

**UNIVERSIDAD COMPLUTENSE DE MADRID**

**FACULTAD DE CIENCIAS MATEMÁTICAS**

Departamento de Estadística e Investigación Operativa I



**EXTENSIONES DEL PROBLEMA DE COLORACIÓN DE  
GRAFOS**

**MEMORIA PARA OPTAR AL GRADO DE DOCTOR  
PRESENTADA POR**

Javier Ramírez Rodríguez

Bajo la dirección del Doctor:

Francisco Javier Yáñez Gestoso

**Madrid, 2001**

**ISBN: 84-669-1855-8**

UNIVERSIDAD COMPLUTENSE DE MADRID  
FACULTAD DE CIENCIAS MATEMÁTICAS  
Departamento de Estadística e Investigación Operativa I

EXTENSIONES DEL PROBLEMA DE  
COLORACIÓN DE GRAFOS  
Javier Ramírez Rodríguez

Memoria para optar al grado de Doctor  
En Ciencias Matemáticas, realizada bajo la  
Dirección del Dr. D. Javier Yáñez Gestoso

Madrid, Noviembre de 2000.



**UNIVERSIDAD COMPLUTENSE DE MADRID**  
**FACULTAD DE CIENCIAS MATEMÁTICAS**  
Departamento de Estadística e Investigación Operativa I

**EXTENSIONES DEL PROBLEMA DE  
COLORACIÓN DE GRAFOS**

Javier Ramírez Rodríguez

Memoria para optar al grado de Doctor  
en Ciencias Matemáticas, realizada bajo la  
dirección del Dr.D.Javier Yáñez Gestoso.

Madrid, Noviembre de 2000

## DEDICATORIA

A Ana Lilia, por haberme *empujado* a experimentar las emociones propias de esta actividad y su apoyo para salir de más de una.

A Emiliano.

A la memoria de mis Padres...

## AGRADECIMIENTOS

Agradezco al Profesor Javier Yáñez Gestoso su guía y apoyo durante todo el desarrollo del proyecto y el haberse involucrado y trabajado en el tema como lo hizo, lo que permitió que la tesis estuviera en el tiempo y profundidad actuales.

A Ana Lilia el haberme enseñado buenas costumbres en el arte de programar.

A mis amigos Rafael López Bracho y Miguel Angel Gutiérrez Andrade, por compartir sus experiencias de teoría de grafos y combinatoria conmigo, en los seminarios previos a esta etapa y el apoyo que me brindaron durante mi ausencia en la UAM-A.

## RESUMEN

Una coloración válida de un grafo es una asignación de colores a cada vértice de tal manera que cualquier par de nodos adyacentes no tengan el mismo color. Algunos problemas de planificación del tiempo o de recursos se pueden plantear como problemas de coloración de los vértices de un grafo; los vértices representan los elementos a programar y hay una arista entre dos vértices cuando los elementos representados por éstos son incompatibles; una coloración válida identifica una asignación de recursos compatibles. Si el objetivo es usar el mínimo número de colores se tiene planteado el problema de coloración mínima.

En este trabajo se presentan extensiones del problema de coloración, al introducir el nuevo requerimiento de que el número de colores no tiene que ser necesariamente el mínimo, esto permite modelizar otros problemas de planificación.

Se introduce primero el Problema de Coloración Robusta, que consiste en determinar una coloración con  $c$  colores que minimice el grado de rigidez; la rigidez de una coloración distingue las aristas complementarias penalizándolas cuando sus extremos comparten el mismo color, permitiendo así obtener coloraciones válidas con diferentes propiedades.

Se proponen dos algoritmos para encontrar soluciones aproximadas uno es de enumeración parcial y el otro es un híbrido de un genético con uno voraz, cuyas soluciones se consideran aceptables después de haber sido comparadas con las exactas, obtenidas de modelos de programación matemática del problema.

También se presenta el Problema de Coloración Robusta Generalizado, en el que se relaja el concepto de incompatibilidad de una coloración respecto al Problema de Coloración Robusta.

Finalmente se presentan dos generalizaciones difusas del problema de coloración, una basada en la difuminación del concepto de color y otra que consiste en plantear el problema de coloración mínima en un grafo difuso. A partir de esta última se introduce el concepto de número cromático difuso.

# Índice General

<b>1</b>	<b>El Problema de Coloración Mínima</b>	<b>13</b>
1.1	Planteamiento del Problema . . . . .	13
1.2	Ejemplos . . . . .	15
1.2.1	Programación de Exámenes . . . . .	15
1.2.2	Determinación de Conglomerados . . . . .	16
1.2.3	Coloración de un mapa . . . . .	16
1.3	Modelos de Programación Matemática . . . . .	17
1.3.1	Modelo de programación entera . . . . .	18
1.3.2	Modelo de programación binaria . . . . .	19
1.3.3	Modelo de partición . . . . .	21
1.3.4	Modelo de asignación . . . . .	22
1.3.5	Modelo de asignación cuadrática . . . . .	23
1.4	Otros problemas de coloración . . . . .	25
<b>2</b>	<b>Problema de Coloración Robusta</b>	<b>29</b>
2.1	Justificación del problema . . . . .	29
2.2	Programación de Exámenes . . . . .	30
2.3	Determinación de Conglomerados . . . . .	34
2.4	Coloración de un mapa . . . . .	38
2.5	Planteamiento del Problema . . . . .	39



2.6	Complejidad algorítmica del Problema . . . . .	48
<b>3</b>	<b>Modelos matemáticos del PCR</b>	<b>51</b>
3.1	Modelo de programación binaria . . . . .	51
3.2	Cliques del grafo complementario. Modelo de partición . . . . .	53
3.3	Modelo de Asignación Cuadrática . . . . .	61
3.4	Modelo de red generalizada . . . . .	64
3.5	Experiencias computacionales . . . . .	67
<b>4</b>	<b>Heurísticas para el Problema de Coloración Robusta</b>	<b>71</b>
4.1	Heurística basada en $k$ -cliques . . . . .	71
4.2	Algoritmo híbrido genético-voraz . . . . .	82
4.2.1	Representación y función de ajuste . . . . .	85
4.2.2	Algoritmo de coloración robusta voraz . . . . .	86
4.2.3	Operador de selección . . . . .	88
4.2.4	Operador de cruce . . . . .	89
4.2.5	Operador de mutación . . . . .	90
4.3	Experiencias computacionales . . . . .	91
<b>5</b>	<b>El Problema de Coloración Robusta Generalizado</b>	<b>95</b>
5.1	Planificación con restricciones de tiempo . . . . .	95
5.2	Planteamiento del Problema . . . . .	99
5.3	Modelo de programación matemática . . . . .	100
<b>6</b>	<b>Enfoque Difuso de Problemas de Coloración</b>	<b>109</b>
6.1	Coloración difusa de un grafo nítido . . . . .	109
6.2	Problema de Coloración Mínima Difusa . . . . .	113
6.2.1	Número cromático difuso . . . . .	117

## PROLOGO

La teoría de grafos ha proporcionado muchos modelos y técnicas de solución eficientes para una gran variedad de problemas que han surgido en diferentes contextos. Uno de estos problemas es el de *colorear los vértices de un grafo*, que consiste en asignar un color —un número entero positivo— a cada vértice de tal manera que cualquier par de nodos adyacentes no tengan el mismo color.

A partir del concepto de coloración de un grafo se plantea de forma natural el *Problema de Coloración Mínima* al buscar el mínimo número de colores que permite colorear un grafo.

El problema de coloración mínima, conocido también como problema de coloración o problema de coloración de grafos, surgió con el teorema de Brooks en 1941, que da una cota superior para el número cromático de cualquier grafo, al analizar la Conjetura de los Cuatro Colores, que ha sido uno de los grandes problemas matemáticos no resueltos durante más de 120 años. En 1852 Francis Guthrie, que planteó este problema a su hermano Frederick, notó que *eran suficientes cuatro colores para distinguir los condados en un mapa de Inglaterra*.

El primer documento escrito que se conoce sobre el problema de los cuatro colores es una carta fechada el 23 de octubre de 1852 escrita por Augustus De Morgan a su colega y amigo Sir William Rowan Hamilton. En esta carta comentaba que su alumno Frederick Gouthrie le había planteado el problema de los cuatro colores y que no había sido capaz de resolverlo. El problema se descuidó hasta 1878 cuando Arthur Cayley dijo a los miembros de la Sociedad Matemática de Londres que no había podido resolverlo.

En [25], Fritsch y Fritsch presentan un estudio completo de las diferentes aportaciones que se fueron haciendo para resolver el problema desde 1880 hasta que fue demostrado en 1976 por Wolfgang Haken y Kenneth Appel de la Universidad de Illinois, EE.UU., quienes usaron un enfoque bien orquestado que comprende entre otras cosas,  $10^{10}$  operaciones en un ordenador de alta velocidad.

La mayoría de las aproximaciones que se han dado para el problema de coloración mínima son para grafos de hasta 1000 vértices, sólo en el trabajo de Joseph C. Culberson y Feng Luo comentan que estaban experimentando con su algoritmo en grafos de hasta 10000 vértices.

En 1972, Richard Karp demostró que el problema de coloración mínima era *NP*-Completo, en 1973 Stockmeyer y en 1976 Garey, Johnson y Stockmeyer reforzaron el resultado de Karp demostrando que el problema de colorear un grafo con  $k$  colores es *NP*-Completo para cualquier  $k$  mayor o igual a 3, ver [53].

Los problemas que se han modelizado como problemas de coloración mínima de grafos son variados y van desde los que sólo tienen interés histórico o pedagógico hasta las aplicaciones en diversas áreas: El problema de las 8 damas, planteado por Gauss, consiste en colocar en un tablero de ajedrez 8 damas sin que se puedan comer entre ellas [2]. El problema de las colegialas, que se puede enunciar como sigue: en un colegio en el que hay  $2p$  chicas se organizan paseos para recorrer en un pequeño tren parte de un parque, las chicas van sentadas en parejas y se pregunta si se pueden hacer  $2p - 1$  paseos en igual número de días, de manera que cualesquiera dos chicas hagan pareja una vez [2]. El problema de conglomerados, que consiste en buscar una partición de un conjunto de datos en clases, tales que dos elementos de la misma clase están más cerca uno del otro que dos elementos de clases distintas [30]. Un problema de sistemas de manufactura en el que se intenta agrupar en un mismo lugar partes de manufactura que requerirán tantas herramientas comunes como sea posible [15]. Problema de asignación de turnos a diferentes líneas de ferrocarril [7]. En [60] se detallan más aplicaciones del problema de coloración de grafos.

Los problemas de planificación del tiempo y de recursos son numerosos y difieren entre sí no sólo por las restricciones que tiene cada uno de ellos, sino también por la complejidad de las mismas. Algunos ejemplos son: problemas de horarios de cursos, formulados en términos de coloración de grafos o coloración de hipergrafos [66], problema de coloración de

aristas para modelos de planificación [67], coloración de intervalos de grafos ponderados, aplicados a problemas de horarios con cursos de diferente duración [8]. Mas referencias pueden verse [5], [6] y [4].

Desde el punto de vista computacional, el problema de coloración mínima es muy interesante, puesto que se ha demostrado que es de los más difíciles de resolver entre los *NP*-Duros, pues no hay algoritmos que en tiempo polinomial den buenas aproximaciones como ocurre con otros problemas de optimización combinatoria, es el caso, por ejemplo, del problema de empaquetamiento o de la versión del problema del viajante en que se cumple la desigualdad triangular [26].

Por lo anterior, y aún con todos los adelantos habidos tanto en teoría de algoritmos como en la capacidad de los ordenadores para hacer cálculos, sólo es posible conocer soluciones aproximadas del problema. En los últimos quince años se han estado proponiendo heurísticas que en su momento han dado buenas soluciones al problema de coloración, entre las que están temple simulado [15], búsqueda tabú [31] y [36], algoritmos genéticos [20], sistemas de hormigas [17], híbridos [24], heurísticas para colorear cierto tipo de grafos [45]. Para más referencias sobre las heurísticas anteriores y otras consultar el censo de Pardalos et al. [53] y la muestra reciente de Johnson y Trick [37].

En esta memoria se han introducido algunas extensiones del problema de coloración mínima que han dado lugar a otros tantos problemas de coloración de grafos.

En el Capítulo 1 se hace una introducción histórica del problema de coloración mínima. Se presentan algunos problemas que pueden ser modelizados como problemas de coloración mínima; concretamente, un problema de programación de horarios de exámenes, un problema de conglomerados y un problema de coloración de grafos planares. Se detallan seis formulaciones diversas del mismo como problemas de programación matemática. Por último se comentan otras extensiones de problemas de coloración introducidas por otros autores.

En el Capítulo 2 se introduce el *Problema de Coloración Robusta*. Se justifica su interés al introducir en determinados problemas de coloración mínima nuevos requerimientos en las coloraciones; concretamente, la posibilidad de convertir el criterio de minimizar el número de colores utilizados en una restricción y buscar nuevos criterios de optimización que permitan comparar las distintas coloraciones obtenidas con un número fijado de colores. Se propone así el concepto de rigidez de una coloración, que se basa en penalizar la coincidencia de colores entre los extremos de las aristas complementarias. Se relaciona este nuevo problema con el de coloración mínima y se analiza su complejidad.

En el Capítulo 3, se presentan distintas modelizaciones del Problema de Coloración Robusta. El primer modelo, de programación binaria, es la extensión natural de uno de los modelos de programación matemática que resuelve el problema de coloración mínima. A partir de los clanes del grafo complementario, se introduce un modelo de partición. El Problema de Coloración Robusta se puede plantear también como un problema de asignación cuadrática. Por último se modeliza como una red generalizada. Se analizan estos modelos y se aportan algunas experiencias computacionales.

A partir del análisis de los modelos del capítulo 3 se puede concluir que, salvo para grafos de tamaño reducido, los algoritmos exactos no pueden aportar soluciones válidas. En el Capítulo 4 se proponen dos algoritmos aproximados para el Problema de coloración Robusta. Se introduce primero un algoritmo de enumeración parcial basado en los clanes del grafo complementario. El segundo algoritmo se basa en la metaheurística de los algoritmos genéticos. Se analizan computacionalmente ambos algoritmos.

En el Capítulo 5 se propone un nuevo problema de coloración de grafos que generaliza el problema de coloración robusta, por lo que se denominará *Problema de Coloración Robusta Generalizado*. En este problema, de forma análoga al problema de coloración robusta, se fija el número de colores válidos, pero se penalizan las aristas complementarias sólo cuando los extremos tengan asignados unos colores suficientemente próximos. La introducción de una distancia entre colores proporciona una gran flexibilidad a los proble-

mas que pueden ser resueltos con este nuevo planteamiento. Se introduce un problema de horarios para justificar este problema. Después de plantear el nuevo problema se propone un modelo de programación matemática, que generaliza el correspondiente modelo del problema de coloración robusta.

Otra extensión que se aporta en esta memoria se basa en enfocar desde la teoría de los conjuntos difusos el Problema de Coloración . Se estudian en el capítulo 6 dos enfoques de este tipo. Se introduce primero el concepto de Coloración Difusa de un grafo, basado en la difuminación del concepto de color en un grafo nítido. Posteriormente se introduce *el Problema de Coloración Mínima Difusa*, que plantea el problema de coloración mínima en un grafo difuso que resulta de difuminar el concepto de arista. Se introduce un problema de programar las fases de un semáforo (traffic light phasing problem) para justificar este nuevo problema y que permitirá introducir el concepto de *Número Cromático Difuso*.



# Capítulo 1

## El Problema de Coloración Mínima

### 1.1 Planteamiento del Problema

El problema de colorear los vértices de un grafo, es muy conocido en teoría de grafos y consiste en asignar un color a cada vértice de forma que cualquier par de vértices adyacentes no tengan el mismo color. Este problema surgió como un intento por resolver la Conjetura de los Cuatro Colores, con el teorema de Brooks en 1941, que da una cota superior para el número cromático de cualquier grafo. El desarrollo de la teoría de grafos ha utilizado, entre otras, muchas de las ideas propuestas para intentar resolver la Conjetura de los Cuatro Colores, que ha sido uno de los grandes problemas matemáticos no resueltos durante más de 120 años, desde que en 1852 Francis Guthrie lo propuso a su hermano Frederick. Francis *notó que eran suficientes cuatro colores para distinguir los condados en un mapa de Inglaterra.*

El primer documento escrito que se conoce sobre el problema de los cuatro colores es una carta fechada el 23 de octubre de 1852 escrita por Augustus De Morgan a su colega y amigo Sir William Rowan Hamilton. Escribió que su alumno Frederick Gouthrie le había planteado el problema de los cuatro colores y que no había sido capaz de resolverlo. El problema se descuidó hasta 1878 cuando Arthur Cayley dijo a los miembros de la Sociedad



Matemática de Londres que no había podido resolverlo.

En 1880 el abogado Alfred Bray Kempe publicó una "prueba" de la conjetura y el mismo año, también Peter Guthrie Tait publicó, según él, una nueva prueba, pero sólo hizo unas pocas reformulaciones interesantes del problema, por lo que se siguió considerando a Kempe como el que había realizado la prueba. En 1890 Percy John Heawood demostró que la prueba de Kempe no era correcta y usó una parte de ésta para hacer la primera demostración del problema de los cinco colores, que dice: Los países de cualquier mapa de la esfera se pueden colorear con cinco colores.

Fritsch y Fritsch [25] presentan un estudio completo de las diferentes aportaciones que se fueron haciendo para resolver el problema, hasta que fue demostrado en 1976 por Wolfgang Haken y Kenneth Appel de la Universidad de Illinois, EE.UU., quienes usaron un enfoque bien orquestado que comprende entre otras cosas,  $10^{10}$  operaciones en un ordenador de alta velocidad.

Sin embargo, la salida no está muy clara debido a la cantidad asombrosa de cálculos que se hicieron en 1200 horas y también a lo elaborado del argumento necesario para entender cómo los cálculos resuelven el problema.

Formalmente, dado un grafo  $G = (V, E)$ , con  $|V| = n$ , se define una *función de coloración* como una aplicación

$$C : \{1, \dots, n\} \longrightarrow \{1, 2, \dots\}$$

que identifica a  $C(i)$  como el color del vértice  $i \in V$  de forma que dos vértices adyacentes,  $\{i, j\} \in E$  no tienen el mismo color, es decir,  $C(i) \neq C(j)$ .

Dada una función de coloración  $C$  de un grafo  $G$ , al conjunto de vértices que tienen asignado el mismo color se le llama *clase de color*:

$$V_C(k) = \{i \in V \mid C(i) = k\} \quad \forall k \in \{1, 2, \dots\}$$

Una  $k$ -coloración es una función de coloración que no utiliza más de  $k$  colores:

$$C^k : \{1, \dots, n\} \longrightarrow \{1, 2, \dots, k\}$$

Un grafo es  $k$ -coloreable si admite una  $k$ -coloración. Al mínimo valor  $k$  tal que  $G$  es  $k$ -coloreable se le llama el *número cromático* del grafo y se denota por  $\chi(G)$ .

Dado un grafo  $G$ , el Problema de Coloración Mínima, denotado por  $PCM$  en lo que sigue, busca una función de coloración que no utilice más de  $\chi(G)$  colores.

En 1972 Richard Karp demostró que el problema de coloración mínima de grafos es NP-Completo, en 1973 Stockmeyer y en 1976 Garey, Johnson y Stockmeyer reforzaron el resultado de Karp demostrando que el problema de colorear un grafo con  $k$  colores es NP-Completo para cualquier  $k$  mayor o igual a 3, [53]; siendo  $k$  el número cromático.

## 1.2 Ejemplos

Con el fin de justificar el interés real que tiene el problema de coloración mínima se presentan los siguientes problemas que se pueden plantear como problemas de coloración.

### 1.2.1 Programación de Exámenes

Se necesita programar los exámenes en un Centro Universitario. Si dos asignaturas comparten al menos un alumno no se pueden programar el mismo día. Teniendo en cuenta esta restricción, se puede construir el grafo de incompatibilidad  $G = (V, E)$  cuyos vértices son las asignaturas. La arista  $\{i, j\} \in E$  indica que hay al menos un estudiante matriculado en las asignaturas  $i, j \in V$ .

Éste es un problema típico de planificación de recursos, los elementos a planificar son las asignaturas y el recurso que determina la incompatibilidad entre las asignaturas es el día en que se realizarán los correspondientes exámenes.

### 1.2.2 Determinación de Conglomerados

Consiste en buscar un conglomerado o partición de un conjunto de elementos  $\{1, \dots, n\}$  en clases, de forma que dos elementos de la misma clase están más cerca uno del otro que dos elementos de clases diferentes.

La cercanía entre los elementos se define a partir de una distancia o medida de disimilitud

$$d : \{1, \dots, n\} \times \{1, \dots, n\} \longrightarrow [0, \infty)$$

y de un umbral  $\alpha > 0$  fijado previamente: dos elementos se consideran próximos si su disimilitud es menor o igual que dicho umbral  $\alpha$ .

Este problema se puede modelar como un problema de coloración de grafos a partir de la distancia  $d$  y el umbral  $\alpha$ : Se introduce el grafo  $G_\alpha = (V, E_\alpha)$ , siendo  $V = \{1, \dots, n\}$  e imponiendo que  $\{i, j\} \in E_\alpha$  si y sólo si  $d(i, j) > \alpha$ . En otras palabras, uniendo con una arista a dos elementos cuando su disimilitud sea mayor que el umbral fijado.

Cualquier coloración válida del grafo  $G_\alpha$  induce un conglomerado en el que las clases están definidas por los vértices que están igualmente coloreados. El menor número de clases del conglomerado se asocia así al número cromático del grafo  $G_\alpha$ .

### 1.2.3 Coloración de un mapa

El problema consiste en determinar el menor número de colores que permite pintar un mapa geográfico de tal manera que dos regiones con una frontera común tengan colores diferentes.

Al mapa se le puede asociar un grafo, que será un grafo planar, cuyos vértices son cada una de las regiones del mapa y habrá una arista entre dos vértices si las regiones correspondientes tienen una frontera común. Una coloración de los vértices es equivalente a una coloración de las regiones. El problema se reduce a determinar el número cromático del grafo asociado al mapa.

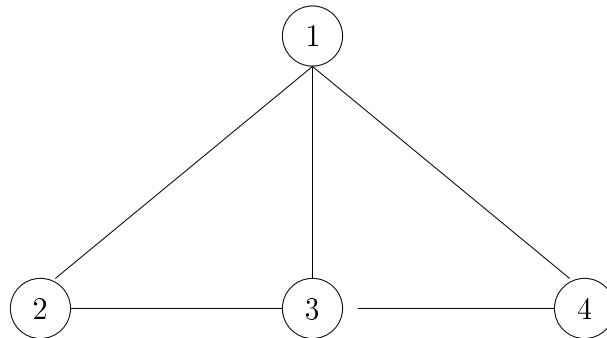


Figura 1.1: Representación del grafo del ejemplo 1.1

El problema de los cuatro colores está resuelto y es equivalente a afirmar que el número cromático de cualquier grafo planar es siempre menor o igual que cuatro.

### 1.3 Modelos de Programación Matemática

El PCM admite varias formulaciones como problema de programación matemática. Al trabajar con grafos finitos, se supondrá, sin pérdida de generalidad, que el conjunto de vértices es  $V = \{1, 2, \dots, n\}$ . Obviamente, cualquier grafo con  $n$  vértices es siempre  $n$ -coloreable, por lo que el número de colores está acotado siempre.

Con el fin de ilustrar las distintas formulaciones, se propone el siguiente grafo.

**Ejemplo 1.1** Sea el grafo  $G = (V, E)$  con cuatro vértices y cinco aristas

$$V = \{1, 2, 3, 4\} \quad E = \{\{1, 2\}; \{1, 3\}; \{1, 4\}; \{2, 3\}; \{3, 4\}\}$$

representado en la figura 1.1

El número cromático es  $\chi(G) = 3$  y una coloración que determina —no la única— es

$$C(1) = 1 \quad C(2) = 2 \quad C(3) = 3 \quad C(4) = 2$$

### 1.3.1 Modelo de programación entera

Las variables de decisión del modelo identificarán la coloración de los vértices:

$$x_i = C(i) \in \{1, \dots, n\} \quad \forall i \in \{1, \dots, n\}$$

El modelo de programación es el siguiente:

$$\begin{aligned} \min \quad & \gamma \\ \text{s.a.} \quad & x_i \leq \gamma \quad \forall i \in V \\ & x_i \geq x_j + 1 - n\delta_{ij}, \quad \forall \{i, j\} \in E \\ & x_i \leq x_j - 1 + n(1 - \delta_{ij}), \quad \forall \{i, j\} \in E \\ & \delta_{ij} \in \{0, 1\}, \quad \forall i, j \in V \end{aligned}$$

La variable endógena  $\gamma$  identifica el color de máximo índice utilizado. El conjunto de variables  $\delta_{ij}$  controla las restricciones disyuntivas que evitan que dos vértices adyacentes tengan el mismo color. Lo que puede verse notando que si  $x_i = x_j$ , entonces ninguna asignación factible de  $\delta_{ij}$  satisfará a ambas restricciones.

El modelo de programación entera asociado al ejemplo 1.1 es:

Min  $\gamma$

$$x_1 \leq \gamma$$

$$x_2 \leq \gamma$$

$$x_3 \leq \gamma$$

$$x_4 \leq \gamma$$

$$x_1 \geq x_2 + 1 - 4\delta_{12}$$

$$x_1 \geq x_3 + 1 - 4\delta_{13}$$

$$x_1 \geq x_4 + 1 - 4\delta_{14}$$

$$x_2 \geq x_3 + 1 - 4\delta_{23}$$

$$x_3 \geq x_4 + 1 - 4\delta_{34}$$

$$x_1 \leq x_2 - 1 + 4(1 - \delta_{12})$$

$$x_1 \leq x_3 - 1 + 4(1 - \delta_{13})$$

$$x_1 \leq x_4 - 1 + 4(1 - \delta_{14})$$

$$x_2 \leq x_3 - 1 + 4(1 - \delta_{23})$$

$$x_3 \leq x_4 - 1 + 4(1 - \delta_{34})$$

una solución óptima es  $\gamma = 3$  y las variables tienen los valores que siguen:  $x_1 = 1$ ,  $x_2 = 2$ ,  $x_3 = 3$ ,  $x_4 = 2$  y  $\delta_{12} = 1$ ,  $\delta_{13} = 1$ ,  $\delta_{14} = 1$ ,  $\delta_{23} = 1$ ,  $\delta_{34} = 0$ .

### 1.3.2 Modelo de programación binaria

En esta formulación las variables de decisión son variables binarias:

$$x_{ik} = \begin{cases} 1 & \text{si } C(i) = k \\ 0 & \text{si } C(i) \neq k \end{cases} \quad \forall i, k \in \{1, \dots, n\}$$

y las variables endógenas

$$y_k = \begin{cases} 1 & \text{si el color } k \text{ es usado} \\ 0 & \text{si el color } k \text{ no es usado} \end{cases} \quad \forall k \in \{1, \dots, n\}$$

El modelo de programación es el siguiente:

$$\begin{aligned}
 \min \quad & \sum_{k=1}^n k \cdot y_k \\
 \text{s.a.} \quad & \sum_{k=1}^n x_{ik} = 1, \quad \forall i \in V \\
 & x_{ik} + x_{jk} \leq 1, \quad \forall \{i, j\} \in E \\
 & \sum_{i=1}^n x_{ik} \leq n \cdot y_k, \quad \forall k \in \{1, \dots, n\} \\
 & y_k, x_{ik} \in \{0, 1\}, \quad \forall i \in V, k \in \{1, \dots, n\}
 \end{aligned}$$

La función objetivo se define así, para asegurar que se agotan antes los colores más bajos. La primera restricción garantiza que se asigna exactamente un color a cada vértice, la segunda previene compatibilidad de colores de los extremos de aristas y la tercera garantiza que si algún color es usado, entonces la variable  $y_k$  vale 1. El valor óptimo de la función objetivo asegura que se agotan antes los colores más bajos.

