, 15, 20

Cuadro 6.9: Asignación de la gráfica 6.6 con A y A'.

A y A' son algoritmos generales que podemos emplear en cualquier gráfica con restricciones y pesos, pero existen gráficas que por sus características podemos agrupar en familias, como por ejemplo las trayectorias, los ciclos, o las gráficas bipartitas, y por supuesto que es conveniente contar con algoritmos específicos para una familia de gráficas.

6.3. Algoritmo de Asignación para gráficas bipartitas.

De la misma manera que el ALGORITMO A, este algoritmo usa una coloración base de los vértices, pero si un vértice tiene una demanda mayor que cualquiera de sus vecinos, entonces éste toma inicialmente alguno de los canales que están separados $2c_1$, lo que permite diseminar los canales de sus vecinos mientras los últimos canales están c_0 alejados.

ALGORITMO B.

Para gráficas bipartitas que cumplen $c_1 \leq c_0 \leq 2c_1$.

PRECONDICIONES:

$G = (V, A, c_0, c_1)$ una gráfica con restricciones.

w un vector de pesos arbitrario.

$f_1 : V \mapsto \{0, 1\}$ una coloración base de $V(G)$

ASIGNACION.

para cada vértice $v \in V(G)$ hacer

$p(v) = \text{máx}\{w(v) \mid uv \in A \text{ o } u = v\}$.

FASE UNO:

para cada $i \in \{0, 1, \dots, \text{mín}\{w(v), p(v)\} - 1\}$ hacer

$f(v) = f_1(v)c_1 + 2ic_1$

FASE DOS:

Si $w(v) > p(v)$ entonces

para cada $i \in \{0, 1, \dots, w(v) - p(v) - 1\}$ hacer

$f(v) = f_1(v)c_1 + 2p(v)c_1 + ic_0$

Cuadro 6.10: ALGORITMO B.

Para gráficas bipartitas con restricciones c_0 y c_1 el ALGORITMO B nos da una asignación óptima si $c_0 \geq 2c_1^1$.

$G = (V, A, 2, 1)$

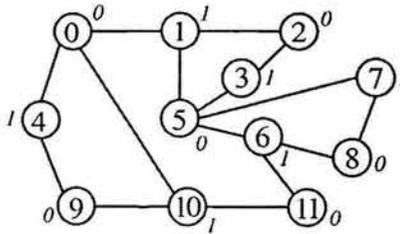


Figura 6.7: Gráfica bipartita.

v	0	1	2	3	4	5	6	7	8	9	10	11
$w(v)$	4	11	4	5	7	3	8	9	3	11	5	8
$f_1(v)$	0	1	0	1	1	0	1	1	0	0	1	0

Cuadro 6.11: Etiquetación inicial para $G=(V,A,2,1)$

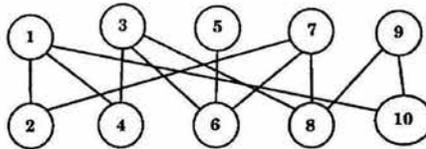


Figura 6.8: Gráfica bipartita 3, $G = (V, A, 4, 2)$.

¹ $\chi(Km, n) = 2$.

V	$f(v)$
0	0, 2, 4, 6
1	1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21
2	0, 2, 4, 6
3	1, 3, 5, 7, 9
4	1, 3, 5, 7, 9, 11, 13
5	0, 2, 4
6	1, 3, 5, 7, 9, 11, 13, 15
7	1, 3, 5, 7, 9, 11, 13, 15, 17
8	0, 2, 4
9	0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20
10	1, 3, 5, 7, 9
11	0, 2, 4, 6, 8, 10, 12, 14

Cuadro 6.12: Asignaciones de la gráfica 6.7 con el ALGORITMO A'.

v	1	2	3	4	5	6	7	8	9	10
$w(v)$	9	7	7	8	8	7	5	5	4	3
$f(v)$	0	1	0	1	0	1	0	1	0	1
$p(v)$	9	9	8	9	8	8	7	7	5	9

Cuadro 6.13: Asignación inicial para la gráfica.

¿Cómo afectan las restricciones de B? Las condiciones que debe cumplir una gráfica bipartita para aplicar B, son específicas, observemos la etiquetación que resulta cuando ignoramos la restricción $c_0 \leq 2c_1$, en el Cuadro 6.14.

V	B
1	2, 6, 10, 14, 18, 22, 26, 30, 34, 38, 42, 46, 50, 54, 58
2	0, 4, 8, 12, 16, 20, 24, 28, 32, 36
3	2, 6, 10, 14, 18, 22, 26, 30
4	0, 4, 8, 12, 16, 20, 24, 28
5	2, 6, 10, 14, 18, 22, 26
6	0, 4, 8, 12, 16, 20, 24, 28
7	2, 6, 10, 14, 18
8	2, 6, 10, 14
9	0, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40
10	2, 6, 10, 14, 18, 22, 26, 30, 34
11	0, 4, 8, 12, 16, 20

Cuadro 6.14: ALGORITMO B con parámetros no recomendados.

Inmediatamente apreciamos que en la asignación hecha, los canales se interfieren, esto es porque en el ALGORITMO B se asume que la restricción se cumple y no se efectúa ningún cálculo con respecto a las restricciones. En primera instancia podríamos decir que B es demasiado específico, aunque en realidad esta condición nos permite tener cotas más justas, ya que para cualquier otra gráficas contamos con los Algoritmos A y A'.

6.4. Algoritmo de Asignación para ciclos impares.

La asignación de canales en ciclos impares fue estudiada por Griggs y Yeh en [6], y parte de dicho estudio se revisó en la sección anterior. Los autores dan la etiquetación para restricciones $c_1, c_2 = 2, 1$ de amplitud 4 y amplitud cíclica 6.

ALGORITMO C.

Para ciclos impares.

PRECONDICIONES:

$G = (V, A, c_0, c_1)$ un ciclo con restricciones, de n nodos, con n impar y $n > 3$.

Los vértices están numerados en el orden del ciclo,

y el *vértice 1* es el nodo de mayor peso, en el ciclo.

w un vector de pesos arbitrario.

ASIGNACIÓN.

$$c_R = 2nc_1/(n-1)$$

$$s = \max\{c_0, c_R\}.$$

Sea $m > 1$ el entero impar más pequeño tal que $s \geq \frac{2m}{m-1}c_1$.

para cada $v_i \in V(G)$ hacer

asignar los canales $b(v_i) + js$ donde

$$j = 0, 1, \dots, w(v_i) - 1.$$

la etiquetación $b: V \mapsto [0, s-1]$ se define como:

Casos

si $1 \leq i \leq m$:

$$b = (i-1)c_1 \pmod{s};$$

si $i > m$ e i es par :

$$b = 0;$$

si $i > m$ e i es impar :

$$b = (m-1)c_1 \pmod{s}.$$

Este primer algoritmo está basado en una etiquetación de la gráfica que satisfice una restricción c_1 . Esta etiquetación tiene una amplitud cíclica $c_R = 2nc_1/(n-1)$ y comienza asignando 0 al primer nodo, después añade c_1 módulo c_R al canal previamente asignado y lo asigna al siguiente nodo en el ciclo. Esto cambia a una asignación alternativa que se usa repetidamente, como en el algoritmo A'.

Nótese que este algoritmo no puede ser implementado, en primera instancia, de una manera distribuida, ya que cada nodo debe conocer todos los pesos, para calcular m y determinar su valor de asignación inicial.

Veamos algunos ejemplos:

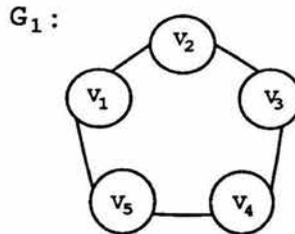


Figura 6.9: $G_1 = C_5 = (V, A, c_0, c_1)$

v	v_1	v_2	v_3	v_4	v_5
$w(v)$	4	2	3	2	3
$f_1(v)$	0	1	0	1	2
$f(v)$	0,3,6,9	1,4	0,3	1,4,7	2,5,8

Cuadro 6.16: Etiquetación para $C_5 = (V, A, 2, 1)$.

Consideremos ahora los siguientes datos:

v	v1	v2	v3	v4	v5
$w(v)$	4	2	3	2	3
$f_1(v)$	0	2	4	1	3

Cuadro 6.17: Etiquetación inicial para $C_5 = (V, A, 3, 2)$

Por las características de esta gráfica, es posible aplicar cualquiera de los algoritmos que se han revisado hasta ahora. En el Cuadro 6.18 mostramos las diferentes etiquetaciones de los algoritmos A, A' y C.

v	A	A'	C
v1	0, 6, 12, 18	0, 5, 10, 15	0, 5, 10, 15
v2	2, 8	2, 7	2, 7
v3	4, 10, 16	4, 9, 14	4, 9, 14
v4	1, 7	1, 6	1, 6
v5	3, 9, 15	3, 8, 13	3, 8, 13

Cuadro 6.18: Asignaciones de la gráfica 6.9 con diferentes algoritmos.