El modelo de programación binaria asociado al ejemplo 1.1 es:

$$\begin{aligned}
 \text{Min} \quad & y_1 + 2y_2 + 3y_3 + 4y_4 \\
 \text{s.a} \quad & x_{11} + x_{12} + x_{13} + x_{14} = 1 \\
 & x_{21} + x_{22} + x_{23} + x_{24} = 1 \\
 & x_{31} + x_{32} + x_{33} + x_{34} = 1 \\
 & x_{41} + x_{42} + x_{43} + x_{44} = 1 \\
 & x_{1k} + x_{2k} \leq 1 \quad \forall k \\
 & x_{1k} + x_{3k} \leq 1 \quad \forall k \\
 & x_{1k} + x_{4k} \leq 1 \quad \forall k \\
 & x_{2k} + x_{3k} \leq 1 \quad \forall k \\
 & x_{3k} + x_{4k} \leq 1 \quad \forall k \\
 & x_{1k} + x_{2k} + x_{3k} + x_{4k} \leq 4y_k \quad \forall k
 \end{aligned}$$

cuyo valor mínimo es 6 y las variables que no valen cero son  $x_{11} = 1$ ,  $x_{22} = 1$ ,  $x_{33} = 1$ ,  $x_{42} = 1$

### 1.3.3 Modelo de partición

El modelo de partición que resuelve el *PCM*, Korman [43], se basa en identificar todos los conjuntos independientes maximales del grafo  $G = (V, E)$ .

Un conjunto  $S \subset V$  es independiente cuando  $\{i, j\} \notin E \quad \forall i, j \in S$ . Obviamente, las clases de color asociadas a cualquier función de coloración son conjuntos independientes. El conjunto independiente es maximal cuando no se puede incluir en otro conjunto independiente.

Sean  $S_1, S_2, \dots, S_t$  todos los conjuntos independientes maximales de  $G$ , que están caracterizados por los siguientes parámetros:

$$e_{ij} = \begin{cases} 1, & \text{si el vértice } i \in S_j \\ 0, & \text{en otro caso} \end{cases} \quad \forall i \in \{1, \dots, n\} \quad \forall j \in \{1, \dots, t\}$$

Las variables de decisión son

$$w_j = \begin{cases} 1, & \text{si } S_j \text{ es una clase de color} \\ 0, & \text{en otro caso} \end{cases} \quad \forall j \in \{1, \dots, t\}$$

El programa es

$$\begin{aligned} \min \quad & \sum_{j=1}^t w_j \\ \text{s.a.} \quad & \sum_{j=1}^t e_{ij} \cdot w_j = 1 \quad \forall i \in \{1, \dots, n\} \\ & w_j \in \{0, 1\}, \quad \forall j \in \{1, \dots, t\} \end{aligned}$$

Se quiere encontrar el mínimo número de conjuntos independientes que formen una partición de  $V$ .

*El grafo del ejemplo 1.1, tiene los conjuntos independientes maximales  $S_1 = \{1\}$ ,  $S_2 = \{2, 4\}$  y  $S_3 = \{3\}$ . El modelo de partición es:*



$$\begin{aligned}
 \text{Min } f(\mathbf{x}) &= \sum_{j=1}^3 w_j \\
 \text{s.a. } w_1 &= 1 \\
 w_2 &= 1 \\
 w_3 &= 1 \\
 w_2 &= 1
 \end{aligned}$$

El mejor valor de la función objetivo es 3 y  $w_1 = 1$ ,  $w_2 = 1$ ,  $w_3 = 1$ ; el resto de las variables vale cero.

El determinar todos los conjuntos independientes maximales supone una dificultad computacional muy seria para la utilidad real de este modelo.

### 1.3.4 Modelo de asignación

Un problema de tipo asignación se puede plantear de la siguiente manera: Dados  $n$  items y  $m$  recursos, determinar una asignación de los items a los recursos, optimizando una función objetivo y satisfaciendo un conjunto de restricciones adicionales.

El *PCM* se puede formular como un problema de asignación, ver Costa y Hertz [17]. Dado un grafo  $G = (V, E)$  donde  $|V| = n$  y  $|E| = m$ , los items son los vértices y los recursos son los colores. Ya que siempre es posible colorear cualquier grafo  $G$  con  $n$  colores, se hace  $m = n$ . Las variables de decisión son

$$x_{ik} = \begin{cases} 1 & \text{si el vértice } i \text{ recibe el color } k \\ 0 & \text{en otro caso} \end{cases} \quad \forall i, k \in \{1, \dots, n\}$$

La función objetivo es

$$\text{Min } f(\mathbf{x}) = \sum_{k=1}^n k \cdot \delta \left( \sum_{i=1}^n x_{ik} \right)$$

sujeito a

$$G_k(\mathbf{x}) = \sum_{\{i,j\} \in E} x_{ik} \cdot x_{jk} \leq 0, \quad 1 \leq k \leq n$$

donde

$$\delta(z) = \begin{cases} 1 & \text{si } z > 0 \\ 0 & \text{en otro caso} \end{cases}$$

La función objetivo suma los números asociados con los colores usados en la coloración de  $\mathbf{x}$ . De esta manera una coloración mínima usa necesariamente todos los colores consecutivos entre 1 y  $\chi(G)$ . Las restricciones  $G_j(\mathbf{x})$  impiden que las aristas tengan el mismo color en sus extremos. Obsérvese que este es un modelo no lineal.

El modelo de asignación para el ejemplo 1.1, es:

$$\begin{aligned} \text{Min } f(\mathbf{x}) = & \delta(x_{11} + x_{21} + x_{31} + x_{41}) + 2\delta(x_{12} + x_{22} + x_{32} + x_{42}) + \\ & + 3\delta(x_{13} + x_{23} + x_{33} + x_{43}) + 4\delta(x_{14} + x_{24} + x_{34} + x_{44}) \\ & \sum_{j=1}^k x_{1j} \cdot x_{2j} & \leq 0 \\ & \sum_{j=1}^k x_{1j} \cdot x_{3j} & \leq 0 \\ & \sum_{j=1}^k x_{1j} \cdot x_{4j} & \leq 0 \\ & \sum_{j=1}^k x_{2j} \cdot x_{3j} & \leq 0 \\ & \sum_{j=1}^k x_{3j} \cdot x_{4j} & \leq 0 \end{aligned}$$

las variables diferentes de cero son :  $x_{11} = 1, x_{22} = 1, x_{33} = 1$  y  $x_{44} = 1$ ; el valor óptimo de la función objetivo es 6.

### 1.3.5 Modelo de asignación cuadrática

El Problema de Coloración Mínima también se puede plantear como un caso especial del Problema de Asignación Cuadrática, *PAC* para abreviar, que consiste en determinar el vector  $\mathbf{x} \in \mathbb{R}^n$  tal que:

$$PAC \left\{ \begin{array}{l} \text{Min } \mathbf{c}^t \mathbf{x} + \frac{1}{2} \mathbf{x}^t Q \mathbf{x} \\ \mathbf{A} \mathbf{x} = \mathbf{b} \\ \mathbf{x} \geq \mathbf{0} \end{array} \right.$$

siendo  $\mathbf{c} \in \mathbb{R}^n$ ,  $\mathbf{b} \in \mathbb{R}^m$ ,  $Q$  una matriz cuadrada simétrica  $n \times n$  y  $A$  una matriz  $m \times n$

El caso especial fue propuesto por Pardalos y Wolkowics [52] como sigue: Dado un número entero  $c$ , el valor óptimo de la función objetivo del siguiente problema es cero si y solo si  $G$  es  $c$ -coloreable.

$$\begin{aligned} \text{Min} \quad & \frac{1}{2} \sum_{k=1}^c \sum_{(i,j) \in E} x_{ik} x_{jk} \\ \text{s.a} \quad & \sum_{k=1}^c x_{ik} = 1, \quad i = 1, \dots, n \\ & x_{ik} \in \{0, 1\}, \quad \forall i, k \end{aligned}$$

El vector  $\mathbf{x}$  identifica la función de coloración  $C$  que resuelve el *PCM*:

$$x_{ik} = \begin{cases} 1 & C(i) = k \\ 0 & C(i) \neq k \end{cases} \quad \forall i \in \{1, \dots, n\} \quad \forall k \in \{1, \dots, n\}$$

donde el número de colores del grafo se ha acotado de forma natural por  $n$  al verificarse siempre que  $\chi(G) \leq n$ .

Sabiendo que el grafo del ejemplo 1.1 es 3-coloreable, el modelo de asignación cuadrática asociado es el siguiente:

$$\begin{aligned} \text{Min} f(\mathbf{x}) = & x_{11} \cdot x_{21} + x_{11} \cdot x_{31} + x_{11} \cdot x_{41} + x_{21} \cdot x_{31} + x_{31} \cdot x_{41} + \\ & + x_{12} \cdot x_{22} + x_{12} \cdot x_{32} + x_{12} \cdot x_{42} + x_{22} \cdot x_{32} + x_{32} \cdot x_{42} + \\ & + x_{13} \cdot x_{23} + x_{13} \cdot x_{33} + x_{13} \cdot x_{43} + x_{23} \cdot x_{33} + x_{33} \cdot x_{43} \\ & x_{11} + x_{12} + x_{13} & = 1 \\ & x_{21} + x_{22} + x_{23} & = 1 \\ & x_{31} + x_{32} + x_{33} & = 1 \\ & x_{41} + x_{42} + x_{43} & = 1 \end{aligned}$$

como ya se dijo, al ser el grafo 3-coloreable el *valor de la función objetivo es cero* y las variables  $x_{11} = 1$ ,  $x_{22} = 1$ ,  $x_{33} = 1$ ,  $x_{42} = 1$  y las demás valen todas cero.

## 1.4 Otros problemas de coloración

De los problemas de optimización que pueden modelarse como un problema de coloración de grafos se han obtenido generalizaciones muy interesantes y Pardalos et al. [53], sugieren que se pueden clasificar en dos grupos, que les llaman generalizaciones *internas* y *externas*.

**Generalizaciones internas** En este grupo se relaja la restricción de que los conjuntos de vértices con un mismo color deben ser conjuntos independientes.

Dos generalizaciones de este tipo son la  $(c, d)^*$ -coloración y la  $(c, k)^-$ -coloración.

Un grafo  $G$  es  $(c, d)^*$ -coloreable cuando admite  $c$  colores distintos, con la propiedad de que cualquier vértice de  $G$  puede ser adyacente hasta con  $d$  vértices que tengan el mismo color.

Cuando a una clase de color se le permite que tenga una cadena con no más de  $k$  aristas y bajo esta relajación  $G$  se puede colorear con  $c$  colores entonces  $G$  es  $(c, k)^-$ -coloreable.

*Nótese que si en los casos anteriores  $d = 0$  y  $k = 0$  respectivamente, entonces se tiene el problema de coloración mínima.*

En [16] y en [35] se estudian estas coloraciones, pero no se aporta algún ejemplo que justifique el interés práctico.

**Generalizaciones externas** Aquí cada clase de color es aún un conjunto independiente, sin embargo, se imponen relaciones adicionales entre las clases de color.

En estas generalizaciones a cada vértice se le asigna un conjunto de colores, mediante una función  $L$ , con la propiedad  $\lambda$ . A la función  $L$  se le llama asignación  $\lambda$  – conjunto. El concepto de *conjunto coloración* o  $\lambda$  – coloración fue introducido por Roberts en [60], y consiste en asignar un conjunto  $L(i)$  de colores a cada vértice  $i$  de tal manera que para todo  $\{i, j\} \in E$  y  $i \neq j$  se cumple que  $L(i) \cap L(j) = \emptyset$ .

La propiedad  $\lambda$  puede ser que cada conjunto  $L(i)$  tenga un número determinado de elementos. *Notar que si en este tipo de coloraciones cada  $L(i)$  verifica  $L(i) \leq 1 \quad \forall i \in V$ , entonces la  $\lambda$  – coloración coincide con la función de coloración.*

Dependiendo de la definición de  $\lambda$  los problemas de conjunto coloración pueden ser diferentes. Se han estudiado muchas variaciones, tales como:

- $k$  L-lista  $\tau$  coloración de grafos de De Luca Cardillo y Mione [22]. La característica de este problema es la existencia de un conjunto  $\tau$ , llamado conjunto de asignaciones incompatibles; para cada asignación de un color a un vértice  $i$ , debe verificarse que el color pertenece a la lista  $L(i)$  de colores asociados al vértice y cualesquiera dos colores asignados a dos vértices adyacentes no pertenezcan al conjunto  $\tau$ . La longitud de la lista  $L(i)$ , para cada  $i$  de  $G$ , debe ser menor o igual a  $k$ , donde  $k$  es el número de colores. Está claro que la condición necesaria para obtener la  $k$  L-lista  $\tau$  coloración es que  $k \geq \chi(G)$ .

Problema de andenes para trenes (*De Luca Cardillo y Mione [22]*). Consiste en asignar trenes que llegan o salen de una estación a los andenes disponibles.

El problema lo resuelven coloreando un grafo, para cuya definición utilizan lo siguiente. Sea  $T = \{1, 2, \dots, n\}$  el conjunto de trenes que se van a asignar, donde a cada tren  $i \in T$ , se le asocia un intervalo  $[TL(i), TS(j)]$ .  $TL(i)$  y  $TS(j)$  indican los tiempos de llegada y de salida de los trenes  $i$  y  $j$  respectivamente. Sea  $IMIN$  un entero no negativo que representa el intervalo mínimo de seguridad que debe respetarse entre cada salida y cada llegada en el mismo andén.

Se define el grafo  $G = (V, E)$ , donde  $V$  es el conjunto ordenado de los  $n$  trenes de  $T$ , los trenes son ordenados de tal manera que si  $i$  llega a la estación después que  $j$ , entonces  $i > j$ , *i.e.*,  $TL(i) > TL(j)$ ; hay incompatibilidad entre dos trenes  $i$  y  $j$ , es decir,  $(i, j) \in E$ , cuando llegan en un intervalo de tiempo menor que  $IMIN$ .

Dados  $k$  andenes disponibles, a cada tren  $i \in T$  se le asocia una lista  $L(i)$  de

posibles andenes, con  $|L(i)| \leq k$ , teniendo en cuenta la capacidad de los andenes y las direcciones de llegada y salida de los trenes. El andén a asignar al tren  $i$  se escoge de esta lista coloreando a  $G$ .

- Sea  $G$  un grafo y  $T$  un conjunto de enteros no negativos. Una  $T$ -coloración de  $G$  es una asignación de un entero positivo  $f(i)$  a cada vértice  $i$  de  $G$  tal que si  $i$  y  $j$  están unidos por una arista de  $G$ , entonces  $|f(i) - f(j)|$  no está en  $T$ . Roberts [60], comenta que las  $T$ -Coloraciones fueron introducidas por Hale con el problema de asignación de canales en comunicaciones.

Problema de mantenimiento de flotas (Opsut y Roberts [51]). Los vértices son los vehículos que llegan para mantenimiento regular y hay una arista entre dos vehículos si son programados en un puesto de mantenimiento en periodos de tiempo que se intersecan. Se busca asignar en el puesto de mantenimiento un periodo de tiempo  $L(i)$  a cada vehículo  $i$ , de tal manera que los vehículos tengan asignados intervalos que no se traslapen.

Problema de asignación de frecuencia a radios móviles (Opsut y Roberts [51]). Los vértices de  $G$  son las zonas en que operan los vehículos con radios móviles y hay una arista entre dos zonas si los vehículos con radios móviles operando en dichas zonas pueden interferir entre sí. Se busca asignar un conjunto  $L(i)$  de frecuencias permisibles a cada zona  $i$  de tal manera que dos zonas con posibles interferencias tengan conjuntos disjuntos.

- Intervalo de coloración de grafos ponderados de Čangalović y Schreuder [8], que es una versión del mismo concepto de Golumbic [29], se dice que un grafo es ponderado si cada vértice del grafo tiene asociado un número entero positivo. Sea  $G = (V, E, \mathbf{p})$  un grafo, donde  $A$  es el conjunto de vértices,  $E$  el de aristas y  $\mathbf{p}$  es un vector que contiene los pesos. Un grafo tiene un intervalo  $k$ -coloración si a cada vértice  $i \in \{1, \dots, n\}$  se le asignan  $p_i$  enteros distintos y consecutivos de un conjunto

$\{1, 2, \dots, k\}$ , de tal manera que vértices adyacentes no tengan un color en común.

Problema en que grupos de alumnos con un programa determinado de cursos, un conjunto de profesores, un conjunto de periodos de tiempo y un conjunto de clases que consisten de una combinación específica de profesores y grupos. Cada clase tiene una duración en horas o periodos necesarios.

El problema consiste en encontrar un horario factible tal que todas las clases sean asignadas a un conjunto de periodos y cada grupo tenga a lo más una clase en cada periodo. Las clases pueden tener duraciones diferentes, deben ser consecutivas y cada profesor puede enseñar en más de un grupo. El conjunto de periodos de tiempo consiste de horas consecutivas de clase por día. Al horario diario de clase se le puede asociar un grafo ponderado, cada clase se representa por un vértice cuyo peso es la duración de la clase y hay una arista entre dos vértices si las clases correspondientes tienen un grupo o un profesor en común.

En Pardalos *et al.* [53] se comentan algunas interpretaciones de las generalizaciones, de los nombres que reciben las coloraciones correspondientes y se dan amplias referencias sobre dichas generalizaciones.

# Capítulo 2

## Problema de Coloración Robusta

### 2.1 Justificación del problema

Algunos problemas de planificación del tiempo o de recursos se pueden plantear como problemas de coloración de los vértices de un grafo, frecuentemente tratando de minimizar el número de colores a utilizar. En [5], [13] y [14] se presenta una descripción muy completa de las técnicas que se han utilizado para tratar de encontrar buenas soluciones a diversas variantes del problema y en [4] y [6] se presenta la gran mayoría de esas técnicas.

Los vértices representan los elementos a programar y cada arista identifica la incompatibilidad entre los elementos correspondientes. Esta incompatibilidad representa el conflicto entre recursos cuando no puedan ser compartidos los elementos a programar.

Una coloración válida identifica una asignación de recursos compatibles. En estos problemas es común que el objetivo sea utilizar el mínimo número de colores, de tal manera que a cada par de elementos a programar que no pueden compartir el mismo recurso se les asigne un color diferente. Cada clase de color contendrá los elementos entre los que no hay conflicto y que pueden compartir el recurso en cuestión.

En algunas circunstancias, el recurso a minimizar con el planteamiento de mínimo número de colores no es crítico, sino que interesa que una solución al problema sea estable,



en el sentido de que al añadir o cambiar aristas al grafo, la coloración continúe siendo válida. Estas consideraciones muestran que el problema de coloración mínima es un modelo muy restrictivo para este tipo de problemas.

Se presenta un nuevo problema de coloración de grafos que contempla las extensiones mencionadas. En este problema, cada planificación se hará teniendo en cuenta un cierto número de recursos, sabiendo que existen programaciones alternativas —coloraciones del grafo asociado— más flexibles y que puedan ser valoradas con otros criterios.

A continuación se ilustran este tipo de problemas, usando como base los ejemplos de problemas de coloración mínima enunciados en la sección 1.2.

## 2.2 Programación de Exámenes

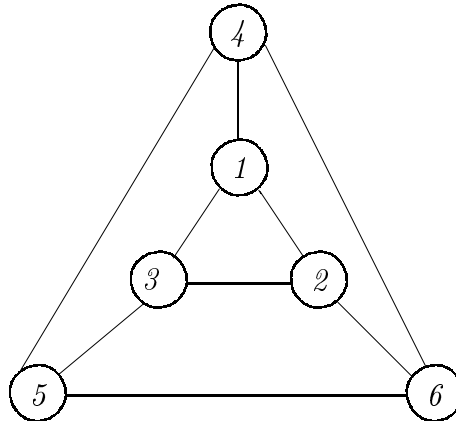
Un caso concreto de grafo de incompatibilidad del problema planteado en la sección 1.2.1 es el siguiente.

**Ejemplo 2.1** Sea  $G = (V, E)$  definido por  $V = \{1, \dots, 6\}$  y con matriz de adyacencia:

$$B_G = \begin{pmatrix} 0 & & & & & \\ 1 & 0 & & & & \\ 1 & 1 & 0 & & & \\ 1 & 0 & 0 & 0 & & \\ 0 & 0 & 1 & 1 & 0 & \\ 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

*Se omite la matriz triangular superior al ser un grafo no dirigido.*

*La representación del grafo  $G$  es la siguiente:*



Si el objetivo es minimizar la duración del periodo de exámenes se plantea el problema de coloración mínima de los vértices del grafo de incompatibilidades  $G$  previamente definido.

*Para este caso concreto, se puede comprobar que el número cromático es 3, es decir  $\chi(G) = 3$  y los exámenes se pueden realizar en 3 días.*

*Una coloración mínima, es decir, la asignación de cada examen a cada uno de los días —colores— es la siguiente:*

$$C(1) = 1, C(2) = 2, C(3) = 3, C(4) = 2, C(5) = 1, C(6) = 3$$

Si se considera que el número de días que duran los exámenes es sólo una restricción y no la función objetivo a minimizar, cualquier coloración mínima será una solución del problema sin más que exigir que el número cromático del grafo de incompatibilidad no supere el número disponible de días .

No sólo la coloración mínima, sino cualquier coloración que no utilice más colores que los días permitidos proporcionará un calendario válido para realizar los exámenes.

Un criterio que permitiría ordenar estas coloraciones podría ser el siguiente: Valorar las distintas coloraciones por la probabilidad de que sigan siendo válidas una vez añadida una arista que no existía en  $E$ . Esta modificación de la topología del grafo  $G$  tiene sentido al considerar la posibilidad de que, una vez fijado el calendario de exámenes, algún alumno cambie de asignatura provocando así la incompatibilidad de dos asignaturas que antes eran compatibles.

Si se conoce la probabilidad de que una pareja de vértices representando a las asignaturas y verificando que  $\{i, j\} \notin E$  sea incluida en el grafo ampliado, tiene sentido plantearse el problema anterior.

En lo que sigue, se denotará por  $\overline{E}$  al conjunto de aristas complementarias de  $E$ , es decir,

$$\{i, j\} \in \overline{E} \iff \{i, j\} \notin E$$

y  $\overline{G} = (V, \overline{E})$  será el grafo complementario de  $G$ .

*Volviendo al ejemplo 2.1, se supone que la probabilidad de que a algún alumno le coincidan las asignaturas  $i$  y  $j$ , cuando  $\{i, j\} \notin E$ , i.e., que se agregue la arista  $\{i, j\}$  al grafo de incompatibilidades, depende de los valores  $n_i$  y  $n_j$ , siendo  $n_i$  el número de alumnos que escogieron la asignatura  $i \in V$ .*

*Sean 50 los alumnos del Centro Universitario que al escoger 2 de las 6 asignaturas, proporcionan los siguientes datos:*

$$n_1 = 5, n_2 = 30, n_3 = 10, n_4 = 30, n_5 = 20, n_6 = 5$$

*Si las probabilidades  $p_{ij}$ , de añadir las aristas complementarias del grafo  $\{i, j\} \notin E$  aumentan con  $n_i$  y  $n_j$ , según se especifica en la siguiente tabla, donde se han determinado las probabilidades normalizadas a partir de los productos  $n_i \times n_j$*

$\{i, j\} \in \bar{E}$	$n_i \times n_j$	$pr_{ij}$
$\{1, 5\}$	100	0.0506
$\{1, 6\}$	25	0.0127
$\{2, 4\}$	900	0.4557
$\{2, 5\}$	600	0.3038
$\{3, 4\}$	300	0.1519
$\{3, 6\}$	50	0.0253

y hay 3 días para programar los exámenes, el problema se puede plantear de la siguiente manera:

Obtener una 3-coloración óptima en el sentido de minimizar la probabilidad de añadir una arista con los dos extremos igualmente coloreados.

El problema de coloración así planteado busca una coloración que se mantenga válida cuando se añada alguna arista. Suponiendo la independencia de estos sucesos, es decir, el cambio de asignatura, la probabilidad de que se añada una arista al grafo de incompatibilidades de asignaturas que no invalide la coloración  $C$  es:

$$\prod_{\{i,j\} \in \bar{E}, C(i)=C(j)} (1 - pr_{ij})$$

En el ejemplo 2.1 de referencia, al considerar la coloración  $C$  esta probabilidad es:

$$(1 - pr_{15}) \cdot (1 - pr_{24}) \cdot (1 - pr_{36}) = (1 - 0.0506) \cdot (1 - 0.4557) \cdot (1 - 0.0253) = 0.5037$$

Si se considera la coloración  $C'$ :

$$C'(1) = 1, C'(2) = 2, C'(3) = 3, C'(4) = 3, C'(5) = 2, C'(6) = 1$$

la probabilidad de que no se invalide la coloración al agregar una arista es:

$$(1 - pr_{16}) \cdot (1 - pr_{25}) \cdot (1 - pr_{34}) = (1 - 0.0127) \cdot (1 - 0.3038) \cdot (1 - 0.1519) = 0.5829$$

*Comparando ambas probabilidades, con el criterio de maximizar la probabilidad de que la coloración sea válida, será preferible  $C'$  a  $C$ .*

*Si se dispone de 4 días para programar los exámenes, al considerar la coloración:*

$$C''(1) = 2, C''(2) = 4, C''(3) = 1, C''(4) = 3, C''(5) = 2, C''(6) = 1$$

*la probabilidad de que se mantenga válida frente a la posible adición de alguna arista es:*

$$(1 - pr_{15}) \cdot (1 - pr_{36}) = (1 - 0.0506) \cdot (1 - 0.0253) = 0.9254$$

*Como era de esperar, al aumentar el número de días disponibles, aumenta la probabilidad de mantener válida la coloración obtenida al añadir una arista.*

## 2.3 Determinación de Conglomerados

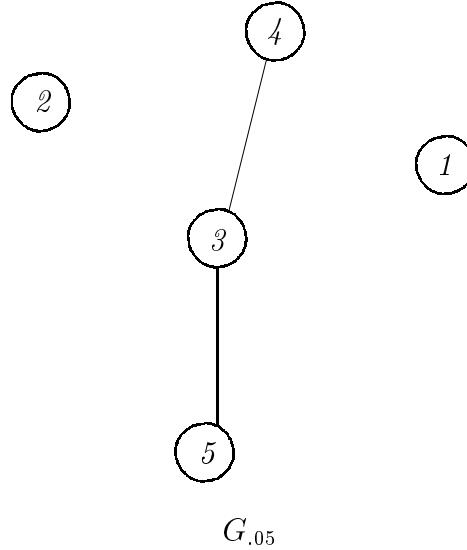
El siguiente ejemplo ilustra un problema de conglomerados del tipo introducido en la sección 1.2.2.

**Ejemplo 2.2** *Considérense el siguiente conjunto de 5 elementos  $\{1, 2, 3, 4, 5\}$  y la distancia  $d$  especificada por la siguiente matriz triangular:*

$$\begin{pmatrix} * & .01 & .02 & .05 & .04 \\ & * & .04 & .03 & .04 \\ & & * & .06 & .07 \\ & & & * & .03 \\ & & & & * \end{pmatrix}$$

*Para distintos valores de  $\alpha$  se obtienen los siguientes grafos y coloraciones:*

Si  $\alpha = .05$  entonces  $G_\alpha$  tiene dos aristas:  $\{3, 4\}$  y  $\{3, 5\}$ , según se puede comprobar con el grafo adjunto.



El número cromático de  $G_{.05}$  es 2, que se puede asociar a la coloración

$$C_1^2(1) = 1 \quad C_1^2(2) = 1 \quad C_1^2(3) = 2 \quad C_1^2(4) = 1 \quad C_1^2(5) = 1$$

por lo que con 2 clases se define un conglomerado formado por

$$V_{C_1^2}(1) = \{1, 2, 4, 5\} \quad V_{C_1^2}(2) = \{3\}$$

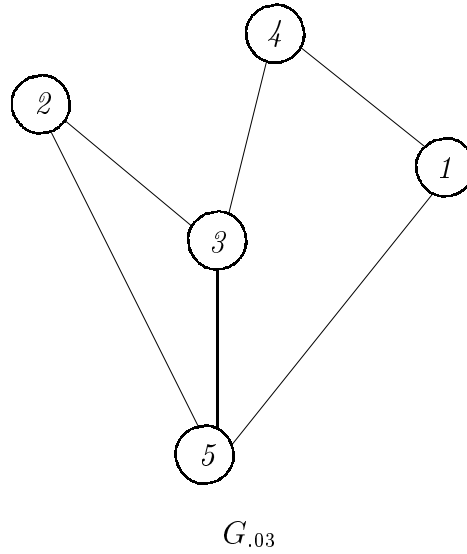
Análogamente se puede definir esta otra 2-coloración

$$C_2^2(1) = 1 \quad C_2^2(2) = 1 \quad C_2^2(3) = 1 \quad C_2^2(4) = 2 \quad C_2^2(5) = 2$$

que tiene asociado el conglomerado formado por

$$V_{C_2^2}(1) = \{1, 2, 3\} \quad V_{C_2^2}(2) = \{4, 5\}$$

Al disminuir el umbral para obtener  $\alpha = .03$ , se deben añadir a las aristas anteriores las aristas  $\{1, 4\}$ ,  $\{1, 5\}$ ,  $\{2, 3\}$  y  $\{2, 5\}$ . Se obtiene así el grafo  $G_{.03}$ .



con número cromático 3, por lo que el número mínimo de clases del conglomerado asociado ha aumentado.

Una coloración válida sería

$$C_1^3(1) = 1 \quad C_1^3(2) = 2 \quad C_1^3(3) = 1 \quad C_1^3(4) = 2 \quad C_1^3(5) = 3$$

que tiene asociado el conglomerado formado por

$$V_{C_1^3}(1) = \{1, 3\}, \quad V_{C_1^3}(2) = \{2, 4\}, \quad V_{C_1^3}(3) = \{5\}$$

Se puede comprobar que para  $\alpha = .02$ , al añadir las aristas  $\{2, 4\}$  y  $\{4, 5\}$ , el número cromático del grafo  $G_{.02}$  es 4. La coloración asociada sería

$$C_1^4(1) = 2 \quad C_1^4(2) = 2 \quad C_1^4(3) = 3 \quad C_1^4(4) = 4 \quad C_1^4(5) = 1$$

Obviamente, en el caso límite  $\alpha = 0$ , el grafo  $G_\alpha$  es un grafo completo y hay tantas clases como elementos.

En este ejemplo puede observarse que el número cromático del grafo  $G_\alpha$  identifica el menor número de clases que se pueden formar con la disimilitud  $d$  y el umbral  $\alpha$ . Este número cromático se obtiene con una coloración válida del citado grafo.

Volviendo al ejemplo de conglomerados, fijando el número de colores  $c = 2$ , la coloración  $C_1^2$  no es válida para el valor del umbral  $\alpha = 0.04$ , puesto que los vértices 1 y 4 tienen asignado el mismo color.

Sin embargo, manteniendo el mismo número de colores, la coloración  $C_2^2$  sigue siendo válida si  $\alpha = 0.04$ . De alguna forma, esta coloración es mejor que la anterior al permitir una mayor disminución del umbral  $\alpha$ .

En el Problema de Conglomerados, fijado el umbral  $\alpha$  y determinada una coloración válida  $C^c$  de  $G_\alpha$ , se define el valor mínimo  $\alpha(C^c) \in [0, 1]$  como el valor del umbral tal que  $C^c$  sigue siendo una coloración válida para  $G_{\alpha'}$  para cualquier  $\alpha' \geq \alpha(C^c)$ .

Según lo anterior,  $\alpha(C_1^2) = 0.05$  y  $\alpha(C_2^2) = 0.04$ .

Se puede comprobar que la coloración  $C_1^3$  tiene asociado un valor  $\alpha(C_1^3) = 0.03$ .

Tiene sentido plantearse el siguiente problema de Conglomerados: fijados el número máximo de clases  $c$  y el umbral  $\alpha$ , ¿cuál es aquella coloración con  $c$  colores  $C_*^c : V \rightarrow \{1, \dots, c\}$  con un valor crítico  $\alpha(C_*^c)$  mínimo?. Tal coloración debe verificar:

$$\alpha^c = \alpha(C_*^c) = \min_{C^c} \alpha(C^c)$$

En otras palabras,  $\alpha^c$  es el menor valor de  $\alpha$  con el que la coloración  $C^c$  es una coloración válida para  $G_\alpha^c$ . No es difícil ver que  $\alpha^c$  disminuye al aumentar el parámetro  $c$ . En el caso límite, cuando  $c = n$ , se verifica que  $\alpha^n = 0$ .

En el ejemplo 2.2, fijando  $\alpha = 0.05$ , se puede comprobar que

$$\alpha^2 = 0.04$$

siendo  $C_*^2 = C_2^2$ .

Con  $c = 3$ , se obtiene

$$\alpha^3 = 0.03$$



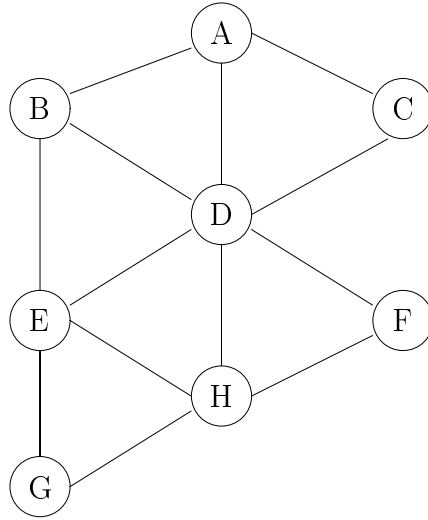


Figura 2.1: Grafo de Regiones

alcanzándose este valor mínimo con  $C_*^3 = C_2^3$ :

$$C_2^3(1) = 1 \quad C_2^3(2) = 1 \quad C_2^3(3) = 2 \quad C_2^3(4) = 3 \quad C_2^3(5) = 3$$

## 2.4 Coloración de un mapa

Se presenta el siguiente ejemplo de problema de coloración de un mapa introducido en la sección 1.2.3.

**Ejemplo 2.3** *Se quiere pintar el mapa cuyo grafo asociado se ha representado en la figura 2.1.*

*Sean las siguientes 4 – coloraciones*

	A	B	C	D	E	F	G	H
$C_1^4$	3	2	2	1	4	2	2	3
$C_2^4$	3	2	4	1	3	2	1	4

que dan lugar a las clases de color

$$V_{C_1^4}(1) = \{D\}, \quad V_{C_1^4}(2) = \{B, C, F, G\}, \quad V_{C_1^4}(3) = \{A, H\}, \quad V_{C_1^4}(4) = \{E\}$$

y

$$V_{C_2^4}(1) = \{D, G\}, \quad V_{C_2^4}(2) = \{B, F\}, \quad V_{C_2^4}(3) = \{A, E\}, \quad V_{C_2^4}(4) = \{C, H\}$$

respectivamente.

Con la coloración  $C_2^4$  las clases de color son homogéneas en lo que se refiere al número de elementos de las mismas, esto hará preferible a la coloración  $C_2^4$  en lugar de la  $C_1^4$ .

## 2.5 Planteamiento del Problema

Dados un grafo  $G = (V, E)$  con  $|V| = n$  y  $|E| = m$  y un entero  $c > 0$ , sea  $C^c$  una  $c$ -coloración, es decir:

$$C^c : V \longrightarrow \{1, 2, \dots, c\}$$

verificando

$$C^c(i) \neq C^c(j) \quad \forall \{i, j\} \in E$$

**Observación 2.1** *En todos los desarrollos posteriores, se supone que cualquier  $c$ -coloración es estricta en el sentido de agotar los  $c$  colores.*

$$V_C(r) \neq \emptyset \quad \forall r \in \{1, \dots, c\}$$

En los ejemplos anteriores se ha justificado el interés de valorar la coloración de un grafo en la medida en que siga siendo una coloración válida al añadir nuevas aristas al grafo. Se introduce así el siguiente concepto.

**Definición 2.1** *Dados el grafo  $G$  y conocida la familia de penalizaciones  $\{p_{ij} \geq 0, \{i, j\} \in \overline{E}\}$ , se define el grado de rigidez de la  $c$ -coloración  $C^c$ , y se denota por  $R(C^c)$  a la suma de las penalizaciones de las aristas complementarias cuyos extremos están igualmente coloreados:*

$$R(C^c) \equiv \sum_{\{i,j\} \in \overline{E}, C^c(i)=C^c(j)} p_{ij}$$

*En el ejemplo 2.1, hay que considerar como penalización,  $p_{ij}$ , de cada arista complementaria  $\{i, j\} \in \overline{E}$  el logaritmo neperiano del complemento de la probabilidad de que se incorpore cada arista cambiado de signo, es decir*

$$p_{ij} = -\ln(1 - pr_{ij})$$

*Con esta penalización, el grado de rigidez de la  $c$ -coloración  $C$  es*

$$R(C^c) \equiv \sum_{\{i,j\} \in \overline{E}, C^c(i)=C^c(j)} -\ln(1 - pr_{ij}) = -\ln \left( \prod_{\{i,j\} \in \overline{E}, C^c(i)=C^c(j)} (1 - pr_{ij}) \right)$$

*Por consiguiente, el grado de rigidez de una  $c$ -coloración en el ejemplo 2.1 coincide con el logaritmo —cambiado de signo— de la probabilidad de que la coloración sea válida, al añadir una arista .*

*En el ejemplo 2.2, sin embargo, la interpretación del grado de rigidez no es tan directa, puesto que la penalización de las aristas complementarias debe considerar el orden de los valores de las disimilitudes. Recordar que en este problema se añaden aristas al grafo  $G_\alpha$  al disminuir el valor de  $\alpha$ .*

*El siguiente procedimiento es válido si las disimilitudes toman un número finito de valores, lo que en la práctica no supone una restricción muy fuerte. Por consiguiente, sea  $\{d^1, d^2, \dots, d^k\}$  el conjunto de disimilitudes de  $\overline{E}$ , de forma que*

$$d_{ij} \in \{d^1, d^2, \dots, d^k\} \quad \forall \{i, j\} \in \overline{E}_\alpha$$

verificando

$$d^1 < d^2 < \dots < d^k$$

Fijado un umbral  $\alpha$ , la penalización al colorear igualmente los extremos de una arista complementaria  $\{i, j\} \in \overline{E}_\alpha$  con una determinada disimilitud asociada debe verificar que su penalización supere la suma de un número arbitrario de las penalizaciones de aristas complementarias con una disimilitud menor.

Sea  $\overline{m} = |\overline{E}|$ , una forma de garantizar la propiedad anterior es a través de la siguiente penalización:

$$p_{ij} = (\overline{m})^{s-1} \quad \text{si } d_{ij} = d^s \quad \forall \{i, j\} \in \overline{E} \quad s \in \{1, \dots, k\}$$

Concretamente, si se considera  $k = 2$  y la situación límite

$$d_{ij} = \begin{cases} d^1 & \{i, j\} \neq \{i_0, j_0\} \\ d^2 & \{i, j\} = \{i_0, j_0\} \end{cases} \quad \forall \{i, j\} \in \overline{E}$$

es decir, si de las  $\overline{m}$  aristas complementarias la arista  $\{i_0, j_0\}$  tiene disimilitud  $d_{i_0 j_0}$  igual a  $d^2$  y las  $\overline{m} - 1$  restantes tienen una disimilitud igual a  $d^1$ , entonces la arista  $\{i_0, j_0\}$  tiene penalización  $p_{i_0 j_0} = \overline{m}$  y todas las demás tienen penalización  $p_{ij} = \overline{m}^0 = 1$ , de forma que

$$\sum_{\{i, j\} \in \overline{E}/d_{ij}=d^1} p_{ij} = (\overline{m} - 1) \cdot 1 < (\overline{m})^{2-1} = \overline{m} = p_{i_0 j_0}$$

por lo que es prioritario el evitar colorear los vértices  $i_0$  y  $j_0$  con el mismo color para conseguir una rigidez menor.

En el ejemplo 2.2, fijado el umbral  $\alpha = 0.05$  y considerando que las disimilitudes de los elementos están incluidas en el conjunto

$$\{0.01, 0.02, 0.03, 0.04, 0.05\}$$

se concluye que  $k = 5$ , siendo

$$d^1 = 0.01 \quad d^2 = 0.02 \quad d^3 = 0.03 \quad d^4 = 0.04 \quad d^5 = 0.05$$

Al ser  $\overline{m} = 8$ , se tiene que las penalizaciones son:

ARISTA	$d_{ij}$	$d^s$	$p_{ij}$
$\{1, 2\}$	.01	$d^1$	1
$\{1, 3\}$	.02	$d^2$	8
$\{1, 4\}$	.05	$d^5$	4096
$\{1, 5\}$	.04	$d^4$	512
$\{2, 3\}$	.04	$d^4$	512
$\{2, 4\}$	.03	$d^3$	64
$\{2, 5\}$	.04	$d^4$	512
$\{4, 5\}$	.03	$d^3$	64

se consigue así que cualquier coloración que trate de minimizar el grado de rigidez evite prioritariamente asignar el mismo color a los vértices 1 y 4, cuya penalización supera cualquier suma del resto de penalizaciones.

Si en el ejemplo 2.3, se considera que las penalizaciones  $p_{ij}$  son iguales a 1 para toda  $\{i, j\} \in \overline{E}$ , el grado de rigidez de una  $c$ -coloración se aumentará con el número de regiones no adyacentes igualmente coloreadas.

Como se ha visto en los tres ejemplos anteriores, con las penalizaciones de  $\overline{E}$  se pueden considerar coloraciones que satisfagan distintos criterios, ampliando así el campo de

problemas de coloración al no restringir el criterio de minimizar el número de colores utilizados en la coloración.

El grado de rigidez de una coloración mide la *robustez* de la coloración respecto de la adición de nuevas aristas al grafo; es decir, la validez de la coloración al no estar coloreados con el mismo color los extremos de las nuevas aristas. Para un  $c$  fijo, cuanto menor sea el grado de rigidez, mayor será la robustez de la coloración.

En el caso límite, cuando  $c = n$ , la coloración que asigna un color diferente a cada vértice, es decir,

$$C^n(i) = i \quad \forall i \in V$$

tendrá un grado de rigidez igual a 0. Esto se justifica al permanecer válida la coloración al añadirse cualquier arista al grafo.

De esta forma, entre las diversas  $c$ -coloraciones que se pueden definir en un grafo, tiene sentido determinar cuál es la de menor grado de rigidez. Este es el Problema de Coloración Robusta, de ahora en adelante *PCR*, y que se formaliza a continuación.

Dados un grafo  $G = (V, E)$ , un número de colores válido  $c > 0$  y una familia de penalizaciones de las aristas complementarias  $\{p_{ij}, \{i, j\} \in \overline{E}\}$ , El *Problema de Coloración Robusta* consiste en determinar aquella  $c$ -coloración  $C_R^c$  con menor grado de rigidez:

$$R(C_R^c) = \min_{C^c} R(C^c)$$

Considerando que la matriz de adyacencia del grafo  $G$  es simétrica y que se conocen las penalizaciones  $p_{ij}$  de las aristas complementarias  $\{i, j\} \in \overline{E}$ , cada ejemplo del *PCR* queda caracterizado por los parámetros  $n$ , número de vértices del grafo;  $c$ , el número de colores permitidos; y la matriz cuadrada  $H$  de dimensión  $n \times n$ , que almacena en la matriz triangular inferior la matriz de adyacencia de  $G$  y en la matriz triangular superior las penalizaciones:

$$h_{ij} = \begin{cases} p_{ij} & \text{si } i < j \\ b_{ji} & \text{si } i > j \end{cases}$$

de tal forma que la matriz  $H$  es:

$$H = \begin{pmatrix} 0 & p_{12} & p_{13} & \cdots & p_{1j} & \cdots & p_{1n} \\ b_{12} & 0 & p_{23} & \cdots & p_{2j} & \cdots & p_{2n} \\ b_{13} & b_{23} & 0 & \cdots & p_{3j} & \cdots & p_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \cdots & \vdots \\ b_{1j} & b_{2j} & b_{3j} & \cdots & 0 & \cdots & p_{jn} \\ \vdots & \vdots & \vdots & \cdots & \vdots & \ddots & \vdots \\ b_{1n} & b_{2n} & b_{3n} & \cdots & b_{jn} & \cdots & 0 \end{pmatrix}$$

$B$  es la matriz de adyacencia:

$$b_{ij} = \begin{cases} 1 & \text{si } \{i, j\} \in E \\ 0 & \text{si } \{i, j\} \notin E \end{cases} \quad i < j$$

y  $p_{ij}$  es la penalización del par  $\{i, j\} \notin E$  que vale 0 si  $\{i, j\} \in E$ .

Los parámetros  $(n, c, H)$  que caracterizan el ejemplo 2.1, son:

$$n = 6 \quad c = 4 \quad H = \begin{pmatrix} 0 & 0 & 0 & 0 & .0519 & .0128 \\ 1 & 0 & 0 & .6083 & .3621 & 0 \\ 1 & 1 & 0 & .1648 & 0 & .0256 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

donde las penalizaciones  $h_{ij}$  se han obtenido de la tabla

$\{i, j\} \in \bar{E}$	$pr_{ij}$	$p_{ij}$
$\{1, 5\}$	0.0506	0.0519
$\{1, 6\}$	0.0127	0.0128
$\{2, 4\}$	0.4557	0.6083
$\{2, 5\}$	0.3038	0.3621
$\{3, 4\}$	0.1519	0.1648
$\{3, 6\}$	0.0253	0.0256

La 4-coloración más robusta es

$$C_R^4(1) = 2, C_R^4(2) = 4, C_R^4(3) = 1, C_R^4(4) = 3, C_R^4(5) = 2, C_R^4(6) = 1$$

Su grado de rigidez es

$$R(C_R^4) = 0.0519$$

de forma que la probabilidad de que se mantenga válida la coloración  $C_R^4$  es

$$\frac{1}{\exp(0.0519)} = 0.9494$$

Por otra parte, los parámetros  $(n, c, H)$  que caracterizan el ejemplo 2.2, son:

$$n = 5 \quad c = 3 \quad H = \begin{pmatrix} 0 & 1 & 8 & 4096 & 512 \\ 0 & 0 & 512 & 64 & 512 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 64 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Hay que notar que el grafo sólo tiene las aristas  $\{3, 4\}$  y  $\{3, 5\}$  lo que indica que se considera un  $\alpha = 0.05$ . La 3-coloración más robusta es



$$C_R^3(1) = 1 \quad C_R^3(2) = 1 \quad C_R^3(3) = 2 \quad C_R^3(4) = 3 \quad C_R^3(5) = 3$$

Su grado de rigidez es  $R(C_R^3) = p_{12} + p_{45} = 65$ . La interpretación es la siguiente: al ser

$$64 = 8^2 \leq 65 < 512 = 8^3$$

la coloración  $C_R^3$  sigue siendo válida para umbrales superiores a  $d^3 = 0.03$  y, por consiguiente, el umbral crítico para 3 colores es:

$$\alpha^3 = \alpha(C_R^3) = 0.03$$

Los parámetros del ejemplo 2.3 son:

$$n = 8 \quad c = 4 \quad H = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

la rigidez de cada coloración es

$$R(C_1^4) = p_{BC} + p_{BF} + p_{BG} + p_{CF} + p_{CG} + p_{FG} + p_{AH} = 7$$

y

$$R(C_2^4) = p_{DG} + p_{BF} + p_{AE} + p_{CH} = 4$$

con lo que la coloración más robusta es  $C_2^4$  que, como ya se dijo, es preferible a  $C_1^4$ .

Si las penalizaciones son constantes, es decir, si  $p_{ij} = p \quad \forall \{i, j\} \in \overline{E}$ , entonces la coloración más robusta de un grafo tiende a homogeneizar el tamaño de las clases de

color, esta situación se puede apreciar en las clases de color  $C$  y  $C'$  del ejemplo 2.2 y en las coloraciones  $C_1^4$  y  $C_2^4$  del ejemplo 2.3. Esta propiedad se formaliza a continuación.

**Proposición 2.1** *Dada una  $c$ -coloración  $C_1^c$  de un grafo  $G = (V, E)$ , con  $c > \chi(G)$  y las penalizaciones de las aristas complementarias todas con el mismo valor positivo  $p_{ij} = p$ , con  $p > 0 \quad \forall \{i, j\} \in \overline{E}$ . Si dos de sus clases de color tienen cardinales que difieren en al menos dos, i.e.,*

$$n_1 = |V_{C_1}(k)| < |V_{C_1}(l)| = n_2$$

*con  $n_2 > n_1 + 1$ , entonces la  $c$ -coloración  $C_2^c$  que resulta de cambiar, si es posible, un vértice de la clase de color 1 a la clase de color  $k$  es más robusta que  $C_1^c$ .*

**Demostración:**

Las clases de color  $k$  y  $l$  de  $C_2^c$  tendrán  $n_1 + 1$  y  $n_2 - 1$  elementos respectivamente. Para demostrar la proposición basta con demostrar que la rigidez aportada por estas clases en la coloración  $C_1^c$  es mayor que la rigidez aportada en la coloración  $C_2^c$ , es decir

$$\binom{n_1}{2} + \binom{n_2}{2} > \binom{n_1 + 1}{2} + \binom{n_2 - 1}{2}$$

desarrollando los coeficientes binomiales, se tiene

$$\frac{1}{2(n_1^2 - n_1 + n_2^2 - n_2)} > \frac{1}{2(n_1^2 + n_1 + n_2^2 - 3n_2 + 2)}$$

que se reduce a

$$-n_1 - n_2 > n_1 - 3n_2 + 2$$

y por tanto es equivalente a

$$n_2 > n_1 + 1$$

que, al ser la hipótesis de partida, es cierta la desigualdad que se pretendía demostrar.

## 2.6 Complejidad algorítmica del Problema

Cuando se plantea un nuevo Problema de Optimización, la primera pregunta que surge es: ¿Puede resolverse por un algoritmo en tiempo polinomial?. Si la respuesta es afirmativa, el algoritmo construido permitirá resolver problemas de gran tamaño y se puede concluir que el Problema está resuelto. En caso contrario, sin embargo, cuando el algoritmo no es polinomial o no se es capaz de construirlo, hay que analizar la Complejidad Algorítmica del Problema de Optimización.

El análisis de la complejidad algorítmica requiere identificar el Problema de Decisión asociado para compararlo con otros problemas de Decisión analizados en la literatura, ver Garey y Johnson [26].

El Problema de Decisión asociado al  $PCR$ , denotado por  $PDCR$ , es el siguiente:

**PDCR:** Dados los parámetros que caracterizan al  $PCR$ ,  $(n, c, H)$ , y una cota superior del grado de rigidez,  $\bar{r}$ ,

**Cuestión:** ¿Existe una  $c$ -coloración  $C^c$  tal que

$$R(C^c) = \sum_{i < j, C^c(i)=C^c(j)} h_{ij} \leq \bar{r} \quad ?$$

A continuación se demuestra que el  $PDCR$  es NP-Completo.

**Proposición 2.2** *El Problema de Decisión  $PDCR$  pertenece a la clase NP.*

**Demostración:**

En efecto, dada una  $c$ -coloración  $C^c$ , en tiempo polinomial se puede comprobar si verifica o no las condiciones exigidas al problema de decisión. •

Para demostrar que el  $PDCR$  es NP-Completo se compara con el problema de decisión asociado al problema de coloración mínima (PDCM).

**PDCM:** Dados un grafo  $G = (V, E)$  y la constante  $\chi \leq |V|$ . ¿Existe una  $c$ -coloración  $C^c$  tal que  $c \leq \chi$ ?

**Proposición 2.3** *El Problema de Decisión PDCM es un caso particular del PDCR.*

**Demostración:**

Basta considerar que  $c = \chi$ , que  $p_{ij} = 0, \forall \{i, j\} \in \overline{E}$ , es decir, que si  $i < j$  entonces  $h_{ij} = 0$  y que  $\bar{r} = 0$ . •

Al ser el Problema de Decisión NP-Completo, se deducen los siguientes corolarios

**Corolario 2.1** *El Problema de decisión PDCR es NP-Completo.*

**Corolario 2.2** *El Problema de optimización PCR es NP-duro.*

Como para todos los problemas de este tipo, la forma de encontrar buenas soluciones para ejemplos de tamaño medio y grande del problema de coloración robusta es utilizando heurísticas.



# Capítulo 3

## Modelos matemáticos del PCR

El PCR admite distintas modelizaciones que se ilustrarán en este capítulo: como un problema de programación binaria, como un modelo de partición, como un modelo de asignación cuadrática o como una red generalizada.

En cualquier caso, todas estas modelizaciones ilustran la dificultad computacional de resolver el *PCR*, que, según se ha visto en el capítulo anterior, es un problema NP-duro. Por otra parte, los algoritmos exactos basados en estas modelizaciones permitirán identificar familias de heurísticas que resolverán eficientemente problemas de tamaño medio y grande.

Se supone que el *PCR* está planteado en un grafo  $G = (V, E)$  con  $n$  vértices y se han fijado el número de colores válido  $c > 0$  y las penalizaciones de las aristas complementarias  $\{p_{ij} / \{i, j\} \in \overline{E}\}$ .