En este caso las asignaciones de A' y de C son una mejora a la que realiza el ALGORITMO A; aunque C es específico para la asignación de canales en los ciclos, el ALGORITMO A', debido a su generalidad, puede ser aplicado para este tipo de gráficas, con una ejecución eficiente y de amplitud mínima. Revisemos un ciclo con orden y demandas mayores en el Cuadro 6.19 y la asignación correspondiente en los Cuadros 6.20 y 6.21.

v	v1	v2	v3	v4	v5	v6	v7	v8	v9
$w(v)$	13	10	10	7	9	11	5	8	9
$f_1(v)$	0	2	0	2	0	2	0	2	4

Cuadro 6.19: Etiquetación inicial para $C_9 = (V, A, 4, 2)$

v	A	A'
v1	0, 8, 16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96	0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60
v2	2, 10, 18, 26, 34, 42, 50, 58, 66, 74	2, 7, 12, 17, 22, 27, 32, 37, 42, 47
v3	0, 8, 16, 24, 32, 40, 48, 56, 64, 72	0, 5, 10, 15, 20, 25, 30, 35, 40, 45
v4	2, 10, 18, 26, 34, 42, 50	2, 7, 12, 17, 22, 27, 32
v5	0, 8, 16, 24, 32, 40, 48, 56, 64	0, 5, 10, 15, 20, 25, 30, 35, 40
v6	2, 10, 18, 26, 34, 42, 50, 58, 66, 74, 82	2, 7, 12, 17, 22, 27, 32, 37, 42, 47, 52
v7	0, 8, 16, 24, 32, 0, 5, 10, 15, 20	0, 5, 10, 15, 20
v8	2, 10, 18, 26, 34, 42, 50, 58	2, 7, 12, 17, 22, 27, 32, 37
v9	4, 12, 20, 28, 36, 44, 52, 60, 68	4, 9, 14, 19, 24, 29, 34, 39, 44

Cuadro 6.20: Asignaciones de la gráfica 6.19 con diferentes algoritmos.

v	C
v1	0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60
v2	2, 7, 12, 17, 22, 27, 32, 37, 42, 47
v3	4, 9, 14, 19, 24, 29, 34, 39, 44, 49
v4	1, 6, 11, 16, 21, 26, 31
v5	3, 8, 13, 18, 23, 28, 33, 38, 43
v6	0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50
v7	3, 8, 13, 18, 23
v8	0, 5, 10, 15, 20, 25, 30, 35
v9	3, 8, 13, 18, 23, 28, 33, 38, 43

Cuadro 6.21: Asignaciones de la gráfica 6.19 con el ALGORITMO C.

6.5. Algoritmo de Asignación para gráficas 3-coloreables.

En este algoritmo el espectro se divide en tres partes, cada una contiene un número determinado de canales, con una separación entre canales consecutivos. La primera parte del espectro consiste de canales alternados de R_1 y G_2 , la segunda parte proviene de B_1 y R_2 y la tercera parte tiene canales alternados de G_1 y B_2 , donde $R_1, R_2, B_1, B_2, G_1, G_2$ son conjuntos de canales determinados por las restricciones.

ALGORITMO E.

Para gráficas 3-coloreables.

PRECONDICIONES:

Sea $G = (V, A, c_0, c_1)$ una gráfica con restricciones.

w un vector de pesos arbitrario.

$f : V \mapsto \{0, 1, 2\}$ una coloración base de G .

FASE CERO.

$s = \text{máx}\{c_1, c_0/2\}$.

$T \geq 3w_{\max}$, T un múltiplo de 6.

Denotemos los colores base 0, 1, 2 como rojo, azul y verde, respectivamente.

FASE UNO.

Los canales Rojos consisten de los conjuntos:

$$\text{I } R_1 = [0, 2s, \dots, (T/3 - 2)s]$$

$$\text{II } R_2 = [(T/3 + 1)s + c_0, (T/3 + 3)s + c_0, \dots, (2T/3 - 1)s + c_0].$$

Los canales Azules consisten de los conjuntos:

$$\text{I } B_1 = [(T/3)s + c_0, (T/3 + 2)s + c_0, \dots, (2T/3 - 2)s + c_0]$$

$$\text{II } B_2 = [(2T/3 + 1)s + 2c_0, (2T/3 + 3)s + 2c_0, \dots, (T - 1)s + 2c_0]$$

Los canales Verdes consisten de los conjuntos:

$$\text{I } G_1 = [(2T/3 + 3)s + 2c_0, (2T/3 + 2)s + 2c_0, \dots, (T - 2)s + 2c_0]$$

$$\text{II } G_2 = [s, 3s, \dots, (T/3 - 1)s]$$

ASIGNACIÓN

para cada $v \in V(G)$ hacer

Asignar $w(v)$ canales de aquellos de su clase de color, donde el primer conjunto es agotado antes de empezar con el segundo conjunto, y siempre se usan los canales con menor valor, dentro de cada conjunto.

Cuadro 6.22: ALGORITMO E.

A diferencia de los algoritmos anteriores, E ocupa dos parámetros, s y T , para calcular los canales en cada conjunto; además para el valor de s consideramos c_1 y $c_0/2$, y en los anteriores era el máximo entre éstas restricciones. Por último para calcular T necesitamos saber todos los pesos en la gráfica, para determinar w_{max} , y

que sea múltiplo de 6 facilita los cálculos, dado que la gráfica y los colores que tiene asignados son múltiplos de 3. Revisemos un ejemplo: la gráfica G en la Figura 6.10 es 3-coloreable, y digamos que tiene restricciones $c_0 = 3$ y $c_1 = 2$.

G:

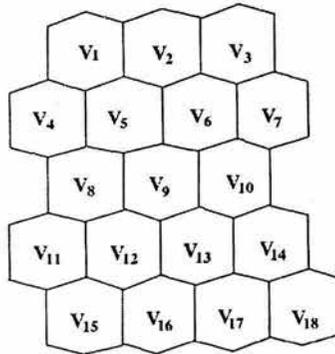


Figura 6.10: Gráfica $G = (V, A, 3, 2)$.

Las demandas para cada vértice y la asignación inicial se dan en el siguiente Cuadro:

v	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9
$w(v)$	5	8	12	6	13	12	11	7	9
$f_1(v)$	0	1	2	1	2	0	1	0	1
v	v_{10}	v_{11}	v_{12}	v_{13}	v_{14}	v_{15}	v_{16}	v_{17}	v_{18}
$w(v)$	13	8	4	12	10	3	2	6	5
$f_1(v)$	2	1	2	0	1	0	1	2	0

Cuadro 6.23: Asignación inicial, usada en el ALGORITMO E.

Tenemos que $c_0 = 3$, $c_1 = 2$ y $w_{max} = 13$, entonces $3 \cdot 13 = 39$, luego $T = 42$ y $s = 2$. Aplicando la FASE UNO:

$$R_1 = \{0, 2 \cdot 2, \dots, (42/3 - 2) \cdot 2\}.$$

$$R_2 = \{(42/3 + 1) \cdot 2 + 3, (42/3 + 3) \cdot 2 + 3, \dots, (2 \cdot 42/3 + 1) \cdot 2 + 3\}.$$

$$B_1 = \{(42/3) \cdot 2 + 3, (42/3 + 2) \cdot 2 + 3, \dots, (42 - 1) \cdot 2 + 2 \cdot 3\}.$$

$$B_2 = \{(2 \cdot 42/3 + 1) \cdot 2 + 2 \cdot 3, \}.$$

$$G_1 = \{(42/3) \cdot 2\}.$$

$$G_2 = \{2, 3 \cdot 2, \dots, (42/3 - 1) \cdot 2\}.$$

El Cuadro siguiente muestra explícitamente los conjuntos obtenidos de la FASE UNO:

R_1	0, 4, 8, 12, 16, 20, 24
R_2	33, 37, 41, 45, 49, 53, 57
B_1	31, 35, 39, 43, 47, 51, 55
B_2	64, 68, 72, 76, 80, 84, 88
G_1	62, 66, 70, 74, 78, 82, 86
G_2	2, 6, 10, 14, 18, 22, 26

Finalmente, realizamos la asignación; obteniendo el Cuadro 6.24.

Revisando con detalle los conjuntos obtenidos observamos una característica importante. Denotemos con números en negritas los canales rojos, en itálicas los canales azules y dejemos sin cambio los canales verdes. Observemos que los canales tienen una separación de 4 entre cada uno; por otra parte, si vemos el conjunto completo de canales asignados, podemos apreciar que están separados en, al menos, 2 unidades, es decir tienen una distancia c_1 entre ellos y además quedan alternados los canales de cada color, para nuestro ejemplo tenemos que:

0, 2, **4**, *6*, 8, 10, **12**, 14, **16**, 18, **20**, 22, **24**, *31*, **33**, *35*, **37**, *39*, **41**, *43*, **45**, *47*, **49**,
51, **53**, *55*, **57**, *62*, **64**, 66, *68*, 70, *72*, 74, *76*, 78, *80*, 82, *84*, 86, *88*.

De ésta manera es fácil ver que, aunque aparentemente en cada par de conjuntos los canales entre el primero y segundo tienen una distancia considerable, en realidad estos canales no se eliminan, la mayoría es elemento en alguna otra clase de color.

v	$f(v)$
v_1	0,4,8,12,16
v_2	31,35,39,43,47,51,55,64
v_3	2,6,10,14,18,62,66,70,74,78,82,86
v_4	31,35,39,43,47,51
v_5	2,6,10,14,18,22,62,66,70,74,78,82,86
v_6	0,4,8,12,16,20,24,33,37,41,45,49
v_7	31,35,39,43,47,51,55,64,68,72,76
v_8	0,4,8,12,16,20,24
v_9	31,35,39,43,47,51,55,64
v_{10}	2,6,10,14,18,22,62,66,70,74,78,82,86
v_{11}	31,35,39,43,47,51,55,64
v_{12}	62,66
v_{13}	0,4,8,12,16,20,24,33,37,41,45,49
v_{14}	31,35,39,43,47,51,55,64,68
v_{15}	0,4,8
v_{16}	31,35
v_{17}	62,66,70,74,78,82
v_{18}	0,4,8,12,16

Cuadro 6.24: Asignaciones con el ALGORITMO E.