### 3.1 Modelo de programación binaria

Sean las variables de decisión

$$x_{ik} = \begin{cases} 1 & \text{si } C(i) = k \\ 0 & \text{si } C(i) \neq k \end{cases} \quad \forall i \in \{1, \dots, n\} \quad \forall k \in \{1, \dots, c\}$$

Se introducen las siguientes variables auxiliares

$$y_{ij} = \begin{cases} 1 & \text{si existe } k \in \{1, \dots, c\} \text{ tal que } x_{ik} = x_{jk} \\ 0 & \text{en caso contrario} \end{cases} \quad \forall \{i, j\} \in \overline{E}$$

con el fin de plantear el PCR como el problema de programación binaria:

$$\begin{aligned} & \text{Min } \sum_{\{i,j\} \in \overline{E}} p_{ij} y_{ij} \\ \text{s.a.} \quad & \sum_{k=1}^c x_{ik} = 1 \quad \forall i \in \{1, \dots, n\} \\ & x_{ik} + x_{jk} \leq 1 \quad \forall \{i, j\} \in E \text{ y } \forall k \in \{1, \dots, c\} \\ & x_{ik} + x_{jk} - 1 \leq y_{ij} \quad \forall \{i, j\} \in \overline{E} \text{ y } \forall k \in \{1, \dots, c\} \end{aligned}$$

La primera familia de restricciones, al igual que en la formulación para el problema de coloración mínima, asegura que a cada vértice se le asigna un solo color. La segunda familia garantiza que la coloración sea válida y la tercera garantiza que si dos vértices  $i, j \in V$  no unidos por una arista tienen el mismo color, entonces la variable auxiliar  $y_{ij}$  vale 1.

Si  $\overline{m} = \frac{n(n-1)}{2} - m$  es el cardinal de  $\overline{E}$ , el número de variables binarias del modelo

es  $n \cdot c + \overline{m}$  y el número de restricciones es  $n + c \cdot m + c \cdot \overline{m} = n + c \cdot \frac{n(n-1)}{2}$ .

Las dos últimas familias de restricciones se pueden reemplazar por la familia de restricciones disyuntivas siguiente, controlada por las variables binarias  $z_{ij}$ , que son los elementos de la matriz de adyacencia del grafo  $G$ :

$$\begin{aligned} x_{ik} + x_{jk} &\leq 1 + (1 - z_{ij}) \quad \forall i, j, k \\ x_{ik} + x_{jk} - 1 &\leq y_{ij} + z_{ij} \quad \forall i, j, k \end{aligned}$$

Hay que observar que fijados  $i$  y  $j$ , si  $z_{ij} = 1$ , entonces  $\{i, j\} \in E$  y se relaja la segunda familia de restricciones; mientras que si  $z_{ij} = 0$ , entonces  $\{i, j\} \in \overline{E}$  y se relaja la primera familia de restricciones.

### 3.2 Cliques del grafo complementario. Modelo de partición

Un grafo  $G$  es *completo* si hay una arista entre cualquier par de vértices  $i, j$ , para todo  $i \neq j$ . Un grafo completo con  $n$  vértices se denota por  $K_n$  y se llama *n-clique*.

Obviamente, en el PCR se supone que  $c \leq n$ . El caso  $c = n$  es trivial, puesto que, en este caso, la coloración más robusta sería:

$$C_R^n(i) = i \quad \forall i = 1, \dots, n$$

al tener grado de rigidez  $R(C_R^n) = 0$ .

Para el caso  $c = n - 1$ , la coloración más robusta se deduce de la coloración  $C_R^n$  sin más que asignar el mismo color a los extremos de la arista complementaria  $\{i, j\} \in \overline{E}$  con menor penalización. Suponiendo una reordenación de vértices de forma que

$$p_{in} \equiv \min_{\{i,j\} \in \overline{E}} p_{ij}$$

la coloración más robusta con  $c = n - 1$  colores sería entonces:

$$C_R^{n-1}(i) = \begin{cases} i & \text{si } i \neq n \\ i_n & \text{si } i = n \end{cases}$$

En el caso general, cuando  $c < n - 1$ , no es cierto que la  $c$ -coloración más robusta sea la que repite color en las dos aristas complementarias de menor penalización. El resultado,



que se recoge en la siguiente proposición, es más general y hace referencia no sólo a las aristas de  $\overline{G}$  —cliques de orden 2— sino a los cliques de orden genérico  $K_r$  de  $\overline{G}$ .

Se entiende que un clique de orden 1 es un clique trivial formado por un solo vértice.

Cualquier  $k$ -clique de  $\overline{G}$  es un conjunto independiente de  $G$ , por lo que permite identificar las clases de color de una coloración válida. Esta idea se formaliza a continuación.

**Proposición 3.1** *Dado un grafo  $G = (V, E)$  con  $n$  vértices, fijado  $c \leq n$ , cualquier  $c$ -coloración de  $G$  tiene asociada la siguiente propiedad del grafo complementario  $\overline{G} = (V, \overline{E})$ :*

*En  $\overline{G}$  existen  $n_1, n_2, \dots, n_s$  cliques de orden  $1, 2, \dots, s$  respectivamente:*

$$\{K_k^t \subset V / t \in \{1, \dots, s\}, k \in \{1, \dots, n_t\}\}$$

*siendo  $1 \leq s \leq n - c + 1$  tal que*

$$n = n_1 + 2 \cdot n_2 + \dots + s \cdot n_s$$

*y estando los vértices de cada clique no trivial igualmente coloreados:*

$$C^c(i) = C^c(i') \quad \forall i, i' \in K_k^t \subset \overline{G}, \forall t \in \{2, \dots, s\} \forall k \in \{1, \dots, n_t\}$$

### **Demostración**

Al ser  $C^c$  una  $c$ -coloración, se verifica que las clases de color forman una partición del conjunto de vértices

$$V = \bigcup_{k=1}^c V_{C^c}(k)$$

Por ser  $C^c$  una coloración estricta, ver observación 2.1, cada una de estas clases es no vacía:

$$V_{C^c}(k) \neq \emptyset \quad \forall k \in \{1, \dots, c\}$$

y, por consiguiente,

$$1 \leq |V_{C^c}(k)| \leq n - c + 1 \quad \forall k \in \{1, \dots, c\}$$

Además,  $V_{C^c}(k)$  es un clique de  $\overline{G}$  para todo  $k \in \{1, \dots, c\}$ . En efecto, si  $|V_{C^c}(k)| = 1$  es un clique trivial, en caso contrario, es decir, si  $|V_{C^c}(k)| > 1$ , dados dos vértices  $i, i' \in V_{C^c}(k)$ , se verifica que  $\{i, i'\} \in \overline{E}$ , puesto que en caso contrario, la coloración  $C^c$  no sería válida.

Sean  $n_1, n_2, \dots, n_{n-c+1}$  el número de clases  $V_{C^c}(k)$  con cardinal  $k \in \{1, 2, \dots, n-c+1\}$  respectivamente. Entonces se verifica que

$$n = n_1 + 2 \cdot n_2 + \dots + s \cdot n_s$$

siendo

$$s \equiv \text{Max} \{t \in \{1, \dots, n-c+1\} / n_t > 0\}$$

•

La demostración de la proposición 3.1 asegura que dos cliques  $K_k^t$  no pueden tener ningún vértice en común. Tiene sentido, pues, introducir la siguiente definición.

**Definición 3.1** *Dado un grafo  $G = (V, E)$ , dos cliques  $K_1, K_2 \subset V$  son independientes si verifican*

$$K_1 \cap K_2 = \emptyset$$

Como corolario de la proposición 3.1 se deduce que la  $c$ -coloración más robusta será aquella que determine los valores óptimos para  $n_1^*, n_2^*, \dots, n_s^*$  y los cliques independientes asociados.

**Corolario 3.1** La  $c$ -coloración más robusta  $C_R^c$  está asociada a la determinación de los  $n_1^*, n_2^*, \dots, n_s^*$  cliques independientes de  $\overline{G}$ :

$$K_{*1}^1, \dots, K_{*n_1^*}^1; K_{*1}^2, \dots, K_{*n_2^*}^2; \dots; K_{*1}^s, \dots, K_{*n_s^*}^s$$

siendo

$$1 \leq s \leq n - c + 1$$

estando igualmente coloreados los vértices de los cliques no triviales

$$C_R^c(i) = C_R^c(j) \quad \forall i, j \in K_{*k}^t \quad \forall t \in \{2, \dots, s\} \quad \forall k \in \{1, \dots, n_t^*\}$$

y minimizando el grado de rigidez:

$$\sum_{t=2}^{s^*} \sum_{k=1}^{n_t^*} \sum_{i,j \in K_{*k}^t} h_{ij} \leq \sum_{t=2}^s \sum_{k=1}^{n_t} \sum_{i,j \in K_k^t} h_{ij}$$

entre todas las  $c$ -coloraciones verificando las condiciones de la proposición 3.1.

A partir de estas propiedades y con el fin de resolver el PCR a partir de los cliques del grafo complementario  $\overline{G}$ , se definirá una matriz basada en la matriz de cliques, ver Nemhauser y Wolsey [49], con  $n+1$  filas y tantas columnas como cliques tenga  $\overline{G}$ , incluidos los cliques de orden 1, que son los propios vértices del citado grafo.

**Definición 3.2** Dado el grafo complementario  $\overline{G}$  y los conjuntos  $m_1, m_2, \dots, m_s$  de cliques de orden  $1, 2, \dots, s$  respectivamente

$$K_1^1, \dots, K_{m_1}^1; K_1^2, \dots, K_{m_2}^2; \dots; K_1^s, \dots, K_{m_s}^s$$

donde  $1 \leq s \leq n - c + 1$ , se define la matriz de incidencia de cliques como

$$B^K = \begin{pmatrix} b_{11}^1 & \cdots & b_{1m_1}^1 & b_{11}^2 & \cdots & b_{1m_2}^2 & \cdots & b_{1j}^t & \cdots & b_{11}^s & \cdots & b_{1m_s}^s \\ \vdots & & \vdots & & & \vdots & & \vdots & & \vdots & & \vdots \\ b_{i1}^1 & \cdots & b_{im_1}^1 & b_{i1}^2 & \cdots & b_{im_2}^2 & \cdots & b_{ij}^t & \cdots & b_{i1}^s & \cdots & b_{im_s}^s \\ \vdots & & \vdots & & & \vdots & & \vdots & & \vdots & & \vdots \\ b_{n1}^1 & \cdots & b_{nm_1}^1 & b_{n1}^2 & \cdots & b_{nm_2}^2 & \cdots & b_{nj}^t & \cdots & b_{n1}^s & \cdots & b_{nm_s}^s \\ b_{n+1,1}^1 & \cdots & b_{n+1,m_1}^1 & b_{n+1,1}^2 & \cdots & b_{n+1,m_2}^2 & \cdots & b_{n+1,j}^t & \cdots & b_{n+1,1}^s & \cdots & b_{n+1,m_s}^s \end{pmatrix}$$

donde el clique  $K_k^t$  de orden  $t$  se caracteriza por el vector columna  $\mathbf{b}_k^t$  definido por:

$$b_{ik}^t = \begin{cases} 1 & \text{si } i \in K_k^t \\ 0 & \text{si } i \notin K_k^t \end{cases} \quad i \in \{1, \dots, n\}$$

y

$$b_{n+1,k}^t = \sum_{i=1}^n b_{ik}^t = t$$

Obviamente, hay tantos cliques de orden 1 como vértices, de forma que  $m_1 = n$ ,

$$b_{ik}^1 = \begin{cases} 1 & \text{si } i = k \\ 0 & \text{si } i \neq k \end{cases} \quad \forall i, k \in \{1, \dots, n\}$$

y

$$b_{n+1,k}^1 = 1 \quad \forall k \in \{1, \dots, m_1\}$$

**Observación 3.1** No confundir la matriz  $B^K$ , que se refiere al grafo complementario  $\overline{G}$ , con la matriz de adyacencia del grafo  $G$ , que es  $B$ .

El PCR se puede plantear como un Problema de Partición. Con este fin se introduce la siguiente notación:

Sea

$$m^K \equiv m_1 + m_2 + \cdots + m_s$$

el número de cliques de  $\overline{G}$ . Introduciendo el índice  $j$  para identificar cada uno de los cliques  $K_k^t$  de orden  $t$  según la relación:

$$j = \sum_{l=1}^{t-1} m_l + k$$

e, inversamente, identificando el clique  $K_k^t$  a partir del índice  $j$  según:

$$t = \text{Min}\{r / \sum_{l=1}^r m_l \geq j\}$$

y

$$k = j - \sum_{l=1}^{t-1} m_l$$

La matriz de incidencia de cliques de  $\overline{G}$  se especifica por

$$B^K = (b_{ij}^K) \quad i \in \{1, \dots, n+1\} \quad j \in \{1, \dots, m^K\}$$

La variable de decisión será el vector de dimensión  $m$  definido por

$$x_j = \begin{cases} 1 & \text{si } K_j \text{ es elegido} \\ 0 & \text{en caso contrario} \end{cases} \quad j \in \{1, \dots, m^K\}$$

que indica si el clique  $j$ -ésimo es elegido para colorear todos sus vértices con el mismo color

$$C^c(i) = C^c(i') \quad \forall i, i' \in K_j$$

Hay que exigir que sean  $c$  los cliques seleccionados, por lo que

$$\sum_{j=1}^{m^K} x_j = c$$

Cada clique  $K_j$  está caracterizado por la penalización  $p_j$  asociada a las aristas complementarias que están incluidas en el clique

$$p_j = \sum_{i, i' \in K_j} p_{ii'}$$

Al exigir que cada uno de los cliques sea independiente, se asegura que cada vértice será coloreado con un único color.

Con todo lo expuesto anteriormente, el PCR se puede plantear como un Problema de Partición al minimizar

$$\sum_{j=1}^{m^K} p_j x_j \quad (3.1)$$

verificando:

$$\sum_{j=1}^{m^K} x_j = c \quad (3.2)$$

$$\sum_{j=1}^{m^K} b_{ij}^K x_j = 1 \quad \forall i \in \{1, \dots, n\} \quad \forall k \in \{1, \dots, s\} \quad (3.3)$$

$$\sum_{j=1}^{m^K} b_{n+1,j}^K x_j = n \quad (3.4)$$

La primera restricción garantiza que se utilicen sólo  $c$  colores. El segundo grupo de restricciones impone que los cliques utilizados sean independientes. La tercera restricción asegura que todos los vértices estén coloreados.

El número de variables es  $m^K$ , un parámetro que crece exponencialmente con  $n$  y  $m$  al especificar el número de cliques del grafo complementario. El número de restricciones es  $n \cdot m^K + 2$ .

*En el ejemplo 2.1, hay un total de 12 cliques, incluidos los 6 cliques triviales formados por los vértices. La matriz de incidencia de cliques es la siguiente:*

$$B^K = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 2 \end{pmatrix}$$

cuyos pesos son, respectivamente:

$$p(j) = (0, 0, 0, 0, 0, 0, 0.0408, 0.0100, 0.4463, 0.2744, 0.1278, 0.0202)$$

Si se considera  $c = 4$ , el óptimo del problema de partición se obtiene con

$$x_2 = 1 \quad x_4 = 1 \quad x_7 = 1 \quad x_{12} = 1$$

y anulando el resto de variables binarias. El peso obtenido es, por consiguiente

$$p_2 + p_4 + p_7 + p_{12} = 0 + 0 + 0.0408 + 0.0202 = 0.0610$$

La coloración asociada se construye asignando el mismo color a cada uno de los vértices incluidos en los 4 cliques elegidos:

- clique 2.  $K_2 = \{2\}$ .  $C_R^4(2) = 4$ .
- clique 4.  $K_4 = \{4\}$ .  $C_R^4(4) = 3$ .
- clique 7.  $K_7 = \{1, 5\}$ .  $C_R^4(1) = C_R^4(5) = 2$ .
- clique 12.  $K_{12} = \{3, 6\}$ .  $C_R^4(3) = C_R^4(6) = 1$ .

En el ejemplo 2.2, hay 19 cliques, de forma que la matriz de incidencia de cliques es:

$$B^K = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 3 & 4 \end{pmatrix}$$

con pesos

$$p(j) = (0, 0, 0, 0, 0, 1, 8, 4096, 512, 512, 64, 512, 64, 521, 4161, 1025, 4672, 640, 5249)$$

Para  $c = 3$ , la solución del problema de partición es el vector de variables binarias:

$$(0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0)$$

que está asociado a la elección de los cliques 3, 6 y 13. El peso de todos estos cliques es

$$p_3 + p_6 + p_{13} = 0 + 1 + 64 = 65$$

La coloración más robusta asociada es :

- clique 3.  $K_3 = \{3\}$ .  $C_R^3(3) = 2$ .
- clique 6.  $K_6 = \{1, 2\}$ .  $C_R^3(1) = C_R^3(2) = 1$ .
- clique 13.  $K_{13} = \{4, 5\}$ .  $C_R^3(4) = C_R^3(5) = 3$ .

### 3.3 Modelo de Asignación Cuadrática

Considerando que el PCR se puede formular como un problema de programación binaria, es natural plantearse si se puede adaptar a un modelo de asignación cuadrática. La respuesta es afirmativa, como recoge la siguiente proposición.



**Proposición 3.2** *El PCR se puede plantear como un problema de asignación cuadrática.*

**Demostración:**

Sea el PCR con los parámetros  $(n, c, H)$ , siendo  $G = (V, E)$  el grafo a colorear,  $n = |V|$  y  $m = |E|$ ;  $c$  es el número de colores válido y la matriz  $H$  la matriz que especifica en la matriz triangular inferior la matriz de adyacencia de  $G$  y en la triangular superior las penalizaciones de las aristas complementarias  $\overline{E}$ .

La variable de decisión se construye a partir del conjunto de variables binarias:

$$((x_{11}, x_{21}, \dots, x_{n1}), (x_{12}, x_{22}, \dots, x_{n2}), \dots, (x_{1c}, x_{2c}, \dots, x_{nc}))$$

que identifican la  $c$ -coloración  $C^c$ :

$$x_{ik} = \begin{cases} 1 & C^c(i) = k \\ 0 & C^c(i) \neq k \end{cases} \quad \forall i \in \{1, \dots, n\} \quad \forall k \in \{1, \dots, c\}$$

La matriz  $A$  de restricciones del PAC debe caracterizar las soluciones factibles del problema de coloración:

1. Cada vértice se colorea de forma única.

$$\sum_{k=1}^c x_{ik} = 1 \quad \forall i \in \{1, \dots, n\} \quad (3.5)$$

2. Los extremos de cualquier arista no pueden ser igualmente coloreados:

$$x_{ik} + x_{jk} \leq 1 \quad \forall \{i, j\} \in E \quad \forall k \in \{1, \dots, c\} \quad (3.6)$$

La matriz  $A$  y el vector  $\mathbf{b}$  tienen  $n \cdot c$  filas:



Si en el ejemplo 2.2 se hace  $c = 3$  y sabiendo que las penalizaciones, *i.e.*, las cajas de  $Q$  son:

$$\begin{pmatrix} 0 & 1 & 8 & 4096 & 512 \\ 1 & 0 & 512 & 64 & 512 \\ 8 & 512 & 0 & 0 & 0 \\ 4096 & 64 & 0 & 0 & 64 \\ 512 & 512 & 0 & 64 & 0 \end{pmatrix}$$

la función objetivo es

$$\frac{1}{2}(x_{11}x_{21} + 8x_{11}x_{31} + 4096x_{11}x_{41} + 512x_{11}x_{51} + \dots + 512x_{13}x_{53} + 512x_{23}x_{53} + 64x_{43}x_{53})$$

Este planteamiento del PCR contiene muchas variables, aún en problemas pequeños, como para ser manejable. Los algoritmos que resuelven este tipo de problemas son métodos de enumeración, que sólo producen una solución exacta en un tiempo razonable en problemas con  $n \leq 15$ , según reporta Santiago Cano [10].

### 3.4 Modelo de red generalizada

A partir del análisis de los cliques del grafo complementario, el *PCR* se puede plantear como un problema de coste óptimo en una red generalizada  $G^K = (V^K, U^K)$ .

Los vértices de la red son de tres tipos  $V^K = V_1^K \cup V_2^K \cup \{F\} \cup \{S\}$ , donde:

1.  $V_1^K$  es el conjunto de vértices del grafo  $G$  del *PCR*.

$$V_1^K = \{1, 2, \dots, n\}$$

2.  $V_2^K$  es el conjunto de cliques del grafo complementario  $\overline{G}$ :

$$V_2^K = \{K_1^1, \dots, K_{m_1}^1; K_1^2, \dots, K_{m_2}^2; \dots; K_1^s, \dots, K_{m_s}^s\}$$

con  $1 \leq s \leq n - c + 1$ .

3. Dos vértices ficticios  $F$  y  $S$  que generan y recogen respectivamente el flujo de la red generalizada.

Se tiene así un total de  $n + m + 2$  vértices.

El conjunto de arcos de la red generalizada incluye arcos de tres tipos  $U^K = U_F^K \cup U_0^K \cup U_S^K$ . Para cada uno de los arcos  $u \in U^K$  se incluye el límite inferior de flujo  $l_u$ , el límite superior  $c_u$ , el coste  $d_u$  y el multiplicador  $m_u$ :

1. Arcos de selección de cliques del grafo complementario. El flujo unitario por uno de los arcos implica la selección del clique y la coloración de todos sus vértices con el mismo color.

$$U_F^K = \{(F, K_k^t) / t \in \{1, \dots, s\}, k \in \{1, \dots, m_t\}\}$$

$$l_u = 0 \quad c_u = 1 \quad d_u = \sum_{i,j \in K_k^t} h_{ij} \quad m_u = |K_k^t| \quad \forall u = (F, K_k^t) \in U_F^K$$

2. Arcos de cubrimiento de vértices por los cliques. Cada clique elegido debe cubrir los vértices que están incluidos en dicho clique.

$$U_0^K = \{(K_k^t, j) / t \in \{1, \dots, s\}, k \in \{1, \dots, m_t\}, j \in K_k^t\}$$

$$l_u = 0 \quad c_u = 1 \quad d_u = 0 \quad m_u = 1 \quad \forall u = (K_k^t, j) \in U_0^K$$

El flujo por estos arcos es igual a 1 cuando el clique asociado al extremo inicial es elegido.

3. Arcos de incompatibilidad de vértices para evitar que un vértice del grafo original sea coloreado con más de un color.

$$U_S^K = \{(j, S) / j \in V_1^K\}$$

$$l_u = 0 \quad c_u = 1 \quad d_u = 0 \quad m_u = 1 \quad \forall u = (j, S) \in U_S^K$$

Se evita con la capacidad de estos arcos que los vértices sean cubiertos por más de un clique.

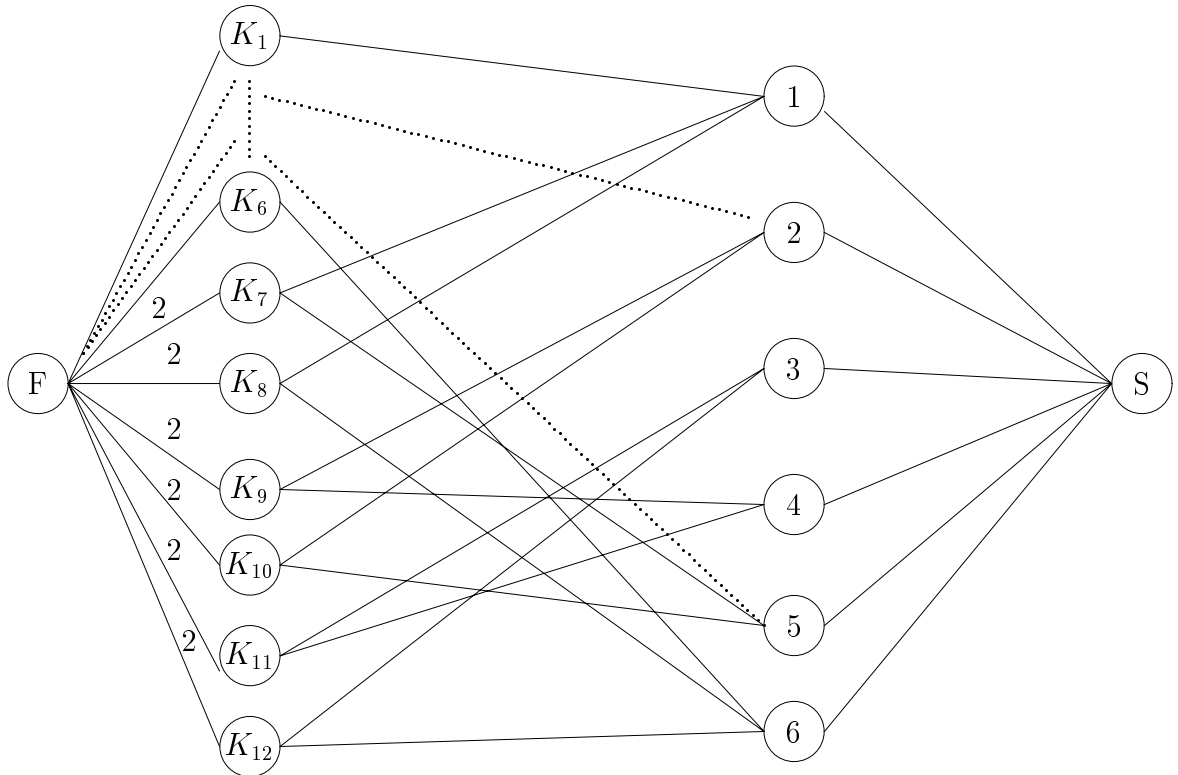


Figura 3.1: Red generalizada ejemplo 2.1

En la red generalizada  $G^K$  así definida, el problema de coloración robusto se plantea como un problema de flujo inicial fijo  $v_F = c$ , el número de colores válido, flujo final fijo  $v_S = n$ , para garantizar que todos los vértices han sido coloreados, y con coste mínimo, ya que el coste sólo depende del coste de los cliques seleccionados.

*En el ejemplo 2.1, con 12 cliques, la red generalizada sería la dibujada en la figura 3.1. Sobre los arcos se ha representado el multiplicador cuando no es igual a 1.*

*El flujo de coste mínimo en esta red generalizada proporciona la solución de PCR. Los arcos que se indican a continuación son los que tienen flujo unidad, el resto tiene flujo nulo:*

$$(F, K_2) \quad (F, K_4) \quad (F, K_7) \quad (F, K_{12})$$

$$(K_2, 2) \quad (K_4, 4) \quad (K_7, 1) \quad (K_7, 5) \quad (K_{12}, 3) \quad (K_{12}, 6)$$

$$(1, S) \quad (2, S) \quad (3, S) \quad (4, S) \quad (5, S) \quad (6, S)$$

Los problemas discretos tienen espacios de soluciones muy grandes en tamaño, y crecen explosivamente con el número de selecciones discretas a resolverse; por ejemplo, un problema que requiere 200 decisiones binarias independientes tiene  $2^{200}$  o cerca de  $10^{60}$  soluciones a considerar.

Como se ha visto, el modelo de programación más sencillo para representar al PCR tiene  $n \cdot c + \overline{m}$  variables donde  $n$  es el número de vértices,  $c$  el número de colores a usar y  $\overline{m}$  es el número de aristas complementarias. Que para un problema pequeño que se puede modelar con un grafo de 25 vértices y  $\overline{m} = 150$ , que representa una densidad de 50%, el modelo de programación binaria para una 8-coloración tiene 350 variables y 2425 restricciones.

Por otra parte, el análisis del politopo asociado al PCR es más complejo que el del Problema de Coloración Mínima, por lo que su análisis es complicado y no se esperan resultados espectaculares al buscar reforzamientos del recinto definido por las soluciones factibles.