Y dado que primero agotamos el primer conjunto y el parámetro T está calculado en base a las restricciones, podemos garantizar que en cada conjunto hay un número de colores suficiente para ser asignado y que la amplitud de la asignación es justa.

Capítulo 7

Algoritmos distribuidos para gráficas Hexagonales

Una buena asignación de canales es esencial para la transmisión y recepción de voz, datos y otras redes de comunicación celular basados en el Acceso Múltiple por División de frecuencias, *FDMA* por *Frequency Division Multiple Access*.

La mayoría de los estudios se han concentrado en el problema estático de encontrar una asignación de canales global para una cierta red y parámetros estáticos predeterminados. Aquí se considera una red celular como un sistema dinámico, donde las decisiones de asignación de canales se hacen localmente, es decir, no se planean globalmente, y además deben ser continuamente actualizadas para ajustar parámetros cambiables. Una red celular se trata como una red distribuida donde cada base es un servidor y los algoritmos de asignación, aquí revisados, son tratados como si se ejecutaran de manera simultánea en cada base.

Asimismo los algoritmos pueden ser ajustados para el problema de asignación en línea; la demanda para canales está sujeta a cambiar y los servidores deben adaptarse a estos cambios, haciendo modificaciones mínimas a la asignación existente.

En este trabajo, los algoritmos para la asignación de canales distribuidos se dan para redes celulares con un trazo hexagonal regular. Las restricciones de interferencia

también se modelan mediante una gráfica con restricciones y pesos (G, \mathbf{w}) (Definición 26), donde $G = (V, A)$ es una gráfica; c_0, \dots, c_k es el conjunto de restricciones que imponen la separación que debe existir en el espectro de radio y \mathbf{w} es el vector de pesos positivos para los vértices de G .

Algoritmos para la asignación de canales en línea (*online*), son descritos, entre otros en [13, 22, 7]. Janssen y colaboradores en [29], estudian asignaciones para asignar canales de manera distribuida para redes celulares hexagonales con la única restricción de que canales asignados a la misma celda o a celdas adyacentes deben ser distintos.

7.1. Gráficas Hexagonales.

Una clase de gráficas con particular importancia en el contexto de la asignación de canales son las gráficas hexagonales. Una **gráfica hexagonal** es una subgráfica inducida de una malla triangular. La malla triangular es considerada como una gráfica cuyos vértices son los puntos de la malla y las aristas existen precisamente entre puntos vecinos.

Las gráficas hexagonales son nombradas así porque pueden usarse para representar una red celular con celdas hexagonales regulares. Ya que el decremento de una señal de radio es proporcional a la distancia desde donde se transmite, el área de cobertura tendrá una forma circular. Las celdas hexagonales tienen la ventaja de aproximarse a esta forma y son fácilmente armables. Aunque en la práctica las limitaciones impuestas por el terreno fuerzan a la red celular a desviarse del ideal. La Figura 7.1 muestra un ejemplo de una gráfica hexagonal y su red celular subyacente.

Si las señales son transmitidas por frecuencias muy cercanas, éstas generalmente tendrán niveles de interferencia mayores que en otro caso. De aquí el requerimiento común en las redes celulares: canales asignados a celdas que geográficamente están

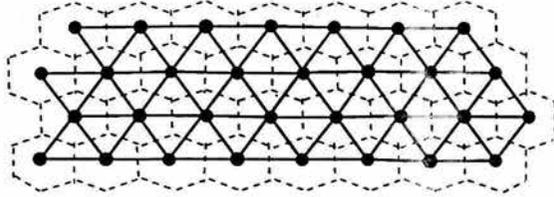


Figura 7.1: Ejemplo de gráfica hexagonal.

cerca deben de estar suficientemente alejadas en el espectro de radio.

La Figura 7.2 muestra una asignación aritmética tomando como restricciones $c_0 = 2$, $c_1 = 5 \pmod{9}$.

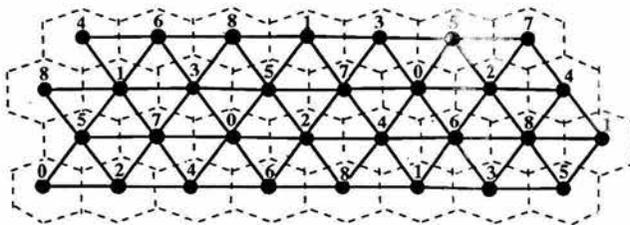


Figura 7.2: Asignación Aritmética en una gráfica hexagonal.

Asumimos que la malla es generada por los vectores $\mathbf{x} = (1, 0)$ y $\mathbf{y} = (\frac{1}{2}, \frac{\sqrt{3}}{2})$ y se usan las coordenadas en la malla para identificar los vértices de las gráficas. Más exactamente, el vértice (i, j) denotará el vértice correspondiente al punto de la malla $i\mathbf{x} + j\mathbf{y}$.

7.2. Algoritmos en línea distribuidos.

Los algoritmos dados son distribuidos y determinísticos. Cada vértice es considerado como un servidor independiente y el Algoritmo está ejecutándose simultáneamente en estos servidores. Cada servidor calcula local y determinísticamente su propia asignación de canal, basado sólo en una pequeña cantidad de información preprocesada, no dependiente de los pesos, e información local del estado de la red. Más exactamente, un servidor en un nodo sabe el peso de dicho nodo y puede conseguir información acerca del peso de los nodos que están a menos de una cierta distancia de él.

Definición 32 *Algoritmo de Asignación k – local.* Un algoritmo de asignación de canal distribuido es k – local si cada servidor calcula su asignación conociendo sólo la información preprocesada y el peso de todos los vértices a distancia k o menos.

El concepto de localidad toma un sentido especial en el contexto de algoritmos de asignación de canal en línea. Ahora la entrada del algoritmo es una gráfica con restricciones y una secuencia w_t de vectores de peso. Con cada ejemplar de tiempo t , cada vértice v es presentado independientemente con su nuevo peso $w_t(v)$. Si $w_t(v) < w_{t-1}(v)$ entonces ciertas llamadas en la celda correspondiente son omitidas. Si $w_t(v) > w_{t-1}(v)$, entonces deben ser asignados de manera que las restricciones de interferencia entre los nuevos y viejos canales asignados se cumplan.

7.3. Algoritmos.

A continuación describimos dos algoritmos de asignación de canal estáticos en línea distribuidos para una gráfica hexagonal con restricciones $G = (V, A, a, a, b)$ donde $a \geq b > 0$. El primer algoritmo está basado en una asignación inicial la cual será óptima si todos los pesos son iguales, combinado con una fase de préstamo para considerar las diferencias de peso. El segundo algoritmo tiene una primera fase de asignación para las restricciones $a = 1, b = 0$, combinada con una segunda parte para encontrar una asignación con parámetros generales a y b . Su desempeño y localidad dependerán de la fase que se use.

7.3.1. Algoritmo de Préstamo Aritmético (AB).

Primero describimos un algoritmo válido cuando $a \geq 2b$. Este algoritmo está basado en una asignación inicial f de los vértices derivados de las coordenadas en la malla triangular. Precisamente:

$$f(i, j) = ia + j(3a + b) \pmod{N},$$

donde $N = 5a + 3b$. La etiquetación se ejemplifica en la Figura 7.3. Etiquetaciones de este tipo fueron tratadas por Jan van den Heuvel en [4].

Nótese que f tiene la propiedad de $N \geq a, a \leq f(i, j) \leq N - a$ para todos los vecinos de $(0,0)$ y $b \leq f(i, j) \leq N - b$ para todos los vértices a distancia 2 de $(0,0)$. Ya que $f(0, 2) = a - b$ esto sólo es cierto si $a \geq b$. Por tanto, f genera una asignación de canales para G , se designa a cada vértice (i, j) solamente canales del conjunto $\{f(i, j) + kN \mid k \in \mathbb{N}\}$. Ahora para cualquier par de canales γ_1, γ_2 asignados a los vértices (i, j) e (i', j') a distancia 2 tenemos que $|\gamma_1 - \gamma_2| \equiv f(i', j') - f(i, j) \equiv f(i' - i, j' - j) \pmod{N}$. Ya que $(i' - i, j' - j)$ tiene en la gráfica una distancia de 2 a $(0,0)$, se sigue que $|\gamma_1 + \gamma_2| \geq b$. Un argumento similar es válido para vértices a distancia 1.

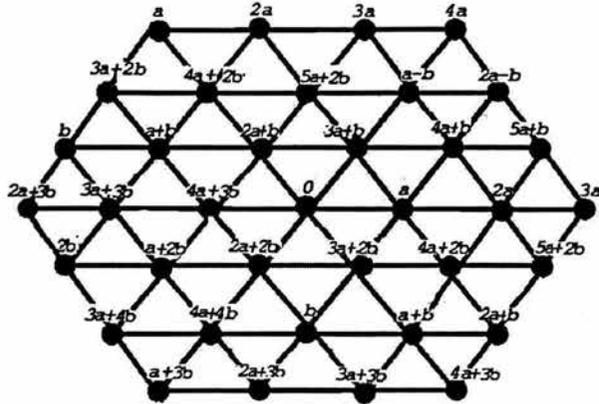


Figura 7.3: La Asignación Aritmética usada en el Algoritmo AB

También en el caso que $b \leq f(i, j) \leq N - b$ incluso para vértices (i, j) a distancia 3 de $(0,0)$. Así cualquier asignación de canales derivada de f tiene la propiedad de que vértices a distancia 3 también tienen separación de al menos b .

Como información local cada vértice $v = (i, j)$ sabe su valor bajo $f(i, j)$, es capaz de identificar a sus vecinos y su posición con respecto a sí mismo, y recibir información acerca de su peso. Específicamente, v es capaz de identificar los vecinos $(i + 1, j)$ e $(i + 1, j - 1)$ y calcular el peso máximo en un clan entre sus vecinos. Es decir, v puede calcular $T(v)$, donde:

$$T(v) = \max \left\{ \sum_{u \in K} w(u) \mid K \text{ es un clan, } d(u, v) \leq 1 \quad \forall u \in K \right\}$$

El ALGORITMO AB comienza asignando canales de acuerdo a f , seguido por 2 "fases de pedir prestado", donde los vértices con alta demanda piden prestados los canales no usados de sus vecinos. La asignación de canales a cada vértice $v(i, j)$ es hecha en 3 fases:

- Fase 1. El vértice v recibe los canales $f(i, j) + kN$, $0 \leq k \leq \min\{w(v), T(v)/3\}$.
- Fase 2. Si v tiene un peso mayor que $T(v)/3$, es decir $w(v) = T(v)/3 + \alpha$, $\alpha > 0$, entonces v trata de pedir prestados canales a su vecino $x = (i + 1, j)$. Estrictamente, si $w(v) < T(v)/3 + \alpha$, entonces v recibe los canales $f(i + 1, j) + kN$, $w(v) \leq k < \min\{w(v) + \alpha, T(v)/3\}$. Sea $\beta = \max\{0, T(v)/3 - w(x)\}$ el número máximo de canales que v recibe en esta fase.
- Fase 3. Si v aún tiene demanda no cumplida, después de la última fase, es decir, si $\alpha > \beta$, entonces v pide prestados los canales sobrantes de sus vecinos $f(i + 1, j - 1) + kN$, $T(v)/3 - \alpha + \beta \leq k < T(v)/3$.

Observemos la asignación aritmética para una gráfica hexagonal; veamos la asignación en la Figura 7.4, considerando $a = 2$ y $b = 1$.

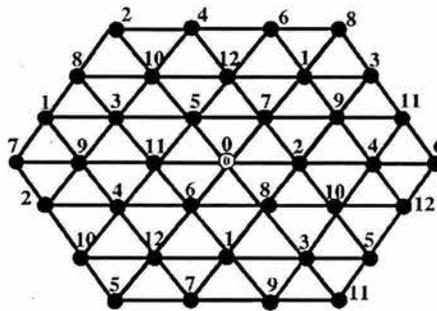


Figura 7.4: Ejemplo de una asignación donde $a = 2$ y $b = 1$.

En la Figura 7.5, representamos los pesos de la gráfica anterior.. Revisemos cómo se hace la asignación para un vértice de G , por ejemplo el vértice $(0,0)$ de la gráfica 7.4. En la Figura 7.6 podemos ver al vértice 0 y sus vecinos a distancia 1.

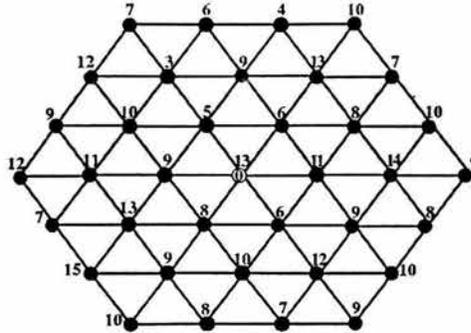


Figura 7.5: Gráfica $G = (V, A, 3, 2)$ con pesos.

Para éste vértice tenemos que: $T(0) = 13 + 11 + 6 = 30$, por otra parte $N = 5a + 3b = 13$. Por tanto la asignación de 0 queda como sigue:

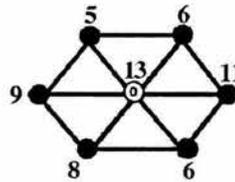


Figura 7.6: Vértice $(0,0)$ y sus vecinos.

- Fase 1: En total son asignados $\min\{w(0), T(0)/3\}$, es decir 10 canales: 0, 13, 26, 39, 52, 65, 78, 101, 114 y 127.
- Fase 2: $\alpha = 3, \beta = 0$, por tanto el vértice no recibe canales en esta fase.
- Fase 3: En esta fase $f(i + 1, j - 1) = 6$, y $k = 7, 8, 9$, finalmente se asignan los canales: 97, 110 y 123.

Con lo que se cumple la demanda.

Capítulo 8

Conclusiones.

Algunos problemas que comprenden aplicaciones de la vida diaria, como la repartición del correo postal, empaquetamiento de mercancía, el traslado de un lugar a otro, son problemas para los cuales aún o se conoce algún algoritmo que obtenga la solución óptima en tiempo polinomial en el tamaño de la entrada. Además, regularmente, para estos problemas se requiere una solución inmediata, que si bien sabemos no será la solución óptima, debe ser aceptable.

El problema de Asignación de Canales consiste en optimizar el uso del espectro bajo la proximidad inducida por las restricciones de frecuencia. El modelo matemático utilizando teoría de gráficas, visto en este trabajo, induce a una generalización del problema que consiste en encontrar el número cromático de una gráfica, para el cual, es bien sabido, no se conoce un algoritmo eficiente que lo resuelva. Por este motivo es necesario proponer modelos matemáticos, crear algoritmos de aproximación y heurísticas, que permitan encontrar soluciones. Una vez hecho esto, se evalúa que tan cercana es la solución obtenida a la óptima esperada o calculada teóricamente, finalmente puede plantearse la posibilidad de mejorar las estrategias para establecer nuevos modelos y generar nuevos métodos de solución.

Actualmente, la aplicación más importante para la asignación de canales es la telefonía celular. Esta es una industria que ha crecido aceleradamente en la última década; su objetivo, como en todo negocio, es invertir poco, ganar mucho y brindar un

excelente servicio. Para lograrlo se requieren modelos matemáticos con las condiciones suficientes para crear algoritmos que encuentren muy buenas soluciones. El concepto de reuso de canal fue un gran avance para solucionar el problema de escasez de espectro y para desarrollar diversas estrategias y modelos para la asignación de canales.

Los algoritmos aquí presentados muestran diversas estrategias para atacar el problema de asignación de canales. Tales algoritmos generan muy buenas soluciones, su desempeño computacional es polinomial y se tienen cotas sobre las soluciones.

Apéndices.

Apéndice A

Algoritmos.

La palabra algoritmo es derivada del nombre del matemático persa Mohammed al Khwarizmî quien vivió durante el siglo IX y es quien recibe el crédito de haber implementado reglas para realizar sumas, restas, multiplicaciones y divisiones de números decimales paso a paso. Al pasarse el nombre al latín éste se convirtió en *Algorismus* y de ahí en castellano a *Algoritmo* [33, 32].

Los algoritmos son una parte central de todas las áreas de las ciencias de la computación y siendo más justos podemos decir que forma parte relevante de casi todas las ciencias, de los negocios y de la tecnología.

La naturaleza de un algoritmo se logra apreciar generalmente en todas aquellas disciplinas que se benefician del uso de computadoras. La computadora no solamente es una máquina que puede realizar procesos para darnos resultados, sin que tengamos la noción exacta de las operaciones que realiza para llegar a ellos.

Con la computadora, además de lo anterior, también podemos diseñar soluciones a la medida de problemas específicos. Sobre todo, si estos involucran operaciones matemáticas repetitivas, o que requieren del manejo de un volumen muy grande de datos.

El diseño de soluciones a la medida de nuestros problemas requiere, como en otras

disciplinas, una metodología que nos enseñe de manera gradual la forma de llegar a las soluciones. Una forma de obtener soluciones, a través de las computadoras son los programas, que no son más que una serie de operaciones que realiza la computadora para llegar a un resultado, con un grupo de datos específicos.

Lo anterior nos lleva al razonamiento de que un programa nos sirve para solucionar un problema específico. Para poder realizar programas, además de conocer la metodología mencionada, también debemos de conocer, de manera específica las funciones que puede realizar la computadora y las formas en que se pueden manejar los elementos que hay en la misma.

Definición 33 *Algoritmo*. Un **algoritmo** es una secuencia finita de pasos, que resuelve un problema en un tiempo finito (que termina en un número finito de operaciones).

Un algoritmo tiene las siguientes características:

- I Trabaja a partir de datos, aunque ocasionalmente podría no recibir entradas.
- II Produce como salida un resultado que corresponde a la solución del problema; o bien termina indicando que no existe tal.
- III Secuencia Finita de pasos. El algoritmo define una secuencia de pasos cuya ejecución transforma a la entrada en la salida. Cada uno de estos pasos debe estar bien definido.
- IV Correctez. El algoritmo debe terminar y garantizar la solución para cualquier ejemplar del problema.