### 3.5 Experiencias computacionales

Debido al crecimiento explosivo del número de variables y restricciones cuando aumenta el tamaño del grafo, el modelo de programación binaria sólo puede resolver problemas de tamaño pequeño o medio. Por ejemplo, en un grafo de cuatro vértices y tres aristas,

para hallar una 2-coloración, el modelo de programación binaria correspondiente tiene 11 variables y 20 restricciones; el modelo para hallar una 5-coloración en un grafo con 10 vértices y 23 aristas tiene 72 variables y 275 restricciones; el modelo para una 7-coloración de un grafo con 20 vértices y 95 aristas tiene 188 variables y 1470 restricciones. Usando el GAMS 2.25 que usa un procedimiento de ramificación y acotamiento para resolver este tipo de problemas; en instancias de grafos con 15 vértices y 5 colores, en algunos casos no se pudo encontrar una solución entera, pues en menos de 50 000 iteraciones se agotaron los nodos disponibles para la ramificación; en otros casos se halló una solución entera, pero no óptima, al agotar los nodos disponibles.

En el modelo de partición en cliques, además del número creciente de variables y restricciones cuando se incrementa el número de vértices, hay que añadir el problema de encontrar todos los cliques de diferentes tamaños. El modelo para hallar una 5-coloración en una instancia  $G_{10,0.5}$ , donde 10 es el número de vértices y 0.5 es la probabilidad de que haya una arista entre dos vértices cualesquiera, tiene 32 variables y 1600 restricciones.

Por lo tanto se ha utilizado el modelo de programación binaria como la alternativa más razonable, dentro de las limitaciones que tienen los modelos de programación entera, para realizar algunas experiencias computacionales con el Problema de Coloración Robusta.

Se generaron aleatoriamente grafos de 10 a 20 vértices con densidad igual a 0.5, es decir, la probabilidad de que exista la arista  $\{i, j\}$  para cualquier pareja  $i, j \in \{1, \dots, n\}$  es igual a 0.5.

La penalización de las aristas complementarias se ha generado aleatoriamente de forma uniforme en el intervalo  $[0, 1]$ .

Los valores del parámetro  $c$  se han fijado entre 4, 5 y 6.

Se han resuelto los correspondientes problemas de programación binaria con el GAMS 2.25. Las pruebas se realizaron en un ordenador con procesador Intel Celeron de 550 Mhz., y se programó en Fortran-90.

Estos resultados se presentan en la siguiente tabla. Cada problema ha sido identificado

por el fichero que almacena la información, el número de vértices y aristas del grafo asociado, las aristas complementarias, el número de variables y restricciones del modelo de programación.

GRAFO	n	m	$\bar{m}$	Var	Rest	c	GAMS
al10	10	24	21	59	190	4	3.8458
				69	275	5	2.1641
al11	11	31	29	70	231	4	3.9532(*)
				81	286	5	2.6373
al12	12	33	33	81	276	4	3.8274(*)
				93	342	5	4.634
al13	13	37	41	93	325	5	4.9117
al14	14	46	45	116	469	5	5.0689
al15	15	52	53	128	540	5	7.5935
al15b	15	57	48	123	645	6	—

Experiencias computacionales con el Modelo de Programación Binaria

La tabla muestra la rigidez obtenida en cada uno de los problemas. El número de nodos en el correspondiente método de ramificación y acotación está limitado por el programa, por lo que no termina siempre el procedimiento de enumeración implícita cuando el número de variables es suficientemente grande, por lo que el resultado presentado cuando alcanza el límite anterior es sólo una aproximación de la solución. Con un asterisco se ha marcado cuando se ha alcanzado el óptimo. Así se concluye que —salvo en los problemas identificados como *al114* y *al124*— el modelo de programación funciona realmente como una heurística aportando sólo una solución aproximada.

Si dentro de este límite de nodos que sondea no se ha encontrado ninguna solución factible, se especificará " — ". En esta situación se encuentra el problema identificado por *al15b*. Este hecho pone de manifiesto que el modelo de programación no se puede aplicar más que a problemas de tamaño muy reducido.





# Capítulo 4

## Heurísticas para el Problema de Coloración Robusta

Al ser el *PCR* un problema NP-duro, es necesario el uso de heurísticas para resolver ejemplos de tamaño medio o grande. Algoritmos heurísticos pueden ser los algoritmos voraces, algoritmos de búsqueda local, aproximaciones por técnicas de relajación, enumeración parcial, técnicas de partición y descomposición, etc.

Estas técnicas se pueden aplicar a los distintos modelos matemáticos del *PCR* descritos anteriormente. En este capítulo se describirán dos heurísticas: la primera de enumeración parcial, está basada en los cliques del grafo complementario, la segunda heurística se basa en la metaheurística de los algoritmos genéticos.

### 4.1 Heurística basada en $k$ -cliques

Dado el grafo  $G = (V, E)$ , el número de colores válido  $c$  y la familia de penalizaciones  $\{p_{ij} / \{i, j\} \in \overline{E}\}$ , se analizó en la sección 3.2 la relación entre el *PCR* y los cliques del grafo complementario  $\overline{G} = (V, \overline{E})$ .

Según aumenta el tamaño del grafo, el número de aristas complementarias  $\overline{m} = |\overline{E}|$

crece y, en consecuencia, el número de columnas de la matriz  $B^K$  crece exponencialmente, por lo que la simple identificación de esta matriz puede ser prohibitiva a partir de determinados tamaños.

Por otra parte, el corolario 3.1 precisa del cálculo de  $n_1^*, n_2^*, \dots, n_s^*$  cliques independientes de  $\overline{G}$ . Como se verá a continuación, el número de cliques debe seguir unas relaciones muy precisas, lo que dificultará aún más la solución exacta del problema.

En efecto, el número de  $k$ -cliques asociados a una  $c$ -coloración, dependerá del valor de  $c$ . Habrá  $c$ -coloraciones en las que baste hallar un número determinado de 2-cliques. En otras, sin embargo, conforme disminuye el valor de  $c$ , será necesario considerar cliques de orden superior, pues no es suficiente considerar sólo 2-cliques. Esta situación se ilustra con el siguiente ejemplo.

**Ejemplo 4.1** *Considérese el PCR cuyos parámetros son los siguientes*

$$n = 7, \quad c = 3, \quad H = \begin{pmatrix} 0 & 0 & .0194 & .0052 & .0544 & 0 & .0712 \\ 1 & 0 & 0 & .0028 & .0103 & 0 & 0 \\ 0 & 1 & 0 & .0009 & .0103 & .0025 & 0 \\ 0 & 0 & 0 & 0 & 0 & .0007 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

*Al tener el grafo 7 vértices, si se consideran sólo 2-cliques, la mínima  $c$ -coloración posible es cuando  $c$  vale al menos 4, por lo que una 3-coloración requeriría al menos un 3-clique. Concretamente, si se consideran tres 2-cliques, es decir, se pintan seis vértices con 3 colores y el vértice restante con el cuarto color.*

Es preciso, pues, estudiar las relaciones entre  $c$  y  $n$  a partir del análisis de los cliques realizado en la sección 3.2 y, concretamente, de las fórmulas:

$$n = n_1 + 2 \cdot n_2 + \cdots + s \cdot n_s$$

y

$$c = n_1 + n_2 + \cdots + n_s$$

asociadas a cualquier  $c$ -coloración que colorea igualmente  $(n_1, \dots, n_s)$  cliques de orden  $(1, \dots, s)$  respectivamente del grafo complementario  $\overline{G}$ :

- Si el valor de  $c$  es tal que sólo se necesitan encontrar 2-cliques, entonces el número de vértices se puede expresar como

$$n = n_1 + 2n_2 \tag{4.1}$$

y el de colores como

$$c = n_1 + n_2 \tag{4.2}$$

De las ecuaciones 4.1 y 4.2 se concluye que los valores de  $c$  están acotados:

$$\frac{n}{2} \leq c \leq n$$

no pudiendo considerar, por tanto, valores de  $c$  menores que  $\lfloor \frac{n}{2} \rfloor$ .

En algunos casos resulta más robusta una coloración que considera un 3-clique y un 1-clique en lugar de dos 2-cliques, esta situación se analiza en el algoritmo, también véase el ejemplo 4.2.

- Para valores de  $c$  menores que  $\lfloor \frac{n}{2} \rfloor$  es necesario considerar cliques de orden superior, puesto que los 1-cliques y los 2-cliques no permitirían resolver el problema.

Si se consideran  $k$ -cliques, con  $k \in \{1, 2, 3\}$ , se deben verificar las siguientes relaciones para  $n$  y  $c$ :

$$n = n_1 + 2n_2 + 3n_3 \quad (4.3)$$

y el número de colores válidos

$$c = n_1 + n_2 + n_3 \quad (4.4)$$

De ambas expresiones, se deduce

$$3c - n = 2n_1 + n_2 \quad (4.5)$$

de donde

$$3c \geq n \quad (4.6)$$

lo que determina el mínimo valor de  $c$ .

De la diferencia entre 4.3 y 4.4 se obtiene también la expresión

$$n - c = n_2 + 2n_3 \quad (4.7)$$

de donde

$$n_2 = n - c - 2n_3 \quad (4.8)$$

De la expresión 4.4 se obtiene

$$n_1 = c - n_2 - n_3 \geq 0 \quad (4.9)$$

que proporciona una cota superior para el valor de  $n_2$

$$n_2 \leq c - n_3 \quad (4.10)$$

con las restricciones 4.8 y 4.10, se define el siguiente recinto:

$$n_2 + 2n_3 = n - c \quad (4.11)$$

$$n_2 + n_3 \leq c \quad (4.12)$$

$$n_2, n_3 \in \{0, 1, 2, \dots\} \quad (4.13)$$

Este recinto, representado en la figura 4.1, permite determinar los valores factibles para los parámetros  $n_2$  y  $n_3$ .

Hay que observar que existirá solución factible siempre y cuando  $c \geq \frac{n - c}{2}$ , que es equivalente a exigir que  $c \geq \frac{n}{3}$ .

Teniendo en cuenta, además, que en el caso en que las penalizaciones sean todas iguales y positivas interesa utilizar los cliques de menor orden posible, se acota en 3 dicho orden. Si son necesarios cliques de orden 3, convendrá que sea el menor número posible, por lo que tiene sentido minimizar  $n_3$ .

*Al determinar este menor valor de  $n_3$  en el ejemplo 4.1, se plantea el problema de programación entera*

$$\begin{array}{ll} \text{Min} & n_3 \\ \text{s.a} & n_2 + 2n_3 = 4 \\ & n_2 + n_3 \leq 3 \\ & n_2, n_3 \in \{0, 1, 2, \dots\} \end{array}$$

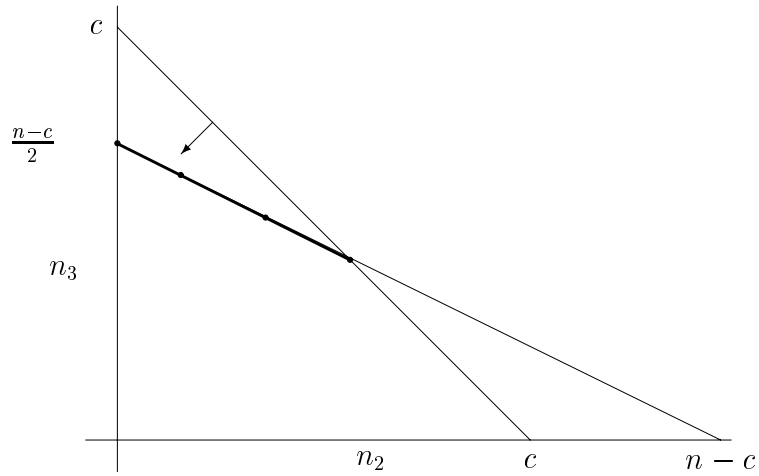


Figura 4.1: Recinto asociado a  $n_2$  y  $n_3$

cuya solución óptima es  $n_3 = 1$  y  $n_2 = 2$ .

No es necesario, sin embargo, resolver ningún problema de programación entera en la heurística que sólo utiliza cliques de orden 3 como máximo, puesto que los valores de  $n_1$ ,  $n_2$  y  $n_3$  deben ser enteros y se verifica que el menor valor posible de  $n_3$  es

$$n_3^0 = \lfloor n - 2 \cdot c \rfloor$$

y los valores de  $n_2$  y  $n_1$  se determinan a partir del anterior según

$$n_2 = n - c - 2 \cdot n_3$$

y

$$n_1 = n - 2 \cdot n_2 - 3 \cdot n_3$$

A continuación se detalla esta heurística que busca la  $c$ -coloración más robusta utilizando sólo cliques de orden 3 como máximo. Con este fin se determina primero la combinación apropiada del número de 2-cliques y 3-cliques del grafo complementario y se eligen de forma voraz a continuación dichos cliques.

Por lo señalado anteriormente, es fácil ver que:

- $c \geq n - \lceil \frac{n}{2} \rceil$ , entonces el mínimo valor de  $n_3$  es cero y  $n_2 = n - c$ .
- $c = n - \lceil \frac{n}{2} \rceil$  y  $n$  es impar, entonces el mínimo valor de  $n_3$  es 1 y  $n_2 = \lceil \frac{n}{2} \rceil - 2$ .
- $c < \lfloor \frac{n}{2} \rfloor$  y
  - $n$  es par, entonces el mínimo valor de la función objetivo es  $n_3 = 2(\lfloor \frac{n}{2} \rfloor - c)$  y  $n_2 = \lfloor \frac{n}{2} \rfloor - 3(\lfloor \frac{n}{2} \rfloor - c)$ .
  - $n$  es impar, entonces  $n_3 = 2(\lfloor \frac{n}{2} \rfloor - c) + 1$  y  $n_2 = \lfloor \frac{n}{2} \rfloor - 3(\lfloor \frac{n}{2} \rfloor - c) - 1$ .

Si no se encuentra ninguna  $c$ -coloración con los  $n_3$  3-cliques y con los  $n_2$  2-cliques, entonces se hace  $n_3 = n_3 + 1$  y  $n_2 = n_2 - 2$  las veces que sea necesario, siempre que  $n_2 \geq 0$ .

Para concretar el procedimiento anterior, considérese un grafo  $G$  con 10 vértices, 5 colores permitidos y 20 aristas; por consiguiente,  $n = 10$ ,  $c = 5$  y  $m = 20$ . El grafo complementario  $\bar{G}$  tiene entonces  $\bar{m} = 45 - 20 = 25$  aristas complementarias.

Según las fórmulas anteriores,  $n_3^0 = \lfloor n - 2 \cdot c \rfloor = 0$ . Los valores posibles de  $n_1, n_2$  y  $n_3$  se recogen en la siguiente tabla:

$n_3$	$n_2$	$n_1$	$n_1 + n_2 + n_3 (= c)$	$n_1 + 2n_2 + 3n_3 (= n)$
0	5	0	5	10
1	3	1	5	10
2	1	2	5	10

La determinación de cada una de los  $(n_3, n_2, n_1)$  cliques de orden  $(3, 2, 1)$  respectivamente se realiza a partir de las  $\bar{m}$  aristas complementarias lo que, en el peor de los casos, requiere identificar

$$\begin{pmatrix} \bar{m} \\ 3n_3 \end{pmatrix} \cdot \begin{pmatrix} \bar{m} - 3n_3 \\ n_2 \end{pmatrix}$$

combinaciones de aristas complementarias.



En el caso anterior de 10 vértices y 25 aristas complementarias se tendrían los siguientes valores:

$n_3$	$n_2$	$n_1$	Aristas complementarias
0	5	0	53130
1	3	1	38500
2	1	2	3364900

que implica, en definitiva, analizar un total de 3456530 combinaciones de aristas complementarias.

Las dificultades computacionales apuntadas justifican la construcción de un algoritmo de enumeración parcial que, en lugar de buscar la mejor combinación de  $(n_3, n_2, n_1)$  cliques, se quede con la primera encontrada después de ordenar convenientemente la búsqueda.

Con este propósito se ordenarán las aristas complementarias por la rigidez que aportan si sus extremos son coloreados igualmente, es decir, se ordenan en forma creciente por la penalización para que sean elegidas primero las de menor penalización.

El algoritmo, una vez fijados los  $(n_3, n_2, n_1)$  cliques, elige primero los  $n_3$  cliques de orden 3 independientes formados por las aristas de menor penalización, después selecciona los  $n_2$  2-cliques (aristas) de menor penalización que sean también independientes y completar con los 1-cliques, que no aportan ninguna penalización al ser vértices que no comparten el color con ningún otro vértice.

A continuación se presenta el esquema de la heurística basada en los cliques de orden 3 y menores basada en las ideas anteriormente expuestas.

```

inicio
  lectura(n,c,H)
  Ordenar las  $\bar{m}$  aristas complementarias según penalización
  if( $3c \geq n$ ) then
     $n_3 = \lfloor n - 2c \rfloor$ 
     $n_2 = \max\{n - c - 2n_3; 0\}$ 
     $n_1 = n - 2n_2 - 3n_3$ 
    do while( $n_1 \geq 0$  .AND. solución  $\neq 1$ )
      Determinar  $n_3$  cliques orden 3 independientes
      Determinar  $n_2$  cliques orden 2 independientes
      if (existen  $n_3$  y  $n_2$  cliques) then
        solución = 1
      else
         $n_3 = n_3 + 1$ 
         $n_2 = \max\{n - c - 2n_3; 0\}$ 
         $n_1 = n - 2n_2 - 3n_3$ 
      endif
    enddo
  else
    No existe coloración. Es preciso aumentar el orden de los cliques.
  endif
fin

```

Obviamente, al limitar el tamaño de los cliques y evitar la búsqueda exhaustiva entre éstos, se disminuye drásticamente el tiempo de computación pero, por otra parte, el espacio de las soluciones queda limitado impidiendo, a veces, la determinación de una  $c$ -coloración aún en el caso en que exista.

A continuación se presentan dos ejemplos que muestran características relevantes que

deben ser consideradas en el algoritmo.

**Ejemplo 4.2** *Encontrar las coloraciones más robustas con 6, 5, 4 y 3 colores respectivamente, en el grafo del ejemplo 4.1.*

*Las aristas complementarias ordenadas de menor a mayor según su peso son:*

$$\{4, 6\}\{3, 4\}\{3, 6\}\{2, 4\}\{1, 4\}\{2, 5\}\{3, 5\}\{1, 3\}\{1, 5\}\{1, 7\}$$

*si  $c$  es 6, entonces se encuentra un 2-clique de peso mínimo en  $G^c$ , que es  $\{4, 6\}$ , se pintan igual ambos vértices en  $G$  y los restantes con el color cuyo número corresponda con el del vértice.*

*Si el valor de  $c$  es 5, entonces hay que encontrar dos 2-cliques de peso mínimo y colorear los vértices de  $G$  como ya se explicó. Aunque a veces resulta más conveniente encontrar un  $K_3$  de peso mínimo en lugar de dos  $K_2$  de peso mínimo, como en este ejemplo.*

*En la siguiente tabla se presentan las coloraciones más robustas para los diferentes valores de  $c$ .*

$c$	$n_3$	$n_2$	$n_1$	cliques	$C_R^c$							$R(C_R^c)$
					1	2	3	4	5	6	7	
6	0	1	5	$\{4, 6\}$	2	3	4	1	5	1	6	.0007
5	1	0	4	$\{3, 4, 6\}$	2	3	1	1	4	1	5	.0041
4	1	1	2	$\{3, 4, 6\}$ $\{2, 5\}$	3	2	1	1	2	1	4	.0144
3	1	2	0	$\{3, 4, 6\}$ $\{2, 5\}$ $\{1, 7\}$	3	2	1	1	2	1	3	.0856

Está claro que si  $c$  es menor que  $\lfloor \frac{n}{2} \rfloor$  debe encontrarse al menos un 3-clique en  $\overline{G}$  para conseguir una coloración válida. En la tabla anterior se observa que la coloración más robusta con  $n - i$  colores se obtiene a partir de la coloración más robusta con  $n - i + 1$  colores; en general esto no sucede, como puede observarse en el siguiente ejemplo.

Coloreando de esta manera, se puede obtener una cota inferior para la robustez de una coloración con  $n - i$  colores, si en la ordenación, a partir del último clique encontrado en la coloración más robusta con  $n - i + 1$  colores, se encuentra un 2-clique, si no es posible, entonces hallar un vértice que complete un 3-clique con uno de los 2-cliques que ya se tienen.

Lo anterior limitará el espacio de búsqueda de los  $K_2$  necesarios o  $K_2$  mas  $K_3$  o  $K_2$  mas  $K_3$  mas  $K_4, \dots$ , que den la coloración más robusta en  $G$ .

**Ejemplo 4.3** *Se quiere encontrar la  $c$ -coloración más robusta, para  $c = 4, 5, 6, 7, 8$ , en el grafo cuyos parámetros son los siguientes:*

$$n = 9, \quad H = \begin{pmatrix} 0 & .0058 & 0 & .0265 & 0 & .0213 & .0251 & 0 & .0105 \\ 0 & 0 & 0 & .0062 & 0 & 0 & 0 & .0034 & .0024 \\ 1 & 1 & 0 & 0 & .0084 & 0 & 0 & .0100 & .0072 \\ 0 & 0 & 1 & 0 & .0130 & 0 & 0 & 0 & .0111 \\ 1 & 1 & 0 & 0 & 0 & .0104 & .0123 & .0071 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & .0089 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & .0061 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \end{pmatrix}$$

*Las aristas complementarias ordenadas de menor a mayor según su peso son:*

$$\{2, 9\}\{2, 8\}\{1, 2\}\{8, 9\}\{2, 4\}\{5, 8\}\{3, 9\}\{3, 5\}\{6, 9\}$$

$$\{3, 8\}\{5, 6\}\{1, 9\}\{4, 9\}\{5, 7\}\{4, 5\}\{1, 6\}\{1, 7\}\{1, 4\}$$

*Si  $c$  es 6, se puede encontrar una cota inferior para la robustez usando la coloración más robusta con 7 colores, y a partir de  $\{5, 8\}$  se encuentra un  $K_2$ ,  $\{1, 6\}$ , sin intersección con  $\{2, 9\}$  ni con  $\{5, 8\}$ ; que junto con dichos cliques permite una coloración con 6 colores.*

Al intentar mejorar esta coloración, la búsqueda de los cliques necesarios se reduce a las aristas que están entre el inicio de la ordenación y la arista  $\{1, 6\}$ .

Las coloraciones más robustas para los diferentes valores de  $c$  son:

$c$	$n_3$	$n_2$	$n_1$	$cliques$	$C_R^c$									$R(C_R^c)$
					1	2	3	4	5	6	7	8	9	
8	0	1	8	$\{2, 9\}$	2	1	3	4	5	6	7	8	1	.0024
7	0	2	5	$\{2, 9\}$ $\{5, 8\}$	3	1	4	5	2	6	7	2	1	.0095
6	0	3	3	$\{1, 2\}$ $\{5, 8\}$ $\{3, 9\}$	1	1	3	4	2	5	6	2	3	.0201
5	0	4	1	$\{1, 2\}$ $\{6, 9\}$ $\{3, 8\}$ $\{5, 7\}$	1	1	3	5	4	2	4	3	2	.0370
4	1	3	0	$\{3, 8, 9\}$ $\{2, 4\}$ $\{5, 7\}$ $\{1, 6\}$	4	2	1	2	3	4	3	1	1	.0631

## 4.2 Algoritmo híbrido genético-voraz

Un algoritmo genético puede considerarse un algoritmo de búsqueda probabilística "inteligente" que puede aplicarse a varios problemas de optimización combinatoria.

Las bases teóricas de los algoritmos genéticos fueron desarrolladas por Holland [33]; la idea está basada en el proceso evolutivo de los organismos biológicos en la naturaleza. Durante el curso de la evolución las poblaciones naturales evolucionan de acuerdo a los principios de la selección natural en que sobreviven y se reproducen los más aptos. Lo que significa que los *genes* de los individuos mejor adaptados se transmitirán a los individuos de generaciones posteriores.

La combinación de buenas características de ancestros altamente adaptados puede producir descendientes aún mejor adaptados. De esta manera, las especies evolucionan y están cada vez mejor adaptadas a su medio ambiente.

Los algoritmos genéticos intentan imitar matemáticamente algunos procesos adaptativos del fenómeno biológico, considerando una población inicial de individuos y aplicando

operadores genéticos en cada reproducción, con dichos operadores se define un mecanismo de búsqueda sin necesidad de imponer restricciones matemáticas adicionales. En términos de optimización, cada individuo de una población es codificado en una cadena o *cromosoma* que representa una posible solución de un problema dado.

El ajuste de un individuo es evaluado con respecto a una función objetivo dada. A los individuos mejor adaptados o mejores soluciones, se les permite reproducirse intercambiando algunos elementos de su información genética en un proceso de cruce con otros individuos también muy aptos. Esto produce nuevas soluciones, que comparten algunas características tomadas de sus padres.

Frecuentemente se aplica la mutación después del cruce alterando algunos genes en las cadenas. Esto como ya se mencionó, con el fin de mejorar el ajuste de un conjunto de soluciones iniciales o soluciones de la población inicial.

Los descendientes pueden reemplazar a la población total, enfoque generacional (*generational approach*), o reemplazar a los individuos menos aptos, enfoque de estado uniforme (*steady-state approach*). Este ciclo de evaluación-selección-reproducción se repite hasta encontrar una solución satisfactoria, para ampliar la información, véase [28], [33], [58], [54] y [1].

Según Hong *et al.*[34], un algoritmo genético estándar se puede resumir en la siguiente tabla:

**inicio**

Representar adecuadamente las soluciones del Problema

Generar población inicial con individuos o soluciones del Problema

Definir una función de aptitud de los individuos de la población

**do while** (NO se haya satisfecho el criterio de parada)

Elegir pares de individuos como padres

Cruzar los padres elegidos para obtener dos hijos

Reemplazar los padres elegidos por sus hijos

Mutar algunas características de los individuos de la generación

Evaluar la aptitud de los individuos de la población

Seleccionar los individuos que sobreviven en la siguiente generación

**enddo**

**fin**

A diferencia de otras heurísticas, como temple simulado o búsqueda tabú, los algoritmos genéticos manejan simultáneamente un conjunto de soluciones (individuos) en cada etapa (la generación). Esta característica es compartida con una heurística de Fred Glover, llamada búsqueda dispersa (scatter search), [27] y [54]. Las características de cada solución deben ser codificadas adecuadamente (el cromosoma) de forma que al combinar dos soluciones para producir una nueva solución, parte de estas características (los genes) se transmitan a ésta.

Los algoritmos genéticos se han utilizados combinados con algoritmos voraces [20], [24] o con otras metaheurísticas [24]; para hallar soluciones aproximadas del problema de coloración mínima. Los algoritmos genéticos han permitido encontrar permutaciones de los vértices para posteriormente colorearlos con un método voraz.

A continuación se detallan los pasos de la construcción de un algoritmo genético para el *PCR*.

El primer parámetro de la metaheurística es el tamaño de la población, que identifica el número  $m$  de soluciones que se gestionan simultáneamente. Cuanto mayor sea  $m$  mayor será la diversificación de la búsqueda pero será más lento el proceso. Tamaños apropiados pueden variar entre 10 y 50 individuos.