El propósito del análisis de Algoritmos es predecir el comportamiento de los algoritmos sin implantarlos en máquina alguna [12]. Además es mucho más conveniente tener medidas sencillas para la eficiencia de un algoritmo que implantarlo y probar su eficiencia cada vez con parámetros específicos para sistemas diferentes.

Tradicionalmente, el tiempo de ejecución de un algoritmo depende del tiempo de ejecución y tamaño de memoria utilizado.

A.1. Análisis de Algoritmos

El tiempo necesario para ejecutar un algoritmo resulta ser una función que depende generalmente de la cantidad de datos a procesar; mayor cantidad de datos significa más tiempo de ejecución. El valor exacto de esta función depende de varios factores, tales como la velocidad de la máquina. Así el resultado del análisis debe indicar cuánto tarda el algoritmo en cuestión, en tiempo de ejecución, para una entrada particular.

Dado que el número de entradas potenciales es enorme y que seguramente dos algoritmos tendrán comportamiento diferente para entradas distintas, debemos considerar una métrica para la entrada, como su tamaño, y entonces hacer un análisis referente a este tamaño.

Por otro lado, los algoritmos no se comportan de manera similar para todas las entradas del mismo tamaño, pero es imposible analizar todas las entradas del mismo tamaño para todos los algoritmos. En lugar de esto, podemos comparar el desempeño de algoritmos diferentes que resuelvan el mismo problema. De esta manera, ¿cómo elegimos un indicador?

Elegir la mejor entrada usualmente no es muy representativo, porque representa una solución trivial. El caso esperado es una buena elección, pero algunas veces será muy difícil obtener una medida eficiente, pues la esperanza del tiempo de ejecución dependerá de parámetros diferentes o de la forma como están organizados los datos. Finalmente, el cálculo del tiempo de ejecución, usando la peor entrada como indicador es muy usual, aunque represente un análisis pesimista.

Para un programa fijo, ejecutándose en una computadora, podemos dibujar la gráfica que representa la función del tiempo de ejecución. La Figura A.1 muestra una gráfica sobre cuatro programas, donde las curvas representan 4 funciones típicas en el análisis de algoritmos: *lineal*, $n \log n$, *cuadrática* y *cúbica*. El tamaño de la entrada n varía de 1 a 100 elementos y los tiempos de ejecución asociados varían de 0 a 10 milisegundos. La Figura A.2 muestra los tiempos de ejecución para los mismos programas, pero con tamaños de entrada mayores.

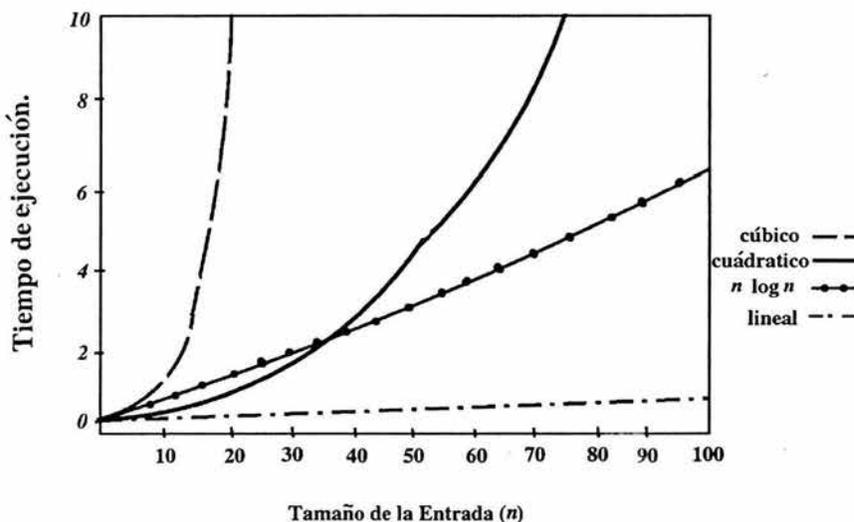


Figura A.1: Tiempos de ejecución para tamaños de entrada pequeños.

Para valores pequeños de n , por ejemplo aquellos menores a 35, en la Figura A.1 mostramos que hay puntos para los cuales una curva es inicialmente mejor que otra, aunque esto deja de ser cierto cuando la entrada n se hace “suficientemente” grande. Para tamaños de entradas pequeños, es difícil comparar las funciones. Por ejemplo, consideremos la función $f(n) = n + 2500$, es mayor que n^2 cuando n es menor que 50, pero a partir de este punto la función lineal será siempre menor que la función cuadrática y la constante pierde importancia.

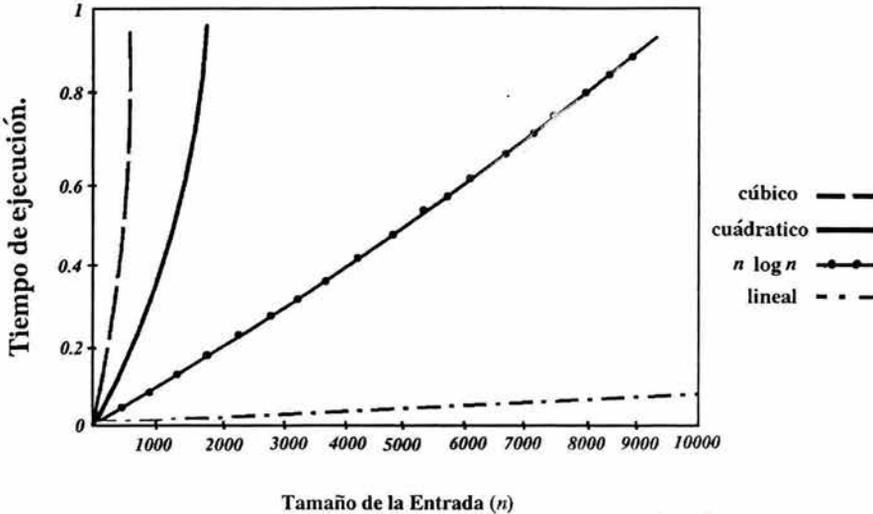


Figura A.2: Tiempos de ejecución para tamaños de entrada mayores.

Otro punto importante es que para tamaños pequeños de entrada los tiempos de ejecución son insignificantes y comúnmente no tenemos que preocuparnos por ellos.

La Figura A.2 muestra mayores diferencias entre las curvas de la primera figura, cuando el tamaño de la entrada es grande. Un algoritmo resuelve un problema de tamaño 10,000 en una fracción de segundo mientras que el algoritmo $n \log n$ utiliza aproximadamente diez veces este tiempo.

La característica más notoria de estas curvas es que los algoritmos cuadráticos y cúbicos no pueden competir con los restantes para tamaños de entrada razonablemente grandes. El cuadro A.1 ordena de manera creciente distintas funciones que describen comúnmente el tiempo de ejecución de los algoritmos.

Función	Nombre
c	Constante.
$\log n$	Logarítmica.
n	Lineal.
$n \log(n)$	$n \log n$.
n^2	Cuadrática.
n^3	Cúbica.
2^n	Exponencial.

Cuadro A.1: Funciones de índice de crecimiento.

A.1.1. Notación asintótica.

Una vez que revisamos las ideas básicas del análisis de algoritmos, podemos dar un enfoque más formal para la notación de las diferentes funciones de crecimiento.

Definición 34 *Notación O* . Decimos que $f(n)$ es de orden de $g(n)$, denotado $f(n) \in O(g(n))$, si existen $c > 0$ y n_0 tales que:

$$f(n) \leq c \cdot g(n) \quad \forall n \geq n_0.$$

La notación O representa una cota superior para el tiempo de ejecución del algoritmo, garantizando que no se va a tardar más, para n suficientemente grande. Esta notación es similar a la relación menor o igual, con respecto al crecimiento de funciones, es decir, que f no crece más rápido que g .

Definición 35 *Notación Ω* . $f(n) \in \Omega(g(n))$ si existen $c > 0$ y n_0

$$f(n) \geq c \cdot g(n) \quad \forall n \geq n_0.$$

Esta notación es similar a la relación mayor o igual, es decir, una función no crece más lento que otra.

Definición 36 *Notación Θ* . $f(n) \in \Theta(g(n))$ si y sólo si:

$$f(n) \in O(g(n)) \text{ y } f(n) \in \Omega(g(n))$$

Utilizamos esta notación para indicar que las dos funciones tienen un índice de crecimiento similar. Θ representa una categoría de orden.

Definición 37 *Notación o* . Decimos que $f(n)$ es *o* pequeña de $g(n)$, denotado $f(n) \in o(g(n))$, si para toda $c > 0$ existe $n_0 > 0$ tales que:

$$0 \leq f(n) < c \cdot g(n) \quad \forall n \geq n_0$$

La notación *o* es similar al menor estricto, es decir, una función crece estrictamente más lento que otra.

Definición 38 *Notación ω* . $f(n) \in \omega(g(n))$ si para toda $c > 0$ existe $n_0 > 0$ tales que:

$$0 \leq c \cdot g(n) < f(n) \quad \forall n \geq n_0$$

La notación ω es similar al mayor estricto, es decir, una función crece estrictamente más rápido que otra.

Generalmente utilizaremos la notación O para representar el índice de crecimiento de una función, así el tiempo de ejecución de un algoritmo cuadrático se describe como $O(n^2)$. La notación O nos permite establecer un orden relativo entre funciones, comparando los términos dominantes. También hay que notar que lo que se busca siempre es dar cotas lo más ajustadas posibles.