### 4.2.1 Representación y función de ajuste

Cada uno de los individuos de la población o soluciones del problema debe ser representado a partir de un esquema que permita, además, valorar la aptitud del individuo o valoración de la función objetivo a maximizar (inverso de la penalización en el caso de minimización).

Un esquema natural sería el que representa cada solución por un vector con  $n$  componentes  $\mathbf{x}^t = (x_1, x_2, \dots, x_n)$  donde cada  $x_i$  indica el color asignado al vértice  $i$ . Cada vector corresponde a una solución. El ajuste de un elemento o individuo se representa por la función objetivo en que se intenta minimizar la rigidez de la coloración

$$R(\mathbf{x}) \equiv \sum_{\{i,j\} \in \bar{E}, x_i = x_j} p_{ij}$$

donde  $p_{ij}$  es la penalización de la arista complementaria  $\{i, j\}$ ; como ya se dijo, la coloración menos rígida es la más robusta.

No obstante, la obtención de coloraciones válidas con esta representación sería muy difícil, por lo que el esquema se basará en la ordenación de los vértices para, siguiendo esta ordenación, colorear vorazmente con  $c$  colores el grafo.

Así pues, el conjunto de soluciones será el conjunto de permutaciones de  $n$  vértices. Cada una de ellas se identificará por el vector de  $n$  componentes  $(i_1, \dots, i_n)$ .

Habrà, por tanto,  $n!$  ordenaciones posibles de los vértices que daràn lugar, con el correspondiente algoritmo de coloración aplicado a esa ordenación, una coloración del grafo.



### 4.2.2 Algoritmo de coloración robusta voraz

Fijado el número de colores permitidos  $c$ , el algoritmo tratará de repartir estos colores uniformemente entre los vértices para permitir una coloración válida. La coloración dependerá del orden de los vértices, por lo que no se garantiza la validez de la coloración pudiendo necesitar más colores de los  $c$  permitidos.

El esquema de este algoritmo es el siguiente:

```

Sea  $kc$  el último color utilizado, inicialmente  $kc = 0$ 
do  $j = 1, n$ 
    Sea el  $k > kc$  (cíclicamente en  $\{1, \dots, c\}$ ) el menor color válido para  $i_j$ 
    (Si existe tal color,  $x_j = C(i_j) = k, kc = k$ )
    (Si no existe tal color,  $x_j = C(i_j) \in \{c, c + 1, \dots\}$ )
enddo
    
```

Algoritmo voraz robusto

Sea  $\mathbf{x}^{(i_1, \dots, i_n)}$  la coloración obtenida a partir de la permutación  $(i_1, \dots, i_n)$ , el conjunto de coloraciones es, entonces,

$$X = \{ \mathbf{x}^{(i_1, \dots, i_n)} / (i_1, \dots, i_n) \in P_n \}$$

Una cuestión que se plantea es si la coloración robusta  $C_R$  se puede obtener con el algoritmo voraz y una ordenación adecuada de vértices, como ocurría en el problema de coloración mínima. En caso afirmativo, existiría una ordenación  $(i_1^R, \dots, i_n^R) \in P_n$  tal que

$$C_R(i) = x_i^{(i_1^R, \dots, i_n^R)} \quad \forall i \in \{1, \dots, n\}$$

Sin embargo, y a diferencia del Problema de Coloración Mínima, en el que una coloración mínima puede ser obtenida a partir de una ordenación adecuada de vértices y el algoritmo voraz, en el *PCR* no es cierto este resultado, como se puede comprobar en el siguiente ejemplo.

**Ejemplo 4.4** Sea el PCR caracterizado por los siguientes parámetros:

$$n = 4, \quad c = 2, \quad H = \begin{pmatrix} & .1 & .1 & .3 \\ 0 & & .1 & .3 \\ 0 & 0 & & .3 \\ 0 & 0 & 0 & \end{pmatrix}$$

Es fácil comprobar que la 2-coloración más robusta es

$$C_R(1) = 1 \quad C_R(2) = 1 \quad C_R(3) = 1 \quad C_R(4) = 2 \quad R(C_R) = 0.3$$

Sin embargo, al aplicar el algoritmo de coloración voraz robusto a cualquier ordenación de vértices, al no tener el grafo aristas, se obtiene siempre una coloración  $C$  verificando:

$$|V_C(1)| = |V_C(2)| = 2 \quad R(C) = 0.4$$

No obstante, para grafos con un número suficiente de aristas es posible encontrar una ordenación de vértices apropiada de forma que la coloración voraz robusta pueda aproximar (si no obtener) la coloración más robusta.

En cualquier caso, dada una solución  $\mathbf{x} = x^{(i_1, \dots, i_n)}$  asociada a una ordenación de vértices, se penaliza su invalidez en el caso en que se hayan necesitado más de  $c$  colores, por lo que se define la función  $Iv(\mathbf{x})$  como el número de colores adicionales a  $c$  que se han utilizado. Es decir,  $Iv(\mathbf{x}) = 0$  si y sólo si la coloración es válida.

Su rigidez se determina por

$$R(\mathbf{x}) \equiv \sum_{\{i,j\} \in \bar{E}, x_i = x_j} p_{ij}$$

Inicialmente, se seleccionan aleatoriamente  $m$  ordenaciones de  $n$  vértices para obtener la generación inicial

$$P^0 = (\mathbf{x}^1, \dots, \mathbf{x}^m)$$

de coloraciones.

Este valor  $m$ , que indica el tamaño de la población, es el primer parámetro de la heurística.

Los vectores

$$(Iv(\mathbf{x}^1), \dots, Iv(\mathbf{x}^m)) \quad (R(\mathbf{x}^1), \dots, R(\mathbf{x}^m))$$

caracterizan la aptitud de cada uno de los individuos de la población inicial a partir de la combinación ponderada de la robustez y la invalidez de la coloración según:

$$A(\mathbf{x}^i) = \frac{1}{R(\mathbf{x}^i) + P \cdot Iv(\mathbf{x}^i)} \quad \forall i \in \{1, \dots, m\}$$

siendo  $P > 0$  una constante de ponderación que penaliza el uso de más de  $c$  colores. Cuando el número de colores utilizado no supere a  $c$ , la aptitud de la coloración es el inverso del grado de robustez. Hay que asignar a este parámetro un valor suficientemente grande con el fin de obtener coloraciones válidas; un valor razonable para penalizaciones de las aristas del orden de la unidad es  $P = 100000$ .

A partir de estos vectores se puede ponderar globalmente la población inicial según:

$$A^0 = \sum_{i=1}^m A(\mathbf{x}^i)$$

### 4.2.3 Operador de selección

Con el fin de permitir que sólo aquellos individuos más aptos sobrevivan a la siguiente generación, los algoritmos genéticos introducen un operador de selección para elegir aquellas soluciones mejores con una probabilidad mayor que aquéllas con una peor aptitud.

Dada una población en la iteración  $k$ , el individuo  $i$ -ésimo de dicha población será seleccionado para formar parte de la población de la iteración  $k + 1$  con una probabilidad:

$$p_i^k = \frac{A(\mathbf{x}^i)}{A^k} \quad \forall i \in \{1, \dots, m\}$$

Al seleccionar  $m$  individuos a partir de esta distribución de probabilidad se pueden repetir algunos de ellos.

Repetiendo este proceso indefinidamente se obtendría a partir de una determinada iteración  $k$  una población uniforme que se iría repitiendo y en la que todos los individuos serían iguales. Este individuo sería uno de la población inicial y probablemente sería el individuo de máxima aptitud.

Con el fin de diversificar la búsqueda de soluciones, se introduce en los algoritmos genéticos el siguiente operador.

#### 4.2.4 Operador de cruce

Cada dos individuos de una generación son elegidos con una determinada probabilidad  $p_c$  como progenitores que se cruzan para obtener dos soluciones que participarán de características de los progenitores. Los hijos sustituirán a los padres en la población. Se usan los términos *padre* y *madre* para ser congruentes con el fenómeno biológico que se imita.

Dependiendo del valor de  $p_c$ , tercer parámetro de la metaheurística, se conseguirá una población estable con valores bajos de  $p_c$  al ser sustituidos muy pocos pares de individuos por sus hijos; por otra parte, valores muy altos de  $p_c$  proporcionarán una población muy inestable al ser sustituidos los padres por sus hijos.

Los valores centrales de  $p_c$ , entre 0.3 y 0.7, consiguen un equilibrio entre los extremos anteriores.

Una forma de obtener dos hijos a partir de los padres, caracterizados por las ordenaciones de vértices  $(i_1^1, \dots, i_n^1)$  e  $(i_1^2, \dots, i_n^2)$ , es la siguiente:

1. Seleccionar al azar dos números  $j_1$  y  $j_2$  verificando  $1 \leq j_1 < j_2 \leq n$ .
2. El primer hijo se caracteriza por la ordenación del padre entre los elementos  $j_1$  y  $j_2$ .
3. Fuera de este intervalo, el primer hijo tendrá los elementos de la madre si ha sido utilizado el índice del padre, en cuyo caso se toma el siguiente índice.
4. El segundo hijo se construye dualmente a partir de los índices centrales de la madre y seleccionando los otros del padre.
5. Los hijos sustituyen a los padres.

*Por ejemplo, sea  $n = 7$  y sean las dos ordenaciones de vértices*

$$(3, 5, 4, 2, 6, 1, 7) \quad (6, 3, 4, 1, 7, 2, 5)$$

*asociadas a los padres seleccionados.*

*Si  $j_1 = 3$  y  $j_2 = 5$ , las ordenaciones de vértices que sustituirán a las dos anteriores son*

$$(1, 7, 4, 2, 6, 5, 3) \quad (2, 6, 4, 1, 7, 3, 5)$$

#### 4.2.5 Operador de mutación

Una vez que los padres elegidos han sido sustituidos por sus hijos, y con el fin de diversificar la búsqueda permitiendo explorar soluciones que nunca podrían ser obtenidas por las limitaciones de los individuos inicialmente generados, se modifican aleatoriamente y sin seguir ningún patrón, algunas características de la generación obtenida con los operadores de selección y cruce. Es el operador de mutación.

La probabilidad de modificar alguna de las características de cada uno de los individuos de la generación o población en cada iteración,  $p_m$ , debe ser pequeña con el fin de asegurar la mejora de la aptitud de cada generación respecto de las anteriores. Un valor alto

provocaría una gran inestabilidad de la población. Un valor aceptable puede ser del orden de  $p_m = 0.05$ .

El operador de mutación se aplica generando para cada ordenación que ha sido elegida dos vértices aleatoriamente, el orden de estos dos vértices es cambiado.

### 4.3 Experiencias computacionales

Se han diseñado dos baterías de experimentos para validar los algoritmos aproximados descritos en este capítulo. En la primera batería se han comparado las heurísticas con el modelo de programación binaria introducido en el capítulo anterior. Se han resuelto los problemas generados aleatoriamente y resueltos por el modelo de programación (MP) descritos en la tabla de experiencias computacionales del capítulo 3 con el algoritmo de cliques (clique) y con el algoritmo basado en algoritmos genéticos (GEN). Los resultados se detallan en la tabla 1.

Los problemas a resolver en esta batería están, por consiguiente, limitados en su tamaño. Con el fin de comprobar los algoritmos en problemas de mayor tamaño, se han generado otros grafos aleatorios que han sido resueltos por las dos heurísticas anteriores. Los resultados se detallan en la tabla 2.

Fichero	n	c	MP	clique	GEN
al104	10	4	<u>3.8458</u>	4.6624	4.4005
al105	10	5	<u>2.1641</u>	2.3698	3.4005
al114	11	4	<u>3.9532</u>	<u>3.9532</u>	6.2907
al115	11	5	<u>2.6373</u>	2.6684	4.6809
al124	12	4	<u>3.8274</u>	—	<u>3.8274</u>
al125	12	5	4.634	3.659	<u>2.6904</u>
al135	13	5	<u>4.9117</u>	5.657	5.8056
al145	14	5	<u>5.0689</u>	6.1605	5.5329
al155	15	5	7.5935	<u>6.5979</u>	<u>6.5979</u>
al156	15	6	—	<u>5.069</u>	9.7643

Tabla 1. Experiencias computacionales

La tabla muestra la rigidez obtenida en cada uno de los problemas con el modelo de programación binaria (MP) y las heurísticas basadas en los cliques (clique) y en los algoritmos genéticos (GEN). La heurística de cliques no ha sido capaz de encontrar una coloración válida en el problema codificado como *al124*.

Para cada problema se ha subrayado la mejor solución obtenida. Se pone de manifiesto así que para ejemplos pequeños no hay una heurística —el modelo de programación se puede considerar así al tener limitado el número de nodos— que domine claramente al resto.

A continuación se ilustra en la siguiente tabla las experiencias computacionales con problemas de gran tamaño, en los que los grafos alcanzan hasta 1000 vértices.

Fichero	n	c	clique	GEN
al50	50	17	14.71	19.87
al75	75	25	20.63	32.12
al100	100	34	21.93	42.71
al150	150	50	33.90	67.05
al200	200	67	39.46	125.21
al250	250	84	50.35	117.86
al500	500	200	—	407.0
al1000	1000	300	—	1233.0

Tabla 2. Experiencias computacionales

Se observa que la heurística de cliques proporciona mejores resultados que la basada en algoritmos genéticos. Sin embargo, ésta permite abordar problemas de mayor tamaño y es más general al permitir valores de  $c < \frac{n}{3}$ , lo que es imposible para la heurística de cliques al trabajar sólo con cliques de orden 3 como máximo.

El problema *al1000* tiene asociado un grafo con  $n = 1000$  y un número de colores  $c = 300$ . Este problema no ha podido ser resuelto con la heurística de cliques al superar el límite del número de aristas complementarias. Ha sido generado con todas las penalizaciones iguales a 1.0 para que pueda ser evaluada la heurística basada en algoritmos genéticos. Concretamente, con dichos valores de  $n$  y  $c$  el valor mínimo que puede alcanzar la rigidez —sin considerar la topología concreta del grafo— es 1200, que se obtendría con  $n_4 = 100$  y  $n_3 = 200$ , cliques de orden 4 y 3 respectivamente. El error cometido está acotado, pues, en un 2.75%.

Se puede concluir inicialmente, y a falta de una experimentación sistemática y más extensa, que para valores mayores de  $c$  —y siempre imponiendo que  $c \geq \frac{n}{3}$ — se comporta mejor la heurística de cliques mientras que para valores menores de  $c$ , y sin limitación alguna siempre que  $c \geq \chi(G)$ , se comporta mejor la heurística basada en algoritmos genéticos.





## Capítulo 5

# El Problema de Coloración Robusta Generalizado

### 5.1 Planificación con restricciones de tiempo

Cuando se plantea un problema de planificación con un conjunto de recursos limitados y en un calendario concreto, es natural plantearse restricciones análogas a las vistas en la planificación de exámenes de la sección 1.2.1. Se identifica así un grafo donde los vértices son los elementos a planificar —las asignaturas en el problema de la sección 1.2.1— y las aristas unen los elementos que no pueden compartir el mismo recurso. La coloración identificará el recurso asignado.

Puede ocurrir, además, que la programación de los eventos esté limitada por restricciones temporales del tipo "dos eventos no pueden ser programados en el mismo día", o "debe haber al menos dos días entre dos eventos de un mismo tipo"; también con restricciones de espacio como "no pueden programarse más de  $k$  eventos en cada hora", por disponer de un número limitado de espacios. En estos casos, el problema de coloración con el grafo anterior no resuelve el problema; no obstante, un nuevo problema de coloración de grafos permitirá abordarlos.

Para ilustrar este problema, se introduce el siguiente ejemplo.

**Ejemplo 5.1** *Se desea planificar la docencia de un título universitario que consta de 15 asignaturas distribuidas en 3 cursos. Cada asignatura se imparte en 1, 2 ó 3 horas semanales según consta en la tabla adjunta:*

CURSO	ASIGNATURAS	HORAS/SEMANA
I	A, B, C	3
	K	1
II	D, E	3
	F, G	2
III	H, I, J	2
	L, M, N, O	1

*Por otra parte, para el citado título universitario se han reservado 3 horas cada uno de los 5 días lectivos de la semana.*

*El problema consiste en asignar, si es posible, las 30 horas de las 15 asignaturas entre las 15 horas disponibles por semana. A la restricción natural que evita la coincidencia en una misma hora de más de una clase de un mismo curso, correspondientes a la misma asignatura o distintas, se añade la restricción de que dos clases de una misma asignatura no pueden programarse en el mismo día ni en días consecutivos.*

*Obviamente, el número de clases que se pueden impartir en cada hora de las 15 disponibles no podrá superar el número de 3, puesto que en caso contrario, se impartiría más de una asignatura de un mismo curso.*

*Se introducen 30 vértices*

$$A_1^I, A_2^I, A_3^I; B_1^I, B_2^I, B_3^I; \dots; N_1^{III}; O_1^{III}$$

*que identifican cada clase  $X_i^k$  por la asignatura  $X \in \{A, B, \dots, O\}$ , el subíndice de la clase asociada  $i \in \{1, 2, 3\}$ , y el superíndice del curso  $k \in \{I, II, III\}$ .*

*El color de cada clase identificará la hora y el día reservados para impartirla. Habrá, por tanto, 15 colores válidos.*

*Sin embargo, las restricciones de tiempo no pueden ser contempladas al modelizar este tipo de problemas con el esquema de la sección 1.2.1. El color distinto de dos elementos no garantiza que se verifiquen las restricciones de tiempo exigidas. Dos colores distintos, asignados a dos clases de la misma asignatura, pueden asociarse a dos horas consecutivas de un mismo día.*

No obstante, el grado de rigidez de una coloración válida permitirá distinguir éstas de forma que se fuerce una solución que contempla las restricciones de tiempo en este tipo de problemas de planificación de horarios.

*Una definición natural de las aristas del ejemplo 5.1 sería aquella que relaciona todas las clases de un mismo curso, para evitar ser coloreadas con el mismo color, que equivale a asignarles la misma hora y día.*

*Sin embargo, esta definición no permitiría distinguir las incompatibilidades temporales de las clases de una misma asignatura, puesto que, o bien no se contemplan, o bien forzarían esas restricciones para todas las clases de un mismo curso.*

*Con esta idea, las 15 horas consecutivas de los cinco días de la semana se asocian ordenadamente a los colores: el color 1 para la primera hora del lunes, el color 2 para la segunda hora del lunes, ..., el color 15 para la tercera hora del viernes. Se identifica así el parámetro del número de colores válidos  $c = 15$ .*

*Por otra parte, y aprovechando la flexibilidad que incorpora en los problemas de coloración el grado de rigidez, se relaja el grafo al suprimir las aristas que unen las clases de una misma asignatura y se mantienen las aristas que unen las clases de un mismo curso pero de distinta asignatura.*

*Obviamente, y para garantizar que no sean coloreadas igualmente dos clases de la misma asignatura, deben ser penalizadas fuertemente este tipo de aristas del grafo com-*

plementario.

*En conclusión, el conjunto de aristas es*

$$E = \{\{X_i^k, Y_j^k\} / k \in \{I, II, III\}, X \neq Y, \forall i, j\}$$

*definiendo así tantas componentes conexas como cursos, y siempre que cada curso tenga al menos dos asignaturas. En el ejemplo, hay 3 componentes conexas.*

*Para el problema de coloración robusta interesa precisar cuál es el conjunto de aristas complementarias  $\bar{E}$ , que está compuesto de dos tipos de aristas complementarias:*

- $\{X_i^k, X_j^k\}$  para todo  $k \in \{I, II, III\}$  y  $\forall X^k$  verificando  $i \neq j$ . Estas aristas están fuertemente penalizadas para evitar que sean coloreadas igualmente.

*Esta penalización fuerza la incompatibilidad real de las clases de una misma asignatura.*

- $\{X_i^k, Y_j^l\}$  tales que  $k \neq l \in \{I, II, III\}$  y  $\forall i, j$ . Estas aristas tienen una penalización mínima, 1 por ejemplo.

*Con esta penalización se consigue distribuir las clases uniformemente a las horas disponibles evitando, por ejemplo, que todas las clases compatibles se impartieran a primera hora del lunes.*

Con este planteamiento, sin embargo, no se puede resolver el problema como un problema de coloración robusta, según se ha estudiado en el capítulo 2, puesto que el grado de rigidez de una coloración se incrementa por cada arista complementaria cuyos extremos estén igualmente coloreados, cuando en realidad hay que incrementar el grado de rigidez cuando los colores estén suficientemente próximos para indicar que las correspondientes clases han sido asignadas muy próximas en el tiempo.

La solución a este problema pasa por considerar la penalización de las aristas complementarias cuando sus extremos estén coloreados con colores muy próximos, en el ejemplo 5.1 los colores próximos identifican asignación temporal próxima.

Con el fin de poder modelizar problemas como el descrito anteriormente en el ejemplo 5.1, se propone un nuevo problema de grafos que generaliza el *PCR*, por lo que se denominará Problema de Coloración Robusta Generalizado, y se denotará por *PCRG*.

## 5.2 Planteamiento del Problema

Con el fin de graduar la disimilitud de los colores, se utilizará una distancia entre los mismos. Recordando el concepto de distancia, ver Mardia *et al.* [48]:

**Definición 5.1** *Dado un conjunto no vacío  $C$ , se define una distancia como una aplicación*

$$d : C \times C \implies \mathbb{R}$$

*verificando las siguientes propiedades:*

1. *Simetría:*

$$d(x, y) = d(y, x) \quad \forall x, y \in C$$

2. *No negatividad:*

$$d(x, y) \geq 0 \quad \forall x, y \in C$$

3. *Identidad:*

$$d(x, x) = 0 \quad \forall x \in C$$

se introduce la siguiente definición:

**Definición 5.2** *Dada una  $c$ -coloración  $C^c$ , se define una  $c$ -distancia  $d$  como una distancia en el conjunto de colores  $\{1, 2, \dots, c\}$*

Un ejemplo de  $c$ -distancia es la trivial:

$$d^0(k, k') = \begin{cases} 0 & k = k' \\ 1 & k \neq k' \end{cases} \quad \forall k, k' \in \{1, 2, \dots, c\}$$

Otra  $c$ -distancia puede ser la basada en el valor absoluto de la diferencia de los colores:

$$d^1(k, k') = |k - k'| \quad \forall k, k' \in \{1, 2, \dots, c\}$$

A partir de una  $c$ -distancia  $d$  y de una cota  $\hat{d} \geq 0$  se define el grado de rigidez generalizado:

**Definición 5.3** *Fijada una  $c$ -distancia de colores  $d$ , a partir del grafo  $G$  y conocida la familia de penalizaciones  $\{p_{ij} \geq 0, \{i, j\} \in \overline{E}\}$ , se define el  $\hat{d}$ -grado de rigidez generalizado de la  $c$ -coloración  $C^c$ , siendo  $\hat{d} \geq 0$ , a la suma de las penalizaciones de las aristas complementarias cuyos extremos están coloreados con colores que tienen una distancia menor o igual  $\hat{d}$ . Se denota por  $R^{\hat{d}}(C^c)$  :*

$$R^{\hat{d}}(C^c) \equiv \sum_{\{i, j\} \in \overline{E}, d(C^c(i), C^c(j)) \leq \hat{d}} p_{ij}$$

El Problema de Coloración Robusta es un caso particular de  $PCRG$ , en el que la  $c$ -distancia es  $d^0$  y  $\hat{d} = 0$ .

### 5.3 Modelo de programación matemática

Con el fin de incorporar la  $c$ -distancia entre colores

$$d : \{1, 2, \dots, c\} \times \{1, 2, \dots, c\} \longrightarrow \{0, 1, 2, \dots\}$$

y la cota o umbral  $\hat{d}$  en el modelo de programación matemática (lineal) del  $PCRG$ , se introduce la matriz  $M$  de dimensión  $c \times c$  definida por:

$$m_{kk'} = \begin{cases} 1 & d(k, k') > \hat{d} \\ 0 & d(k, k') \leq \hat{d} \end{cases} \quad \forall k, k' \in \{1, \dots, c\}$$

Por ejemplo, fijado  $c = 4$ , si la 4-distancia es  $d^0$ , la distancia trivial, y el umbral es  $\hat{d} = 0$ , la matriz  $M$  es:

$$M = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

Con la 4-distancia  $d^1$  basada en el valor absoluto de la diferencia, con  $\hat{d} = 0$  y  $\hat{d} = 1$  se tienen las siguientes matrices:

$$M = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \quad M' = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

Una vez incorporados los datos de la  $c$ -distancia y la cota  $\hat{d}$  en la matriz  $M$ , el *PCRG* queda caracterizado por el grafo  $G = (V, E)$ , el número de colores válido  $c$  y las penalizaciones  $\{p_{ij}, \{i, j\} \in \overline{E}\}$ .

Las variables de decisión asociadas a la coloración  $C^c$  están definida por:

$$x_{ik} = \begin{cases} 1 & C^c(i) = k \\ 0 & C^c(i) \neq k \end{cases} \quad i \in \{1, \dots, n\}, k \in \{1, \dots, c\}$$

Las restricciones del modelo de programación son las siguientes:

- La coloración está bien definida en todos los vértices:



$$\sum_{k=1}^c x_{ik} = 1 \quad \forall i \in \{1, \dots, n\}$$

- La coloración es válida:

$$x_{ik} + x_{jk} \leq 1 \quad \forall \{i, j\} \in E \quad \forall k \in \{1, \dots, c\}$$

Con el fin de identificar las aristas complementarias que unen vértices con coloraciones suficientemente cercanas, se introducen las variables auxiliares:

$$y_{ij} = \begin{cases} 1 & \text{si } d(C^c(i), C^c(j)) \leq \hat{d} \quad \{i, j\} \in \overline{E} \\ 0 & \text{si } d(C^c(i), C^c(j)) > \hat{d} \quad \{i, j\} \in \overline{E} \end{cases} \quad \forall \{i, j\} \in \overline{E}$$

Es decir,  $y_{ij} = 1$  si y sólo si los extremos de la arista complementaria  $\{i, j\} \in \overline{E}$ , con colores  $k = C^c(i)$  y  $k' = C^c(j)$ , verifican  $m_{kk'} = 0$ .

Con el fin de garantizar esta propiedad se introduce, por un lado, la siguiente familia de restricciones lineales:

$$x_{ik} + \sum_{k'=1}^c x_{jk'}(1 - m_{kk'}) \leq y_{ij} + 1 \quad \forall \{i, j\} \in \overline{E}, \quad \forall k \in \{1, \dots, c\}$$

de forma que si existen dos colores  $k, k'$  suficientemente próximos,  $m_{kk'} = 0$ , que colorean los extremos de una arista complementaria  $\{i, j\} \in \overline{E}$ , entonces el valor de las variables auxiliares son  $y_{ij} = 1$ .

Por otro lado, al considerar la función objetivo que penaliza las aristas complementarias con colores suficientemente próximos

$$R^{\hat{d}}(C^c) = \sum_{\{i, j\} \in \overline{E}} p_{ij} y_{ij}$$

al ser  $p_{ij} \geq 0$  garantizará el valor  $y_{ij} = 0$  excepto cuando  $i$  y  $j$  sean los extremos de aristas complementarias con colores suficientemente próximos.

Este modelo consta, si la densidad del grafo es 0.5, de  $n \times c + c \times c + \frac{1}{2} \cdot \binom{n}{2}$  variables. En este caso, el número de aristas complementarias es  $\frac{1}{2} \cdot \binom{n}{2}$  y el de restricciones es  $n \times c + \binom{n}{2} \times c$ .