El Cuadro A.2 muestra algunos ejemplos de órdenes de complejidad y el nombre que se les asocia.

Función	Nombre
$O(1)$	Constante.
$O(\log n)$	Logarítmico.
$O(n)$	Lineal.
$O(n \log n)$	$n \log(n)$.
$O(n^2)$	Cuadrática.
$O(n^3)$	Cúbica.
$O(2^n)$	Exponencial.

Cuadro A.2: Órdenes de complejidad.

Los distintos órdenes de complejidad, en las diferentes notaciones, particionan a las funciones en clases de equivalencia, una por cada orden de complejidad. Listamos a continuación varias propiedades que preservan estas clases:

1. $g(n) \in O(f(n)) \iff f(n) \in \Omega(g(n))$
2. $k \in \mathbb{N}$, $f(n) \in O(g(n)) \iff k \cdot f(n) \in O(g(n))$
3. $g(n) \in \Theta(f(n)) \iff f(n) \in \Theta(g(n))$
4. Si $b > 1$, $a > 1$, $\log_a n \in \Theta(\log_b n)$
5. Si $0 < a < b$, $a^n \in o(b^n)$
6. $\forall 0 < a$, $a^n \in o(n!)$

A.2. Problemas P, NP y NP-Completos.

Este material está basado en el libro *Compared to what?: an introduction to the analysis of algorithms*, de Gregory J. E. Rawlins [19].

Las Ciencias de la Computación tratan problemas de *complejidad computacional*, es decir, estudian los costos de cómputo necesarios para resolver un problema, hablando en términos de recursos computacionales, fundamentalmente memoria, espacio de almacenamiento y tiempo de cómputo. La teoría de complejidad se restringe a trabajar con problemas de decisión, para los cuales su solución corresponde a una respuesta SÍ o NO.

Cuando nos enfrentamos a un problema concreto, habrá una serie de algoritmos aplicables. Se suele decir que el orden de complejidad de un problema es el del mejor algoritmo que se conozca para resolverlo. Así se clasifican los problemas, y se realizan los estudios sobre algoritmos que se aplican en el mundo real. Estos estudios nos indican que existen problemas muy difíciles y problemas que no son computacionalmente tratables.

¿Cómo identificamos los problemas que se pueden resolver? Empecemos por definir el concepto *problema*.

Definición 39 *Problema.* Es una cuestión a resolver, que generalmente posee parámetros o entradas.

Después de definir un problema, debemos decidir cómo tratarlos para resolverlos. El siguiente es un plan general que podemos seguir:

1. Primero construimos o elegimos un **modelo** abstracto, de manera que podamos decidir entre diferentes soluciones. Cada solución debe tener alguna medida para evaluar la dificultad de implementar dicha solución. Por ejemplo, la cantidad de memoria, el acceso a disco, el número de comparaciones o el de asignaciones. A este modelo se le conoce como modelo de cómputo y no es otra cosa que un conjunto de suposiciones que se asumen acerca de una máquina (virtual o real) sobre la cual será ejecutado el algoritmo.

2. Después diseñamos un algoritmo que resuelva el problema, teniendo en cuenta el modelo elegido. Este algoritmo es una manera de resolver el problema y debe ser suficientemente claro para que pueda efectuarse en una computadora.
3. Finalmente, comparamos las cotas máxima y mínima para comprobar si la solución es buena, para el problema particular. De no ser así, tratamos de rediseñar el algoritmo para obtener mejores cotas. Si esto no funciona, entonces rediseñamos el modelo.

A.2.1. Clasificación de problemas.

Los problemas computacionales se pueden clasificar según los requerimientos y según la complejidad.

En términos de los requerimientos, existen seis problemas computacionales:

1. Búsqueda de datos. Encontrar una X en la entrada que satisface la propiedad Y .
2. Estructura. Transformar la entrada para satisfacer la propiedad Y . Por ejemplo, ordenar un conjunto de números.
3. Construcción. Construir una X que satisface la propiedad Y . Por ejemplo, una gráfica con restricciones.
4. Optimización. Encontrar la mejor X que satisface la propiedad Y . Por ejemplo, encontrar la ruta más rápida en los canales de comunicación.
5. Decisión. Decidir cuándo la entrada satisface la propiedad Y .
6. Adaptivos. Mantener la propiedad Y todo el tiempo. Por ejemplo mantener un servidor de aplicaciones en determinadas condiciones físicas.

También podemos clasificar los problemas según la complejidad. Existen cuatro categorías de problemas difíciles:

1. Conceptualmente difícil. No se tienen algoritmos para resolver estos problemas, porque éstos no se entienden lo suficiente para diseñar un modelo o un algoritmo. Muchos de estos problemas se tratan en el área de Inteligencia Artificial.
2. Analíticamente difícil. Se tiene un algoritmo para resolver este problema, pero no se sabe como analizarlo, es decir, no se sabe cuánto tiempo tomará para resolver cada ejemplar.
3. Computacionalmente difíciles. Existe un algoritmo y su análisis, pero tal análisis sugiere que para ejemplares de problemas relativamente pequeños, tardaría varios años en resolverse.
Estos problemas se dividen en dos grupos: problemas que se sabe son computacionalmente difíciles y problemas que se sospecha lo son.
4. Computacionalmente sin solución. No se tiene un algoritmo para resolver este problema porque tal algoritmo no se puede diseñar.

Todavía quedan por resolver algunas preguntas: ¿Hay problemas mucho más difíciles que otros? De ser así, ¿cuáles? ¿Qué tan difíciles son estos problemas? Finalmente, si resultan ser mucho más difíciles, ¿qué podemos hacer?

Primero consideremos que, actualmente, nombramos a un problema como factible o eficiente si tiene una solución con costo polinomial, pero ¿por qué polinomial? Veamos porque:

- Los Polinomios son cerrados bajo composición y suma.

Si tenemos 2 algoritmos factibles podemos usar uno como subrutina del otro y

obtendremos un algoritmo factible, esto significa que es cerrado bajo composición. Por otra parte, si ejecutamos uno después del otro, también obtenemos un algoritmo factible, lo que indica que es cerrado bajo suma.

- Todas las computadoras de secuencia digital están relacionadas polinomialmente. Si en una máquina podemos resolver un problema en un tiempo $f(n)$, podremos resolverlo en el mismo tiempo polinomial $f(n)$ en otra máquina. Si un problema tiene una solución polinomial en cualquier computadora digital actual, entonces este problema tendrá una solución polinomial en cualquier otra computadora digital. Esto hace más sencillo nombrar a un problema como polinomial, independientemente de la máquina en el que se ejecute.

- Por lo general, un algoritmo polinomial hará una cantidad considerable de trabajo. Si un algoritmo es exponencial, entonces será factible sólo para entradas pequeñas.

A.2.2. Problemas NP.

En la sección anterior revisamos por qué preferimos tener algoritmos polinomiales. A continuación formalizamos la idea de que un problema con solución polinomial es “fácil” y uno donde cada solución crece más rápido que cualquiera polinomial es “difícil”.

Definición 40 *Polinomialmente transformable.* Decimos que un problema P_1 es **polinomialmente transformable** a un problema P_2 , si podemos transformar cualquier entrada de P_1 en un ejemplar de P_2 en un tiempo polinomial.

De esta manera podemos resolver cualquier ejemplar de P_1 transformándolo, en tiempo polinomial, en un ejemplar de P_2 , y entonces resolvemos el ejemplar de P_2 . De

esta manera podemos resolver cualquier ejemplar de P_1 , así que si P_2 es fácil, entonces P_1 es fácil. Pero si P_2 es difícil, no podemos decir que P_1 es difícil, ya que podría existir otra manera de resolver P_1 aunque sí podemos asegurar que P_1 no es más difícil que P_2 .

Para demostrar que P_1 es polinomialmente transformable en P_2 tenemos que encontrar un algoritmo de tiempo polinomial que transforma cualquier ejemplar de P_1 en un ejemplar de P_2 . Si se puede hacer esta transformación, entonces éstos dos problemas son **polinomialmente equivalentes**. Y esto se cumple, aún sin considerar si son fáciles o difíciles.

Se dice que los problemas que se resuelven en tiempo polinomial están en la clase P . Diremos que un problema es computacionalmente difícil si no está en P , es decir, si este problema no tiene una solución de tiempo polinomial.

Existe una clase que contiene la clase P . Estos problemas son *No-determinísticos de tiempo polinomial*. Algunos de estos problemas pueden caracterizarse por el hecho de que puede aplicarse un algoritmo polinomial para comprobar si una posible solución es válida o no. Esta característica lleva a un método de resolución no determinista, que consiste en aplicar heurísticas para obtener soluciones que se van aceptando o rechazando.

La clase de estos problemas se le llama NP . La clase P es un subconjunto de NP , ya que un algoritmo determinístico en P puede plantearse como un algoritmo no determinístico que cuenta con una solución polinomial.

Definición 41 *Problema NP-difícil.* Un problema X es llamado un problema NP -difícil, si cada problema en NP es polinomialmente reducible a X .