Para el ejemplo 5.1, que tiene 115 aristas y 320 aristas complementarias y dado que  $c = 15$ , el modelo tiene 570 variables y 6555 restricciones el cual resulta demasiado grande para resolver de manera exacta.

Si en este modelo se hace  $\hat{d} = 0$ , entonces se tiene el modelo de programación binaria para el PCR, lo que demuestra que el PCR es un caso particular del PCRG. En efecto, si  $\hat{d} = 0$  no es necesario definir la matriz  $M$  y las variables  $y_{ij}$  valdrán uno si se usa el color  $k$  y cero en otro caso.

El problema de coloración robusta generalizado es muy flexible al permitir definir diferentes distancias entre los colores. En el problema de un horario diario, por ejemplo, que se repite sistemáticamente, puede interesar que las asignaciones de las primeras horas del día, los colores  $1, 2, \dots$ , sean incompatibles con las asignaciones de las últimas horas, los colores  $\dots, c - 2, c - 1, c$ , por lo que una  $c$ -distancia circular determinaría para una cota fija  $\hat{d}$  una matriz  $M$  del tipo

$$M = \begin{pmatrix} 0 & 0 & 0 & \cdots & 1 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 1 & 1 \\ \vdots & & \vdots & & \vdots & & \\ 1 & 1 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 & 0 \end{pmatrix}$$

Por otra parte, esta matriz  $M$  permite identificar situaciones más generales que las contempladas por una  $c$ -distancia definida en el conjunto de colores y una cota o umbral  $\hat{d}$ .

Concretamente, en el ejemplo 5.1, se podría fijar  $\hat{d} = 6$  y utilizar la distancia  $d^1$  para evitar que una asignatura se imparta en dos días consecutivos, el primero a última hora y el siguiente a primera hora. En consecuencia el modelo impuesto garantizaría las restricciones del problema aunque es más restrictivo.

Definiendo de forma apropiada una matriz  $M$ , se podría obtener un modelo menos restrictivo que el anterior y verificando todas las restricciones originales del problema. Esta matriz  $M_{15 \times 15}$  que tendría la siguiente estructura:

$$M = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

El grafo de 30 vértices, los asociados a las clases de las asignaturas, tiene 3 componen-

tes conexas, una por curso. Cada componente conexa tiene tantos vértices como clases y todos los vértices de las clases de cada asignatura del curso están unidos al resto de los vértices de las otras asignaturas; no están unidos los vértices correspondientes a clases de una misma asignatura.

Las aristas complementarias se penalizan de forma diferente según su origen:

- Las aristas complementarias uniendo clases de una misma asignatura se penalizan con una cantidad suficientemente grande; por ejemplo,  $p_{X_i^k, X_j^k} = 100$  para todo par de clases de una misma asignatura  $X_i^k, X_j^k$ .
- Las aristas complementarias que unen clases de asignaturas de distintos curso se penalizan con una cantidad positiva incomparablemente inferior a la anterior; por ejemplo,  $p_{X_i^k, Y_j^l} = 1$  para cualquier par de clases  $X_i^k, Y_j^l$  verificando  $k \neq l$ .

Algunas de las aristas del grafo que corresponde al ejemplo 5.1, son  $\{1,4\}$ ,  $\{11,18\}$ ,  $\{22,29\}$  y algunas aristas complementarias con penalización 100 son  $\{1,2\}$ ,  $\{1,3\}$ ,  $\{1,4\}$  y con penalización 1,  $\{5,11\}$ ,  $\{5,21\}$  y  $\{20,30\}$ ; el modelo de programación quedaría

$$\begin{aligned}
 \text{Min } R^{\hat{d}}(C^c) &= 100y_{1\ 2} + \dots + y_{5\ 21} + \dots + y_{20\ 30} \\
 \text{s.a} & \quad x_{i\ 1} + \dots + x_{i\ 15} = 1 && \forall i \\
 & \quad \dots && \forall k \\
 & \quad x_{1\ 1} + x_{4\ 1} \leq 1 \\
 & \quad \dots \\
 & \quad x_{1\ 15} + x_{4\ 15} \leq 1 \\
 & \quad \dots \\
 & \quad x_{20\ 1} + x_{30\ 1}(1 - m_{1\ 1}) + \dots + x_{30\ 15}(1 - m_{1\ 15}) \leq y_{20\ 30} + 1 \\
 & \quad \dots \\
 & \quad x_{20\ 15} + x_{30\ 1}(1 - m_{15\ 1}) + \dots + x_{30\ 15}(1 - m_{15\ 15}) \leq y_{20\ 30} + 1
 \end{aligned}$$

**Observación 5.1** Dada la similitud entre el PCR y el PCRG, y considerando la flexibilidad de los algoritmos genéticos, la adaptación de un algoritmo genético que resuelva el primer problema se realiza fácilmente de forma que pueda resolver el segundo.

La 15-coloración obtenida con un algoritmo genético es la que asigna los colores siguientes:

CURSO	CLASE	COLOR	CLASE	COLOR	CLASE	COLOR
<i>I</i>	$A_1$	1	$A_2$	7	$A_3$	13
	$B_1$	2	$B_2$	8	$B_3$	14
	$C_1$	3	$C_2$	9	$C_3$	15
	$K_1$	6				
<i>II</i>	$D_1$	1	$D_2$	7	$D_3$	13
	$E_1$	2	$E_2$	8	$E_3$	14
	$F_1$	4	$F_2$	10		
	$G_1$	5	$G_2$	11		
<i>III</i>	$H_1$	4	$H_2$	10		
	$I_1$	5	$I_2$	11		
	$J_1$	6	$J_2$	12		
	$L_1$	3				
	$M_1$	9				
	$N_1$	15				
	$O_1$	14				

Asignación de colores a las clases

Fijando las asignaturas a cada una de las horas de los días de la semana según el color asignado a las clases asociadas, se tiene la siguiente tabla:

<i>HORA</i>	<i>LUNES</i>	<i>MARTES</i>	<i>MIÉRCOLES</i>	<i>JUEVES</i>	<i>VIERNES</i>
1	<i>A(I)</i> <i>D(II)</i>	<i>F(II)</i> <i>H(III)</i>	<i>A(I)</i> <i>D(II)</i>	<i>F(II)</i> <i>H(III)</i>	<i>A(I)</i> <i>D(II)</i>
2	<i>B(I)</i> <i>E(II)</i>	<i>G(II)</i> <i>I(III)</i>	<i>B(I)</i> <i>E(II)</i>	<i>G(II)</i> <i>I(III)</i>	<i>B(I)</i> <i>E(II)</i> <i>O(III)</i>
3	<i>C(I)</i> <i>L(III)</i>	<i>K(I)</i> <i>J(III)</i>	<i>C(I)</i> <i>M(III)</i>	<i>J(III)</i>	<i>C(I)</i> <i>N(III)</i>

*Planificación semanal de las clases*

*La coloración así obtenida es válida al no coincidir en una misma hora de un día dos asignaturas del mismo curso. El grado de rigidez penaliza sólo la coincidencia de más de una clase en una misma hora, el valor obtenido para el grado de rigidez es igual a 16, no pudiéndose alcanzar el valor mínimo 15, que correspondería a distribuir las 30 clases uniformemente —dos por hora— entre las 15 horas disponibles en la semana.*



# Capítulo 6

## Enfoque Difuso de Problemas de Coloración

A partir del concepto de número difuso, y considerando que una función de coloración asigna números (colores) a los vértices de un grafo, es natural plantearse qué ocurre cuando el color asignado es un número difuso, introduciendo así la coloración difusa de un grafo nítido. También se puede uno plantear qué ocurre cuando la función de coloración es nítida pero el grafo es difuso; a partir de esta idea se plantea el problema de coloración mínima difusa y se introduce el concepto de número cromático difuso.

### 6.1 Coloración difusa de un grafo nítido

Una de las formas en que se puede representar a un conjunto  $A$  es usando el concepto de función característica  $\mu_A(\cdot)$  cuyo valor uno o cero indica si un elemento pertenece o no a  $A$ .

Sean los conjuntos  $V = \{v_1, v_2, v_3, v_4, v_5\}$  y  $A = \{v_1, v_3, v_5\}$ ; entonces se puede escribir  $\mu_A(v_1) = 1$ ,  $\mu_A(v_2) = 0$ ,  $\mu_A(v_3) = 1$ ,  $\mu_A(v_4) = 0$  y  $\mu_A(v_5) = 1$ , lo que permite describir al conjunto  $A$  con los elementos de  $V$  acompañados del valor de la función característica:



$A = \{(v_1, 1), (v_2, 0), (v_3, 1), (v_4, 0), (v_5, 1)\}$ . Los elementos con grado de pertenencia igual a cero normalmente no se escriben. Al considerar que la función característica puede tomar cualquier valor en el intervalo  $[0, 1]$ , en 1965 Zadeh introdujo el siguiente concepto de conjunto difuso, que generaliza al concepto clásico de conjunto.

**Definición 6.1** (Zadeh [70]) *Sea  $V$  un conjunto finito o infinito de elementos denotados genéricamente por  $v$ , entonces un subconjunto difuso  $D$  de  $V$  es el conjunto de parejas  $\{v, \mu_D(v)\}$ ,  $\forall v \in V$  donde  $\mu_D(v)$  es el grado de pertenencia de  $v$  a  $D$ . Se puede decir que  $\mu_D(v)$  toma sus valores de un conjunto  $M$  llamado conjunto de pertenencia, con la función  $\mu_D : V \rightarrow M$ . Esta función se llama también función de pertenencia.*

Si  $M = \{0, 1\}$ , entonces el subconjunto difuso  $D$  se vuelve un subconjunto *no difuso* o subconjunto común, aquí normalmente se considerará  $M = [0, 1]$ . En [38], se comenta una generalización del conjunto  $M$  realizada por J. A. Goghen.

Considerando que la función de coloración de un grafo asigna colores (números) a cada vértice, y que esta coloración implica una determinada asignación, es natural plantearse la cuestión de generalizar esta asignación permitiendo que el color sea un número difuso. La coloración difusa  $\tilde{C}$  así introducida debe asignar a cada vértice del grafo una graduación de cada uno de los colores seleccionados de un conjunto fijado previamente. Formalmente, se introduce la siguiente definición:

**Definición 6.2** *Dados un grafo  $G = (V, E)$  y el conjunto finito de colores  $\{1, 2, \dots, c\}$ , una coloración difusa es una aplicación*

$$\tilde{C}^c : V \rightarrow (\mu_{\tilde{C}^c}(1), \dots, \mu_{\tilde{C}^c}(c))$$

Es decir, cada color  $k$ ,  $k \in \{1, \dots, c\}$  está presente en la coloración de un vértice  $i \in V$  con una cierta intensidad  $\mu_{\tilde{C}^c}(k)$ .

*Por ejemplo, si  $c = 2$ , considerando los colores blanco y negro, una coloración difusa asignará diferentes intensidades a los colores 0 (blanco) y 1 (negro) y se obtienen las*

tonalidades de grises. Si  $c = 3$  y se seleccionan los tres colores básicos, se obtendrán con todas las intensidades posibles toda la gama cromática. En el procesador de textos LaTeX [46], los colores básicos utilizados son el rojo, el verde y el azul para obtener diferentes tonalidades asignando intensidades entre 0 y 1.

Los problemas de coloración (nítidos) se basan en la distinción de colores entre los extremos de las aristas del grafo. Surge entonces la siguiente cuestión al definir una coloración difusa: ¿cuándo dos colores difusos son iguales? Dos respuestas a esta pregunta son las siguientes:

1. Utilizando la definición de *igualdad de conjuntos difusos* de Kaufmann [38], se dirá que dos vectores de intensidades de color  $\mu_{\tilde{C}_1^c}$  y  $\mu_{\tilde{C}_2^c}$ , con  $c$  colores cada uno, son iguales si y solo si

$$\mu_{\tilde{C}_1^c}(k) = \mu_{\tilde{C}_2^c}(k) \quad \forall k \in \{1, \dots, c\}$$

es decir, si la intensidad del color  $k$  es la misma en ambas coloraciones, para todo  $k$ .

Si al menos un  $k \in \{1, \dots, c\}$  es tal que  $\mu_{\tilde{C}_1^c}(k) \neq \mu_{\tilde{C}_2^c}(k)$ , entonces los vectores de intensidades de color correspondientes a las coloraciones  $C_1^c$  y  $C_2^c$  son diferentes.

Esta propuesta es muy restrictiva y haría que el conjunto de colores fuera no numerable.

2. Una relación difusa  $x\tilde{\mathcal{R}}y$  con  $x, y \in \mathbb{R}^+$  y

$$\mu_{\tilde{\mathcal{R}}}(x, y) = e^{-F(y-x)^2}, \quad F > 1$$

de acuerdo con Zimmermann [74], se puede definir por la frase difusa *x e y están muy próximos*, para cierto valor de  $F$ .

A partir de esta relación, dos valores  $x$  e  $y$  están muy próximos, si  $\mu_{\tilde{\mathcal{R}}}(x, y)$  es mayor que un determinado nivel  $\alpha$ , por ejemplo,  $\alpha = 0.5$ .

Esta relación no es transitiva como lo demuestra Kaufmann [38].

Se dirá que dos vectores de intensidades de color, con  $c$  componentes cada uno, son equivalentes, lo que se escribirá como

$$\mu_{\tilde{C}_1^c} \simeq \mu_{\tilde{C}_2^c}$$

si  $\mu_{\tilde{C}_1^c}(k)$  está muy próximo de  $\mu_{\tilde{C}_2^c}(k)$  para toda  $k \in \{1, \dots, c\}$ ; es decir,

$$e^{-F(\mu_{\tilde{C}_1^c}(k) - \mu_{\tilde{C}_2^c}(k))^2} > \alpha \quad \forall k \in \{1, \dots, c\}$$

*Por ejemplo, considérese que los colores primarios amarillo, azul y rojo tienen respectivamente, las siguientes intensidades de color en las coloraciones*

$$\mu_{\tilde{C}_1^c} = (0.3, 0.9, 0.58) \quad \mu_{\tilde{C}_2^c} = (0.6, 0.72, 0.25)$$

*Entonces las dos coloraciones son equivalentes al nivel  $F = 5$ , ya que*

$$\mu_{\tilde{R}}(0.3, 0.6) = 0.6376 > 0.5 \quad \mu_{\tilde{R}}(0.9, 0.72) = 0.8504 > 0.5$$

*y*

$$\mu_{\tilde{R}}(0.58, 0.25) = 0.5801 > 0.5$$

*En cambio, los vectores de intensidades  $\mu_{\tilde{C}_1^c}$  y  $\mu_{\tilde{C}_3^c} = (0.8, 0.6, 0.3)$  no son equivalentes debido a que  $\mu_{\tilde{R}}(0.3, 0.8) = .2865 \not> 0.5$*

La generalización difusa del concepto de coloración de los vértices de un grafo es más amplia que la generalización *externa* presentada en la sección 1.4, pues todos los colores pueden estar presentes en cada vértice. Se ha presentado con carácter descriptivo aunque no se excluyen resultados interesantes al plantear problemas de asignación difusa como problemas de coloración.

## 6.2 Problema de Coloración Mínima Difusa

Se repasan algunos conceptos y notación de uso común necesarios para formalizar el planteamiento de este nuevo problema.

El soporte de  $D$  es  $S(D) = \{v \in V | \mu_D(v) > 0\}$ .  $D$  es no trivial si  $S(D)$  es no vacío. La altura de  $D$ ,  $a(D) = \max\{\mu_D(v) | v \in V\}$ ;  $D$  es normal si  $a(D) = 1$ .

**Definición 6.3** (Zadeh [70]) Si  $\mu_D(v) \in [0, 1]$ , se puede escoger un umbral  $\alpha \in (0, 1]$  y se define el conjunto

$$D_\alpha = \{v \in S(D) | \mu_D(v) \geq \alpha\}$$

como conjunto de corte al nivel  $\alpha$  o bien  $\alpha$  – corte.

**Definición 6.4** (Zadeh [70]) Un conjunto difuso  $D$  es convexo si

$$\mu_D(\lambda v_1 + (1 - \lambda)v_2) \geq \min\{\mu_D(v_1), \mu_D(v_2)\} \quad v_1, v_2 \in V \quad \lambda \in (0, 1]$$

Alternativamente, un conjunto difuso  $D$  es convexo si todos sus  $\alpha$ –cortes son convexos.

Un conjunto difuso también se puede definir según Dubois y Prade [23], especificando la familia  $\mathcal{C}(D) = \{D_\alpha | \alpha \in (0, 1]\}$  de  $\alpha$  – cortes que es monótona, *i.e.*

$$0 < \alpha \leq \beta \leq 1 \Rightarrow D_\alpha \supseteq D_\beta \tag{6.1}$$

que es una representación de  $D$  por medio de conjuntos clásicos ya que

$$\forall v \quad \mu_D(v) = \sup\{\alpha | v \in D_\alpha\} \tag{6.2}$$

Inversamente, una familia finita monótona de conjuntos  $\{G_{\alpha_1}, \dots, G_{\alpha_m}\}$  cuyos pesos  $\alpha_i$  satisfacen 6.1 forma el conjunto de  $\alpha$  – cortes de un conjunto difuso definido por 6.2. Para más propiedades de los  $\alpha$  – cortes véase Kerre [39].

Desde que en 1979 Mitsumoto y Tanaka [74], introdujeron el concepto de número difuso, han aparecido varios artículos sobre el tema por lo que se han propuesto muchas definiciones, Kerre presenta en [39] el concepto más general y llega al más específico agregando condiciones suplementarias sucesivamente.

**Definición 6.5** (Zimmermann [74]) *Un número difuso  $\tilde{M}$ , es un conjunto convexo normalizado de  $\mathbb{R}$  siempre que*

1. *Exista exactamente un  $x_0 \in \mathbb{R}$  tal que  $\mu_{\tilde{M}}(x_0) = 1$ .*
2.  *$\mu_{\tilde{M}}(x)$  es continua por partes.*

Cualquier número real  $r \in \mathbb{R}$  tal que  $\mu_{\tilde{M}}(r) = 1$  se llama *valor modal* de  $\tilde{M}$ . Si  $\tilde{M}$  es un número difuso con valor modal  $r$  entonces  $\tilde{M}$  es una representación posible de "casi  $r$ ".

La primera definición de grafo difuso la presentó Kaufmann en 1973, ver [38], a partir de las relaciones difusas propuestas por Zadeh en 1971 [71]. En 1975 Rosenfeld [61], presentó otra más elaborada, también basada en las relaciones difusas y es la que se utilizará en este capítulo.

Se considerarán grafos finitos no dirigidos, *i.e.*, se supone que las relaciones difusas son simétricas, lo que significa que la aristas se pueden ver como parejas no ordenadas de vértices.

**Definición 6.6** (Rosenfeld [61]) *Sea  $V = \{1, \dots, n\}$ , un grafo  $\tilde{G} = (V, \sigma, \mu)$  es un grafo difuso, donde*

1.  *$\sigma : V \rightarrow [0, 1]$ , que indica el nivel de pertenencia de cada vértice.*
2.  *$\mu : V \times V \rightarrow [0, 1]$ , que indica el nivel de pertenencia de cada arista.*

*Dados  $x, y \in V$ , si  $\mu(x, y) = 0$  entonces no existe la arista  $\{x, y\}$  y si  $\mu(x, y) > 0$  entonces existe la arista difusa  $\{x, y\}$ , siendo  $\mu(x, y)$  el grado de pertenencia de  $\{x, y\}$  al grafo  $\tilde{G}$ .*

Los grafos difusos considerados verifican que  $\sigma(v) = 1 \forall v \in V$ , que Kóczy [42] llama grafo con aristas difusas que se denota por  $\tilde{G} = (V, \mu)$ ; donde la imagen de  $\mu$  es un conjunto totalmente ordenado, no necesariamente el intervalo  $[0, 1]$ .

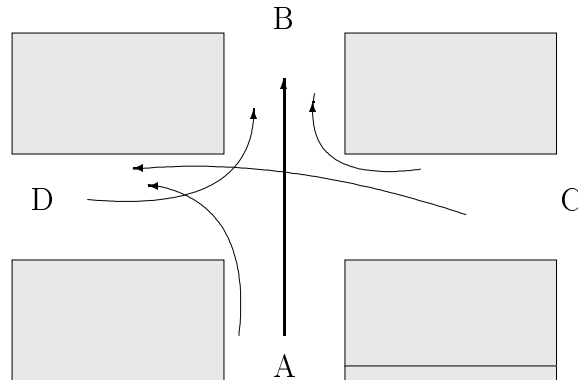


Figura 6.1: Cruce de Calles

**Definición 6.7** Sea el grafo con aristas difusas  $\tilde{G} = (V, \mu)$ , para cada  $\alpha$  que sea elemento de un conjunto totalmente ordenado, se define el  $\alpha$ -corte de  $\tilde{G}$  como el grafo nítido  $G_\alpha = (V, E_\alpha)$  donde  $E_\alpha = \{(i, j) \in E \mid \mu(i, j) \geq \alpha\}$ . Ya que los grafos a tratar tienen un conjunto finito de aristas, tendrán un número finito de  $\alpha$ -cortes diferentes.

A continuación se presenta un ejemplo del problema de coloración mínima y posteriormente se definirá el grafo difuso asociado.

**Ejemplo 6.1** Sea el cruce de cuatro calles ilustrado en la figura 6.1. Las flechas indican los sentidos posibles de circulación de los automóviles.

Se trata de regular el tráfico del cruce mediante semáforos que indiquen a los conductores cuándo pueden pasar. Hay que identificar el ciclo mínimo del semáforo que evite los choques, indicando en cada uno de los periodos de ese ciclo cuáles son los movimientos permitidos.

Este problema ha sido tratado por Golubic [29], que lo estudia como un grafo de intersección. Otros autores, ver Stoffers [62], Opsut y Roberts [51] y Roberts [59], lo tratan como un conjunto asignación.

En este capítulo se estudia como un problema de coloración mínima de los vértices de

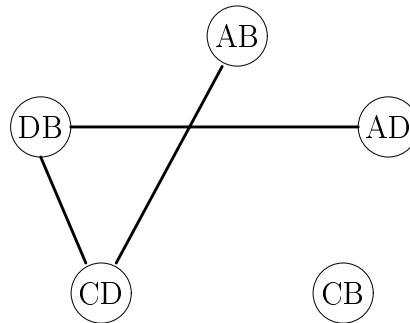


Figura 6.2: Grafo de incompatibilidad del ejemplo 6.1

un grafo  $G$ , cuyos vértices son cada uno de los sentidos de circulación posibles y habrá una arista entre dos vértices cuando sean incompatibles, i.e., cuando los sentidos de circulación correspondientes haya riesgo de colisión al permitir el paso a ambos sentidos por tener el semáforo en verde simultáneamente.

Los vértices son  $AB$ ,  $AD$ ,  $CB$ ,  $CD$  y  $DB$ ; como puede apreciarse en el diagrama del cruce, hay sentidos de circulación que son compatibles, i.e., que pueden realizarse simultáneamente sin que haya peligro de colisiones, tales como los sentidos  $AD$  y  $CB$ . También hay otros en los que existe un riesgo de colisión, es el caso de los sentidos  $AD$  y  $CD$ . El grafo de incompatibilidad  $G$  así definido se ilustra en la figura 6.2.

Una coloración del grafo, indicará con cada clase de color los trayectos que se pueden realizar simultáneamente y con esto, qué semáforos deben tener luz verde y cuáles luz roja.

La coloración de  $G$  es la siguiente:

$$C(AB) = 1, C(AD) = 2, C(CB) = 1 \text{ ó } 2, C(CD) = 2 \text{ y } C(DB) = 1$$

de donde se concluye que el sentido de circulación  $CB$  puede realizarse siempre, pues ser compatible con todos, admite cualesquiera de los dos colores. Los sentidos de circulación  $AB$  y  $DB$  tendrán el semáforo en luz verde cuando los sentidos de circulación  $AD$  y  $CB$  tengan luz roja en sus semáforos y viceversa.

### 6.2.1 Número cromático difuso

*El ciclo del semáforo en el ejemplo 6.1 depende de la definición de incompatibilidad de los sentidos de circulación. Este concepto admite graduaciones. Está claro que si en este ejemplo se consideran todos los giros incompatibles, el grafo de incompatibilidad es un grafo completo, por lo que se requieren 5 colores para pintarlos; lo que indica que debería haber en el semáforo un periodo de color verde para cada uno de los sentidos de circulación posibles, que no es una opción práctica. En el extremo opuesto, si todos los giros son compatibles, el grafo de incompatibilidad no tendrá aristas y se pueden pintar con un solo color todos los vértices; la circulación en el cruce estaría permitida para todos los giros, lo que tampoco es una opción.*

*Situaciones intermedias se obtienen cuando el grafo se hace más restrictivo, es decir, limitando cada periodo de luz verde en un semáforo a los sentidos de circulación de menos riesgo, e.g., al realizar los trayectos AD y CD, véase la figura 6.1, hay mayor riesgo de colisiones que cuando se recorren CB y DB.*

El permitir que las variables sean palabras u oraciones del lenguaje natural o artificial (linguistic variables), es otra de las formas posibles en que se puede representar la vaguedad, este concepto se desarrolló como un complemento del concepto de variable numérica [41]. La teoría de variables lingüísticas fue desarrollada por L.A. Zadeh en [73]. Esta forma de imprecisión gradúa la pertenencia de una arista determinada con calificativos tales como *alta, moderadamente alta,...*, etc.

*Volviendo al ejemplo 6.1, dependiendo del riesgo que implique el que dos sentidos de circulación se realicen al mismo tiempo, las incompatibilidades se clasifican según su grado en nula, baja, media y alta, que en la figura 6.1, corresponden a los sentidos de circulación CB-DB, AB-CB, AD-CD y AB-CD respectivamente.*

*De lo anterior, el problema puede modelarse como un grafo con aristas difusas  $\tilde{G} = (V, \mu)$  donde  $\mu : V \times V \rightarrow \{\text{nulo, bajo, medio, alto}\}$  y  $\mu(i, j)$  indica el grado en que el*



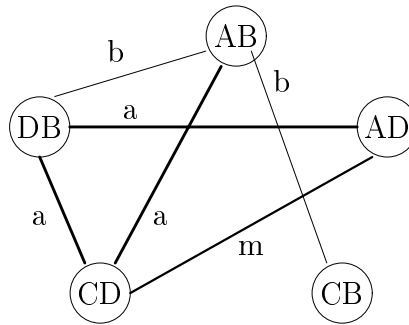


Figura 6.3: Grafo difuso del ejemplo 6.1

*sentido de circulación  $i$  es incompatible con el sentido de circulación  $j$ .*

*Este grafo difuso se ilustra en la figura 6.3, en el que se han señalado las aristas con las letras  $a$ ,  $m$  y  $b$  para indicar el grado de incompatibilidad alto, medio y bajo respectivamente.*

*La cuestión es si se puede plantear el problema de coloración del ejemplo cuando el grafo es difuso, así como interpretar los resultados obtenidos.*

Considerando una variante del esquema general para resolver problemas de optimización en un ambiente difuso de Delgado et al. [21], que es usando la representación  $\alpha$  - cortes para obtener un conjunto de problemas nítidos que dependen del grado de pertenencia; al resolver estos problemas se obtienen soluciones que dependen de dicho grado y se llaman *soluciones difusas* del problema.