Definición 42 *Problema NP-Completo.* Un problema X es llamado Problema NP -Completo si:

1. X pertenece a NP .
2. X es NP -difícil.

Lema 22 *Un problema X es un problema NP -Completo si:*

1. X pertenece a NP .
2. $\exists Y$ polinomialmente reducible a X , para algún Y que es NP -Completo.

Demostración: Por la condición 2 en la definición de NP -Completo, cada problema en NP es polinomialmente reducible a Y . Pero debido a que Y es polinomialmente reducible a X y la reducibilidad es una relación transitiva, cada problema en NP es polinomialmente reducible a X también.

■

Muchos de los problemas clásicos NP -completos pertenecen a la Teoría de gráficas; por ejemplo, el problema del conjunto independiente máximo o el problema de coloración de vértices. Para estos problemas, hasta el momento, no se conocen algoritmos de orden polinomial que los resuelvan.

Apéndice B

Sistema GSM

El sistema GSM de telefonía móvil celular debe sus siglas al *Group Spéciale Mobile* creado en 1980 con el objeto de definir un sistema de cobertura paneuropea y de naturaleza digital, capaz de dar servicio a un gran número de suscriptores y que pudiera integrarse fácilmente en las nuevas redes de telecomunicación RDSI. Este grupo inició sus trabajos en el año 1985 concluyendo la especificación del sistema en 1991.

El sistema GSM es un sistema:

- De telefonía móvil. Puesto que permite la comunicación entre usuarios móviles.
- Celular. La zona de cobertura se divide en zonas llamadas celdas a las cuales se asigna un cierto número de canales de radio o frecuencias. Estas celdas cuentan con una infraestructura de estaciones base (a cada celda le corresponde una estación base), que transmiten con la potencia necesaria para alcanzar toda el área de cobertura.

La división en celdas permite el reuso de frecuencias y el consiguiente aumento de suscriptores a los que la red puede dar servicio simultáneamente, sin necesidad de aumentar el grupo de frecuencias contratadas. Cada celda o estación base puede asimismo incluir uno o varios transmisores según el volumen de tráfico cursado.

Así, una celda con 3 transmisores podrá emitir con 3 frecuencias simultáneamente y soportar 3 veces más tráfico que una celda con un solo transmisor.

- Digital. Dado que supone la solución más económica, sencilla y eficaz para asumir la demanda del mercado e incorporar los diferentes avances tecnológicos, frente a la solución analógica empleada por otros sistemas de telefonía móvil.

En el año 1993 había 36 redes GSM en 22 países y otros 25 estaban considerando la posibilidad de implantarlo. Aunque el sistema GSM es un sistema típicamente europeo, existen redes GSM operativas en alrededor de 80 países de todo el mundo. Con motivo de este desarrollo, se ha asignado un nuevo término al acrónimo GSM: Global System for Mobile Communications.

El sistema GSM ofrece importantes ventajas tanto a los operadores como a los usuarios del sistema. De entre ellas destacan:

- Utilización eficiente del medio de transmisión.
- Mejora en la calidad de la señal de voz (incluso en condiciones de recepción débil de señal y alta interferencia) a pesar de que el uso del codificador de voz puede provocar ciertas variaciones en el timbre de voz y un retardo hasta cierto punto apreciable.
- Posibilidad de comunicación entre distintos países sin la necesidad de cambiar de terminal, con el único requisito de acuerdos pertinentes entre los diferentes operadores.
- Aparición de la SIM del tamaño de una tarjeta de crédito, en donde se registran los datos del inscrito. Evita también robos y usos fraudulentos.
- Conversaciones seguras por la encriptación de la información en el canal radio.

- Prestaciones adicionales como la transmisión de datos y fax, y servicios suplementarios tales como el redireccionamiento de llamada, llamada en espera, identificación de llamada entrante, etcétera.

B.1. El subsistema radio.

El subsistema radio, que constituye el enlace físico entre las estaciones móviles y la red fija, está formado por una serie de canales. El sistema GSM emplea una modulación multiplexada en tiempo y frecuencia. De esta forma, una celda puede transmitir diversas frecuencias (llamadas portadoras) simultáneamente. Cada frecuencia portadora puede soportar además distintos canales (en general dedicados al tráfico de llamadas) gracias a la multiplexación en tiempo, dividiéndose cada trama en 8 ranuras o segmentos de tramas, correspondientes con otros tantos canales lógicos, generalmente 1 de control y 7 de tráfico por cada portadora, o bien 8 de tráfico si otra de las frecuencias portadoras de la misma celda ya contiene un canal de control. El estándar GSM distingue los canales de usuario, que cursan y gestionan las llamadas, de los canales de control propiamente dichos, según la jerarquía siguiente:

- Canales de usuario:
 - Canal de tráfico.
 - Canales de control, asociados durante la comunicación.
 - Lento: Control de potencia, medidas, avance temporal.
 - Rápido: Información de traspaso.
- Canales de control:
 - Difusión: Base-Móviles.
 - Acceso: Base-Móviles-Dedicados.

Como se mencionó, los canales de usuario ocupan las ranuras no destinadas a los canales de control, apareciendo éstos como máximo 1 vez por trama y por tanto quedando 7 u 8 ranuras de una trama destinadas al tráfico y al control de tráfico según el siguiente esquema:

Los canales de tráfico soportan la transmisión de la información de voz y datos en forma de ráfagas.

Los canales de control asociados aparecen una vez que se ha establecido la comunicación y son propios del usuario. Se distinguen dos tipos de canales de control asociado:

- **Lentos:** Transportan información de control de potencia, la estación base informa al móvil si ha de aumentar o disminuir la potencia transmitida, y medidas de potencia del móvil a la estación base para que ésta decida el control de potencia e información de avance temporal.
- **Rápidos:** Transmiten la información de traspaso cuando el usuario se traslada de una celda a otra, que debe transferirse al móvil antes de que se interrumpa la comunicación. Por este motivo la transmisión se lleva a cabo sobre un canal de tráfico, pudiendo producirse una ligera degradación de la transmisión.

Por otra parte, los canales de control son comunes a todos los usuarios de una celda siendo utilizados en los pasos previos al establecimiento de la llamada. Estos canales, que han de estar presentes en todas las celdas de la red ocupan normalmente una sola ranura de una de las portadoras de la celda, respondiendo a la siguiente jerarquía:

- Los canales de difusión se transmiten en el enlace descendente, de la estación base a la terminal móvil y proporcionan distintos tipos de información: configuración del sistema, frecuencia portadora de la estación base que permite sintonizar el móvil, frecuencias con canal de control, sincronización con la trama de la estación base, entre otros.

- Los canales de acceso son canales ascendentes o de subida, es decir, van del móvil a la estación base, que, al ser comunes a todos los usuarios, requieren la presencia de un mecanismo de acceso al medio. Entre otras funciones, controlan las peticiones de llamadas de los usuarios.

- Finalmente, se encuentran los canales de control dedicados, destinados a transmitir información de control entre la red y el móvil con funciones específicas, como el intercambio de datos de usuario entre la estación base y el móvil antes de realizarse la comunicación.

B.2. Reducción del nivel de interferencias.

Como todo sistema de telefonía celular, GSM permite el reuso de frecuencias de diferentes estaciones base con el objeto de aumentar la capacidad de tráfico de la red. Esta reutilización conlleva, sin embargo, la posible aparición de interferencias entre celdas. De este modo, junto al diseño de una planificación de frecuencias adecuada para determinar cuál debe ser la distancia de reuso entre celdas y la utilización de una modulación robusta, el estándar GSM incorpora tres mecanismos para compensar los efectos de las interferencias, siendo decisión de la operadora incluirlos o no en su servicio:

1. Control de potencia: permite modificar dentro de un cierto rango la potencia transmitida en ambas direcciones, de móvil a base y viceversa. En GSM se aplica un control de potencia, independiente en ambas direcciones, en función de los niveles de recepción medidos por la estación base y la terminal móvil.

2. Transmisión discontinua: detenta la existencia de silencios en la comunicación e inhibe la transmisión radio durante este tiempo (aproximadamente el 50 % del tiempo de una llamada nos encontramos en la escucha) para no interferir en celdas contiguas.

3. Conmutación de Frecuencia (*frequency hopping*) o (FH): varía la frecuencia portadora de transmisión, de una trama a otra.

A pesar de que un operador suele contratar un número fijo de frecuencias, es posible encontrar diseños frecuenciales GSM en rangos de distinta amplitud, en el sentido no sólo de que el número de frecuencias empleadas en cada celda sea distinto de un diseño a otro, sino de que el propio número de frecuencias disponibles por la red pueda variar significativamente. La planificación frecuencial se puede llevar a cabo de forma segmentada, dividiendo el área de cobertura, sea una región, un área metropolitana o una ciudad de dimensiones variables.

Apéndice C

Lista de símbolos

- $\chi(G)$ El número cromático de la gráfica G .
- $d_G(u, v)$ La distancia entre los vértices u y v en la gráfica G .
- $\delta(G)$ Es el grado mínimo de un vértice u para todo $u \in V(G)$
- $w(u)$ El componente u del vector de pesos w .
- w_{max} $\text{máx}\{w(u) \mid v \in V(G)\}$
- w_{min} $\text{mín}\{w(u) \mid v \in V(G)\}$
- $w(S)$ Denota la suma de los pesos de todos los nodos en un conjunto S
- $\rho(G)$ la densidad de G .
- $\Delta(G)$ Es el grado máximo de un vértice u para todo $u \in V(G)$

(G, w) Un grafo ponderado no dirigido.

$\sigma(f)$ Diferencia entre la mínima y la mayor asignación de canales asignados por f

$\sigma(G, w)$ Amplitud mínima para cualquier asignación de canales para (G, w)

$[x]$ función el menor entero mayor o igual a x .

$\lfloor x \rfloor$ función mayor entero menor o igual a x .

$\|f(G)\|$ $\|f(G)\| = \text{máx}\{f(v) \mid v \in V(G)\}$

$\lambda(G, d)$ El mínimo de $\|f(G)\|$

C.1. Vocabulario técnico bilingüe.

Se elaboró el presente vocabulario con la traducción que hemos dado en estas notas a los principales términos tomados de la literatura en inglés.

Amplitud	Span.
Amplitud Cíclica	Cyclic span.
Canales de Comunicación	Communication links.
Clan	Clique.
Conmutación de Frecuencia	Frequency hopping.
Ejemplar	Instance.
Ejemplar	Instance of a problem.
Etiquetación	Labelling.
Gráfica con restricciones	Constrained graph.
Gráfica con restricciones y pesos	Constrained, weighted graph.
Malla	Grid. Lattice.
Np-difícil	NP-Hard.
Problema difícil	Hard problem.
Razón de desempeño	Performance ratio.
Restricción co-canal	Co-channel constraint.
Restricción co-sitio	Co-site constraint.
Restricción inter-sitio	Inter-site constraint.
Reuso de canales	Frequency reuse.

Bibliografía

- [1] G. Chartrand. *Introductory Graph Theory*. Dover Publication, 1977.
- [2] T. H. Cormen. *Introduction to Algorithms*. Mc Graw-Hill, Inc., 1990.
- [3] H. M. Deitel & P. J. Deitel. *Cómo Programar en Java*. Prentice Hall, México, 1998.
- [4] J. Van den Heuvel & R. Leese & M. Shepherd. *Graph labelling and radio channel assignment*. *Journal of Graph Theory*., 29(4), 1998.
- [5] R. Diestel. *Graph Theory*. Graduate texts in mathematics. Springer, 2 ed., 2000.
- [6] R. K. Yeh & J. R. Griggs. *Labelling graphs with a condition at distance 2*. *SIAM J.Discr. Math.*, 5(4:586-595), 1992.
- [7] S. M. Elnoubi & R. Sing & S. C. Gupta. *A new frequency channel assignment algorithm in high capacity mobile communication systems*. *IEEE Trans. Vehe. Technol*, 31(3):125-131, 1982.
- [8] F. Harary. *Graph Theory*. Addison-Wesley, Reading, Mass, 1969.
- [9] J. C. M. Janssen. *Channel assignment and graph labeling*. Dept. of Math & Stats. Dalhousie University. Halifax, N. S. Canada, 2002.
- [10] M. R. Garey & D. Johnson. *Computers and Intractability*. Bell Telephone Laboratories., 1979.
- [11] D. C. Kozen. *The design and analysis of algorithms*. Springer-Verlag, New York., 1991.

- [12] U. Manber. *Introduction to Algorithms. A Creative Approach*. Addison Wesley, USA, 1990.
- [13] J. Janssen & K. Kilakos & O. Marcotte. *Fixed preference frequency allocation for cellular telephone systems*. *IEEE Trans. Veh. Techn.*, 48(2):533–541, 1999.
- [14] S. Fitzpatrick & J. Janssen & R. Nowakowski. *Distributive Online Channel Assignment for Hexagonal Cellular Networks with Constraints*. *Technical Report*, G-2000-14, GERARD, 2000.
- [15] S. Fitzpatrick & J. Janssen & R. J. Nowakowski. Distributive online channel assignment for hexagonal cellular networks with constraints. In *ICALP Satellite Workshops*, pages 147–154, 2000.
- [16] G. Chartrand & O. R. Oellermann. *Applied an Algorithmic Graph Theory*. International Series in Pure and Applied Mathematics. McGraw-Hill, 1993.
- [17] J. Ozón. *Contribución al coloreado de grafos y las redes pequeño-mundo*. 2001.
- [18] J. Abril & A. Cortes & J. Ozón. *Sistemas Multiagente para la asignación de frecuencias en Redes celulares*. 2000.
- [19] G. J. E. Rawlins. *Compared to what?: an introduction to the analysis of algorithms*. W. H. Freeman and Company, USA, 1991.
- [20] A. Raychaudhuri. *Further results on T-coloring and frequency assignment problems*. *SIAM J. Discrete Math.*
- [21] A. Raychaudhuri. *Intersection assignment, T-coloring and powers of graphs*. Ph. D. Thesis, Department of Mathematics, Rutgers University, 1985.
- [22] P. A. Raymond. *Performance analysis of cellular networks*. *IEEE Trans. Comun.*, 39(12):1787–1793, 1991.
- [23] F. S. Roberts. *T-colorings of graphs. Recent results and open problems*. *Research Report*, RRR 7-86, 1986.

- [24] F. S. Roberts. *T-colorings of graphs. Recent results and open problems. Research Report*, private communication, 1988.
- [25] M.B. Cozzens & F. S. Roberts. *T-colorings of graphs and the channel assignment problem. Congress Numer.*, 35, 1982.
- [26] D. Sakai. *Labeling chordal graphs: distance two condition. SIAM J.Discr. Math.*, 7:133-140, 1994.
- [27] R. Sedgewick. *Algorithms in C*. Addison-Wesley Publishin Company, USA, 1996.
- [28] R. Sedgewick. *An introduction to the analysis of algorithms*. Addison-Wesley Publishin Company, USA, 1996.
- [29] J. Janssen & D. Krizanc & L. Narayanan & S. M. Shende. *Distributed Online frequency assignment in Cellular Networks. J. Algorithms*, 1988.
- [30] L. Narayan & S. Shende. *Static frequency assignment in cellular networks*. In Proceedings of SIROCCO 97. Carleton Scientific Press, 1977.
- [31] url.: <http://www.morfonet.cl/secciones/informe/1a02.htm>. Consultada en Enero 2004.
- [32] url.: <http://es.wikipedia.org/wiki/Algoritmo>. Consultada en Julio 2004.
- [33] url.: <http://algoritmia.cjb.net/>. Consultada en Mayo 2004.
- [34] url.: <http://www.monografias.com/trabajos11/alcom/alcom.shtml>. Consultada en Noviembre 2003.
- [35] url.: <http://www.revistapolis.cl/2/hardy.htm>. Consultada en Noviembre 2003.
- [36] M.B. Cozzens & D. I. Wang. *The general channel assignment problem. Congress Numer.*, (41):115-129, 1984.
- [37] M. A. Weiss. *Data structures & Algorithm analysis in Java*. Addison Wesley Longman, USA, 1999.

- [38] M. A. Weiss. *Estructuras de datos en JAVA*. Addison Wesley, Madrid, 2000.
- [39] R. Eckstein & M. Loy & D. Wood. *Java Swing*. O'Reilly, USA, 1998.

Índice alfabético

- algoritmo, 106
 - análisis, 106, 107
 - características, 13, 106
 - para la Asignación de canales, 67
- algoritmos distribuidos, 96
 - k-local, 96
 - préstamo Aritmético, 97
- amplitud, 53
- amplitud cíclica, 72
- análisis de Algoritmos, 106, 107
- asignación aritmética, 95
- asignación de canales, 53
 - algoritmos, 67
 - algoritmos distribuidos para la, 96
 - asignación fija, 72
 - complejidad, 62
 - distribuidos, 93
- camino, 9
 - uv*-camino, 9
- categoría de orden, 111
- ciclo, 11
 - etiquetación, 58
- clan, 11
- clase de color, 19
- clases de complejidad, 112
- coloración, 19
 - número cromático, 20
 - válida, 19
- conjunto independiente, 20
- densidad, 12
- distancia de reuso, 52
- etiquetación, 72
 - óptima, 72
 - mediante progresión aritmética, 72
- etiquetación a distancia 2
 - complejidad, 62, 65
 - NP-completez, 65
- etiquetación de la gráfica, 51
- FMDA, 93
- gráfica, 7
 - asignación de canales, 53
 - bipartita, 10
 - bipartita completa, 10
 - camino
 - longitud, 9
 - camino en, 9
 - complemento de una gráfica, 10

- completa, 10
- con restricciones, 52
- con restricciones y pesos, 52
- conexa, 9
- diámetro, 10
- distancia, 9
- grado, 8
- orden, 8
- subgráfica, 8
- tamaño, 8
- trayectoria en, 9
- gráfica hexagonal, 94
- multicoloración, 38
 - número r -multicromático, 38
- n -coloración, 19
- número cromático, 20
- notación asintótica., 110
- NP Completo, 117
- NP-difícil, 117
 - problema, 117
- orden
 - categoría, 111
- polinomialmente transformable, 116
- préstamo Aritmético, 97
- problema, 113
 - clasificación, 114
 - según la complejidad, 115
 - según los requerimientos, 114
 - no determinístico, 117
 - NP, 117
 - NP Completo, 117
 - NP-difícil, 117
 - polinomial, 115
 - polinomialmente transformable, 116
- restricción
 - co-sitio, 52
 - de adyacencia de canal, 52
 - inter-sitio, 52
- reuso de canales, 2
- span, 53
- subgráfica, 8
- trayectoria, 9
 - etiquetación, 57
- vértice
 - excentricidad de un, 9
 - peso de un, 53
- vector de pesos, 53