De las definiciones 6.7 y 6.1, se puede concluir que existe una secuencia  $\mathcal{A} = (\alpha_1, \dots, \alpha_k)$ , donde  $\alpha_i \in \mathbb{R}$  ó  $\alpha_i$  es elemento de un conjunto de adjetivos identificando cada  $\alpha_i$  un intervalo  $I_i = (\alpha_{i-1}, \alpha_i)$  dentro del cual el  $\alpha$  - corte,  $G_{\alpha_i}$ , no cambia. De esta manera, con los distintos valores de  $\alpha_i$  se genera una secuencia  $\mathcal{I}$  de intervalos dentro de cada uno de los cuales se define un grafo nítido.

La solución al problema de optimización se puede expresar de la siguiente manera: Existe una secuencia  $\mathcal{A} = (\alpha'_1, \dots, \alpha'_p)$  de niveles de pertenencia y el conjunto de soluciones

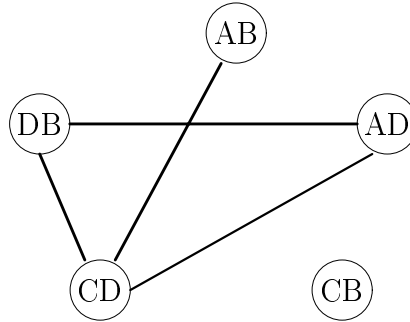


Figura 6.4:  $G_{media}$  del ejemplo 6.1

$\mathcal{S} = \{s_1, \dots, s_p\}$  tales que  $s_1$  es solución de  $G_{\alpha_1}$  y es óptima en  $J_1 = (0, \alpha'_1]$ ,  $s_2$  es solución de  $G_{\alpha_2}$  y es óptima en  $J_2 = (\alpha'_1, \alpha'_2]$ ,  $\dots$ ,  $s_p$  es solución de  $G_{\alpha_p}$  y es óptima en  $J_p = (\alpha'_{p-1}, \alpha'_p]$ .

Se hace notar que los intervalos  $J_l$  generalmente no coinciden con los intervalos  $I_k \in \mathcal{I}$  definidos por los  $\alpha$  – cortes y que obviamente,  $\alpha'_p = \alpha_k$ .

Una solución difusa se puede considerar como el conjunto difuso  $\tilde{\mathcal{S}} = \{(s_1, \alpha'_1), \dots, (s_p, \alpha'_p)\}$ .

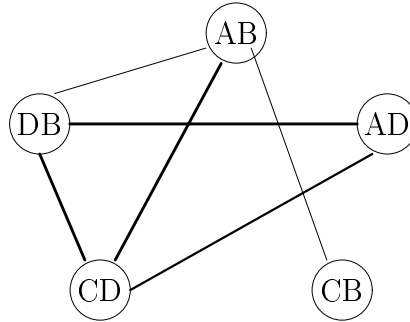
Con esta definición cada  $s_i$  de los elementos de  $\mathcal{S}$  será una solución nítida que es óptima para niveles de pertenencia mayores o iguales a  $\alpha'_i$  con  $i \in \{1, \dots, p\}$ .

*Continuando con el ejemplo 6.1, si además de las aristas que son incompatibles se consideran:*

1. *El grafo de incompatibilidad alta  $G_{alta}$  es el grafo nítido representado en la figura 6.2.*
2. *Al añadir al grafo anterior las aristas que tienen incompatibilidad media, se define el grafo  $G_{media}$ , que se representa en la figura 6.4:*

*Este grafo tiene número cromático 3; una 3 – coloración es la siguiente*

$$C(AB) = 1 \text{ ó } 3, \quad C(AD) = 3, \quad C(CB) = 1, 2 \text{ ó } 3, \quad C(CD) = 2 \text{ y } C(DB) = 1$$

Figura 6.5:  $G_{baja}$  del ejemplo 6.1

se observa que con esta coloración se incrementa a tres el número de periodos de color verde en el ciclo del semáforo, esta situación no exige tanta pericia de los conductores como en el grafo  $G$  que parece en el ejemplo 6.1, al evitar aquí que puedan efectuarse simultáneamente los sentidos de circulación  $AD$  y  $CD$  además de los no permitidos en el grafo  $G$ .

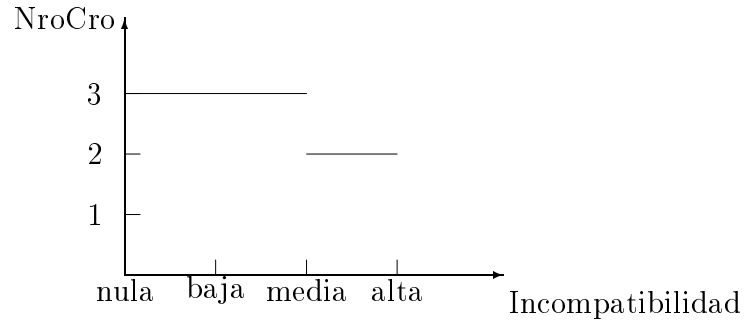
3. Al añadir al grafo  $G_{media}$  las aristas que tengan incompatibilidad baja, es decir, las aristas  $\{AB, CB\}$  y  $\{AB, DB\}$ , se define el grafo  $G_{baja}$ , representado en la figura 6.5.

$G_{baja}$  tiene número cromático 3 y una 3 – coloración es la siguiente

$$C(AB) = 1, C(AD) = 1, C(CB) = 2 \text{ ó } 3, C(CD) = 2 \text{ y } C(DB) = 3$$

en este caso, el número de periodos de color verde en el ciclo del semáforo sigue siendo tres y corresponde a los sentidos de circulación  $AB$  y  $AD$ ,  $CB$  y  $CD$  finalmente  $CB, DB$ ; el tiempo que dure en cada uno dependerá, entre otros factores, del volumen de tráfico en cada sentido de circulación.

La ventaja de los casos  $G_{media}$  y  $G_{baja}$  sobre  $G$ , al que se designará por  $G_{alta}$ , es que disminuyen el riesgo de accidentes, pero también tienen el inconveniente de hacer más lento el tráfico.

Figura 6.6:  $\chi(\tilde{G})$  del ejemplo 6.1

La solución difusa es  $\tilde{\mathcal{S}} = \{(3, \text{baja}), (2, \text{alta})\}$  y para cada elemento proporciona el número cromático que se obtiene con el  $\alpha$  correspondiente y ese valor de  $\alpha$  es el mayor con el que se puede obtener dicho número cromático. En otras palabras, cada elemento proporciona el número de periodos de luz verde en el semáforo y el mayor nivel de riesgo de colisiones que se corre con dichos periodos; a mayor incompatibilidad le corresponde mayor riesgo. La solución se puede representar en la figura 6.6.

Notar que en la solución anterior los intervalos  $J_i$  de la solución no coinciden, como ya se dijo, con los intervalos  $I_k \in \mathcal{I}$ . Concretamente,  $m = 3$  y  $p = 2$ .

A partir del esquema general de optimización y de estas ilustraciones se presenta la siguiente definición.

**Definición 6.8** Dado un grafo con aristas difusas  $\tilde{G} = (V, \mu)$  el número cromático difuso de  $\tilde{G}$  estará formado por el conjunto difuso no normalizado cuyos elementos son las parejas formadas por el número cromático del  $\alpha_i$  – corte  $\tilde{G}_{\alpha_i}$  y  $\alpha_i$ , con  $i$  que toma valores en  $\{1, \dots, p\}$ .

No es difícil demostrar que el número cromático difuso es un conjunto convexo no-normalizado cuyo valor modal está asociado con el grafo más restrictivo.

El problema que se plantea ahora es el de la determinación del número cromático de un grafo difuso. Según lo expuesto anteriormente, habrá que identificar los distintos grafos que resultan de variar el nivel  $\alpha \in \{\alpha_1, \dots, \alpha_m\}$ , determinar el número cromático de cada uno de estos grafos (nítidos)  $G_{\alpha_m}$  resultantes y componer la función que asocia a cada nivel  $\alpha$  el valor  $\chi(G_{\alpha_m})$ .

Se pueden excluir de estos cálculos los valores extremos  $\alpha = 0$ , que asignaría un color diferente a cada vértice (el número cromático sería igual a  $n$ , el número de vértices del grafo), y el valor  $\alpha = 1$  que permitiría colorear todos los vértices con un único color, por lo que el número cromático sería igual a 1.

Un algoritmo exacto que puede resolver el problema de coloración mínima para grafos pequeños es el de Korman [43]. Para tamaños mayores habrá que utilizar heurísticas.

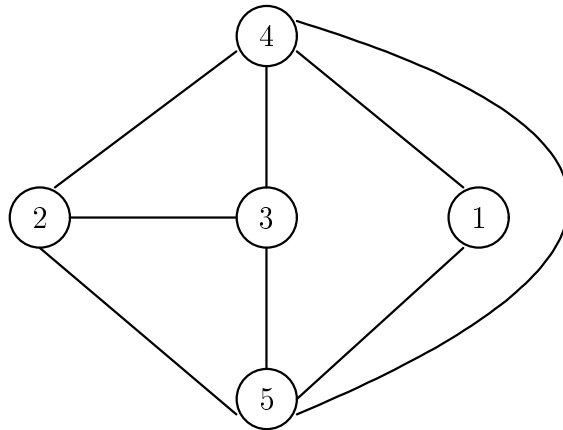
*Si en el ejemplo de conglomerados 2.2 se considera un  $\alpha < 0.01$  entonces el grafo correspondiente es un grafo completo y  $\chi_\alpha(G) = 5$  lo que significa que todos los puntos están suficientemente retirados entre sí. El conglomerado tendrá  $|V|$  elementos.*

*Si  $\alpha \geq 0.07$  entonces el grafo no tiene aristas y  $\chi_\alpha(G) = 1$ , lo que quiere decir que todos los puntos están cerca y el conglomerado tiene solo un elemento que es igual a  $V$ . Este caso, como el anterior, carecen de interés.*

*Los valores de  $\alpha$  mayores o iguales a 0.01 y menores que 0.07 darán lugar a diferentes conglomerados o formas en que se pueden agrupar los elementos de acuerdo a la distancia que los separa. Las distancias pueden considerarse como el nivel de pertenencia de las aristas y el grafo puede verse como un grafo con aristas difusas.*

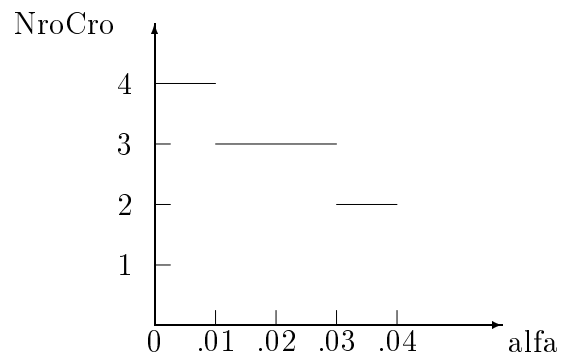
*En este grafo se obtienen los siguientes  $\alpha$  – cortes con sus respectivas coloraciones.*

- *Si  $\alpha = 0.01$ , es decir, si se considera que dos elementos están cerca solo si su distancia es menor o igual a 0.01, entonces se tendrá un grafo completo menos la arista  $\{1, 2\}$  con  $\chi_\alpha(G) = 4$ . Notar que el grafo será el mismo si  $\alpha$  es cualquier valor del intervalo  $[0.01, 0.02) = I_1$ .*

Figura 6.7:  $G(.02,.03]$ 

- Para  $\alpha = 0.02$ , hay que retirar también la arista  $\{1, 3\}$ . El correspondiente grafo se representa en la figura 6.7. con  $\chi_\alpha(G) = 4$ , como antes, el grafo no cambia si  $\alpha \in [0.02, 0.03) = I_2$ .
- Cuando  $\alpha \in [0.03, 0.04) = I_3$ , entonces al grafo anterior se le retiran las aristas  $\{2, 4\}$  y  $\{4, 5\}$  y queda el grafo en el que  $\chi_\alpha(G) = 3$ .
- Si  $\alpha \in [0.04, 0.05) = I_4$  al grafo tiene las aristas  $\{1, 4\}$ ,  $\{3, 4\}$  y  $\{3, 5\}$  y  $\chi_\alpha(G) = 2$ .
- Para  $\alpha \in [0.05, 0.06) = I_5$  el grafo tiene las aristas  $\{3, 4\}$  y  $\{3, 5\}$  y también  $\chi_\alpha(G) = 2$ .
- Si  $\alpha \in [0.06, 0.07) = I_6$  el grafo tiene la arista  $\{3, 5\}$  y  $\chi_\alpha(G) = 2$ .

Se han identificado la secuencia de intervalos  $\mathcal{I} = (I_1, \dots, I_6)$ , la secuencia de intervalos  $\mathcal{J} = (J_1, J_2, J_3)$  donde  $J_1 = [0.02, 0.03)$ ,  $J_2 = [0.03, 0.04)$ ,  $J_3 = [0.04, 0.07)$  y el número cromático difuso es  $\mathcal{S} = \{(4, 0.01), (3, 0.03), (2, 0.04)\}$ , en donde también puede verse que los intervalos de  $\mathcal{J}$  no coinciden con los intervalos de  $\mathcal{I}$ , que se representa en la figura 6.8.

Figura 6.8:  $\chi(\tilde{G})$  del ejemplo 2.2

*En las coloraciones de los dos ejemplos anteriores, el valor modal de cada coloración está asociado al grafo de mayor sensibilidad o más restrictivo, para cada uno de los  $\alpha$ -cortes,  $G_{\alpha_i}$ ,  $i = 1, \dots, p$ .*

# Bibliografía

- [1] BEASLEY, J.E., AND CHU, P.C. (1996). A genetic algorithm for the set covering problem. *European Journal of Operational Research* **94**, 392-404.
- [2] BERGE, C. (1973). Graphes et hypergraphs. Dunod, Paris.
- [3] BRON, C., AND KERBOSCH, J. (1973). Finding All Cliques of an Undirected Graph [H]. *Communications of the ACM, Algorithm 457* **16**, **9**, 575-577
- [4] BURKE, E., AND ROSS, P., EDS. (1996). Practice and Theory of Automated Timetabling, Lecture Notes in Computer Science 1153. *Springer-Verlag*.
- [5] BURKE, E. JACKSON, K., AND KINGSTON, J.H. WEARE, R. (1997). Automated University Timetabling The State of the Art. *The Computer Journal* **40**, **9**, 565-571.
- [6] BURKE, E., AND ROSS, P., EDS. (1998). Practice and Theory of Automated Timetabling II, Lecture Notes in Computer Science 1408. *Springer-Verlag*.
- [7] BUTTLER, C., AND MATTHEWS, L.R. (1982). An application of graph colouring on the railways. **En: Wilson, R. J. (editor). Applications of combinatorics. Shiva Publishing limited, England, 19-28.**
- [8] ČANGALOVIĆ, M., AND SCHREUDER, J.A.M. (1991). Exact Algorithm for Weighted Graphs Approach. *European Journal of Operational Research*. **112**, 517-530



- [9] ČANGALOVIĆ, M., AND SCHREUDER, J.A.M. (1992). Modelling and Solving an Acyclic Multiperiod Timetabling Problem. *Discrete Applied Mathematics*. **35, 3**, 177-195.
- [10] CANO ALSUA, S. (1999). Problema de asignación cuadrática. *Trabajo de Investigación; Programa de Doctorado, Facultad de Matemáticas, Universidad Complutense de Madrid*.
- [11] CARTER, M.W. (1986). A survey of practical applications of examination timetabling algorithms. *Operations Research*. **34**, 193-202.
- [12] CARTER, M.W. LAPORTE, G., AND CHINNECK, J.W. (1994). A General Examination Scheduling System. *Interfaces* **24**, 109-120.
- [13] CARTER, M.W., AND LAPORTE, G. (1996). Recent Developments in Practical Examination Timetabling. **En: Burke, E. Ross, P., eds, Practice and Theory of Automated Timetabling, Springer-Verlag Lecture Notes in Computer Science 1153**, 3-21.
- [14] CARTER, M.W., AND LAPORTE, G. (1998). Recent Developments in Practical Course Timetabling. **En: Burke, E. Ross, P., eds, Practice and Theory of Automated Timetabling, Springer-Verlag Lecture Notes in Computer Science 1408**, 3-19.
- [15] CHAMS, M., HERTZ, A., AND DE WERRA, D. (1987). Some experiments with simulated annealing for coloring graphs. *European Journal of Operational Research* **32**, 260-266.
- [16] CHARTRAND, G., GELLER, D.P., AND HEDETNIEMI, S. (1971). Graphs with Forbidden Subgraphs. *Journal of Combinatorial Theory (B)* **10**, 12-41.

- [17] COSTA, D. HERTZ, A. (1997). Ants can colour graphs. *Journal of the Operational Research Society* **48**, 295-305.
- [18] CRAINE, W.L. (1994). Characterizations of fuzzy interval graphs. *Fuzzy sets and systems* **68**, 181-193.
- [19] CULBERSON, J.C., AND LUO, F. (1996). Exploring the k-coloreable Landscape with Iterated Greedy. **En: Johnson, D.S., and Trick, M. A., editors, DIMACS. Series in Discret And Theoretical Computer Sciences, 26**, 245-281.
- [20] DAVIS, L. (EDITOR). (1991). Handbook of genetic algorithms. *Van Nostrand Reinhold*.
- [21] DELGADO, M. VERDEGAY, J.L., AND VILA, M.A. (1990). On Valuation and Optimization Problems in Fuzzy Graphs: A General Approach and Some Particular Cases. *ORSA Journal on Computing* **2, 1**, 74-83.
- [22] DE LUCA CARDILLO, D., AND MIONE, N. (1998). k L-list  $\tau$  colouring of graphs. *European Journal of Operational Research*, **106**, 160-164.
- [23] DUBOIS, D. PRADE, H. (1985). Possibility Theory. An Approach to Computerized Processing of Uncertainty. *Plenum Press, New York*, 13-36.
- [24] FLEURENT, C., AND FERLAND, J.A. (1996). Genetic and Hybrid Algorithms for Graph Coloring. *Annals of Operational Research*, **63**, 437-461.
- [25] FRITSCH, R., AND FRITSCH, G. (1998). The Four-Color Theorem. *Springer-Verlag, New York*, 1-41.
- [26] GAREY, M.R. AND JOHNSON, D.S. (1979). Computers and Intractability: A Guide to the Theory of NP-Completeness. *W.H. Freeman and Company, New York*.
- [27] GLOVER, F. (1994). Genetic Algorithms and Scatter Search: Unsuspected Potentials. *Statistics and Computing*, **4**, 131-140.

- [28] GOLDBERG, D.E. (1989). Genetic Algorithms in Search, Optimization and Machine Learning. *Reading, MA: Addison Wesley*.
- [29] GOLUMBIC, M.C. (1980). Algorithmic Graph Theory and Perfect Graphs. *Academic, New York*.
- [30] HANSEN, P., AND DELATTRE, M. (1978). Complete-link cluster analysis by graph coloring. *Journal of the American Statistical Association*, **73**, 397-403.
- [31] HERTZ, A., AND DE WERRA, D. (1987). Using tabu search techniques for graph coloring. *Computing*, **39**, 345-351.
- [32] HIFI, M. (1997). A genetic algorithm-based heuristic for solving the weighted maximum independent set and some equivalent problems. *Journal of the Operational Research Society* **48**, 612-622.
- [33] HOLLAND, J. H. (1975). Adaptation in Natural and Artificial Systems. *University of Michigan Press: Ann Arbor, MI*.
- [34] HONG, T-P., WANG, H-S., AND CHEN, W-C. (2000). Simultaneously Applying Multiple Mutation Operators in Genetic Algorithms. *Journal of Heuristics* **6**, 439-455.
- [35] IRVING, R.W. (1983). NP-Completeness of a Family of Graph-Coloring Problems. *Discrete Applied Mathematics* **5**, 111-117.
- [36] JOHNSON, D.S., ARAGON, C.R., MCGEOCH, L.A., AND SCHEVON, C. (1991). Optimization by Simulated Annealing: an Experimental Evaluation; Part II, Graph Coloring and Number Partitioning. *Operations Research* **39**, 378-406.
- [37] JOHNSON, D.S., AND TRICK, M.A. (1997). Cliques Coloring and Satisfiability. *DIMACS*.

- [38] KAUFMANN, A. (1973). Introduction à la théorie des sous-ensembles flous, 1. Éléments théoriques de base. *Masson et Cie, Paris, France*.
- [39] KERRE, E.E. (1993). Basic Principles of Fuzzy Set Theory for the Representation and Manipulation of Imprecision and Uncertainty. **En: Kerre, E.E. Editor. Introduction to the Basic Principles of Fuzzy Set Theory and Some of its Applications, Second Revised Edition Communication and Cognition, Gent, Belgium, 57-70.**
- [40] KIAER, L., AND YELLEN, J. (1992). Weighted Graphs and University Course Timetabling. *Computers and Operations Research* **19**, **1**, 59-67
- [41] KICKERT, W.J.M. (1978). Fuzzy theories on decision-making. A critical review. *Martinus Nijhoff Social Sciences, Leiden*
- [42] KÓCZY, L. (1992). Fuzzy graphs in the evaluation and optimization of networks. *Fuzzy sets and systems* **46**, 307-319.
- [43] KORMAN, S.M. (1979). The Graph-Colouring Problem. **En: Christofides, N. Mingozzi, A. Toth, P., and Sandi, C. Editors. Combinatorial Optimization. John Wiley and Sons. 211-235.**
- [44] KUČERA, L. (1991). The greedy coloring is a bad probabilistic algorithm. *Journal of Algorithms* **12**, 674-684.
- [45] LAGUNA, M., AND MARTÍ, R. (2000). A GRASP for Coloring Sparse Graphs. <http://bus.colorado.edu/faculty/laguna.html>.
- [46] LAMPORT, L. (1994). Latex. User's Guide and Reference Manual. *Addison-Wesley Publishing Company, Reading, Mass*, 117-132.

- [47] LÓPEZ BRACHO, R. GUTIÉRREZ ANDRADE, M.A. ET RAMÍREZ RODRÍGUEZ, J. (1996). Une condition suffisante pour l' integralité du problème de partage. 14<sup>th</sup> *Triennial Conference IFORS, Vancouver Canadá* , 8-12 julio.
- [48] MARDIA, K.V. KENT, J.T., AND BIBBY, J.M. (1979). *Multivariate Analysis. Academic Press, London.*
- [49] NEMHAUSER, G.L., AND WOLSEY, L.A. (1988). *Integer and Combinatorial Optimization. John Wiley and Sons, New York.*
- [50] ODIJK, M. (1999). Sensitivity analysis of a railway station track layout with respect to a given timetable. *European Journal of Operational Research* **112**, 517-530.
- [51] OPSUT, R.J., AND ROBERTS, F.S (1983). I-Colorings, I-Phasing, and I-Intersection Assignments for Graphs, and Their Applications. *Networks* **13**, 327-345.
- [52] PARDALOS, P.M., AND WOLKOWICS, H. (1994). *Quadratic Assignment and Related Problems. DIMACS Series, American Mathematical Society.*
- [53] PARDALOS, P.M. MAVRIDOU, T., AND XUE, J. (1998). The Graph Coloring Problem: A Bibliographic Survey. **En: Du, D.-Z., and Pardalos, P.M. (Eds.). Handbook of Combinatorial Optimization (Vol. 2). Kluwer Academic Publishers, Boston, 331-395.**
- [54] PIRLOT, M. (1996). General local search methods. *European Journal of Operational Research* **92**, 493-511.
- [55] RAMÍREZ RODRÍGUEZ, J. (1997). Nuevos enfoques algorítmicos para el problema de coloración de grafos. *Trabajo de Investigación; Programa de Doctorado, Facultad de Matemáticas, Universidad Complutense de Madrid.*
- [56] RAMÍREZ RODRÍGUEZ, J. Y YÁÑEZ GESTOSO, J. (2000). Un problema de coloración robusta generalizada. Aplicación a un problema de horarios. *Memorias del*

*X Congreso Latino-Iberoamericano de Investigación de Operaciones y Sistemas, en disco compacto, México D.F., 4-8 septiembre.*

- [57] RAYCHAUDHURI, A. (1994). Further results on T-coloring and frequency assignment problems. *SIAM Journal Discret Mathematics*. **7**, **4**, 605-613.
- [58] REEVES, C.R., AND BEASLEY, J.E. (1993). Combinatorial problems. *En: Reeves, C.R., editor, Modern Heuristic Techniques for Combinatorial Problems, Blackwell Scientific Publications, Oxford*, 1-19.
- [59] ROBERTS, F.S. (1979). On the mobil radio frequency assignment problem and the traffic light phasing problem. *Annals New York Academy of Sciences* **319**, 466-483.
- [60] ROBERTS, F.S. (1991). T-Colorings of graphs: recent results and open problems. *Discrete Mathematics* **93**, 229-245.
- [61] ROSENFELD, A. (1975). Fuzzy graphs. *En: Zadeh, L.A. Fu, K.S., and Shimura, M., eds, Fuzzy sets and their applications to cognitive and decision processes, Academic Press New York*, 77-95.
- [62] STOFFERS, K.E. (1968). Scheduling of traffic lights- A new approach. *Transportation Research* **2**, 199-234.
- [63] TESMAN, B.A. (1993). List T-Colorings of graphs. *Discrete Applied Mathematics* **45**, 277-289.
- [64] VERHOEVEN, M.G.A. (1998). Tabu search for resource-constrained scheduling. *European Journal of Operational Research* **106**, 266-276.
- [65] DE WERRA, D. (1985). An Introduction to Timetabling. *European Journal of Operational Research* **19**, 151-162.

- [66] DE WERRA, D. (1996). Some combinatorial Models for Course Scheduling. *En: Burke, E. Ross, P., eds, Practice and Theory of Automated Timetabling, Springer-Verlag Lecture Notes in Computer Science 1153*, 296-308.
- [67] DE WERRA, D. (1996). Extensions of coloring models for scheduling purposes. *European Journal of Operational Research* **92**, 474-492.
- [68] WILSON, J.M. (1997). A genetic algorithm for the generalised assignment problem. *Journal of the Operational Research Society* **48**, 804-809.
- [69] XUE, J. (1994). Edge-Maximal Triangulated Subgraphs and Heuristics for the Maximum Clique Problem. *Networks* **24**, 109-120.
- [70] ZADEH, L.A. (1965). Fuzzy sets. *Information and control* **8, 3**, 338-353.
- [71] ZADEH, L.A. (1971). Similarity relations and fuzzy ordering. *Information Science* **3**, 177-200.
- [72] ZADEH, L.A. (1975). Calculus of fuzzy restrictions. *En: Zadeh, L.A. Fu, K.S., and Shimura, M., eds, Fuzzy sets and their applications to cognitive and decision processes, Academic Press New York*, 1-39.
- [73] ZADEH, L.A. (1975). The concept of linguistic variable and its application to approximate reasoning, Part I. *Information Science* **8**, 199-249, Part II, idem anterior, 301-357 and Part III, **9**, 43-80.
- [74] ZIMMERMANN, H.J. (1985). Fuzzy Set Theory and its Applications. *Kluwer-Nijhoff Publishing, Boston*, 39-81