Tesis de Licenciatura

Entre k-coloreo y coloreo con listas: μ-coloreo

Mariano Agustín Cecowski Palacio
mcecowsk@dc.uba.ar

Departamento de Computación
Facultad de Ciencias Exactas y Naturales - UBA

Directora: **Flavia Bonomo**
Co-Director: **Guillermo Durán**

Marzo de 2005

A mi Vieja,

quien nunca me falló, y a quien espero nunca fallarle.

Agradecimientos

- A Flavia, directora y compañera, por toda la paciencia, el empuje, y la paciencia.
- A Willy Durán, por darme una mano cuando la necesité, y por ser un excelente profesor, y un aún mejor ser humano.
- A la *Troupe Brullante* del departamento (Andrés, Jimena, Jaka, Pablo, Marcelo) por divertirnos, ayudarnos, y acompañarnos durante todos estos años.
- A Roberto Bevilaqua y Juan Santos. Por demostrar que para ser un buen profesor, hace falta ser primero un gran tipo.
- Al Departamento de Computación, a la Facultad de Ciencias Exactas y Naturales, y a la Universidad de Buenos Aires, por la educación que todos nos merecemos.

- A Saša, mi anti-musa que hizo lo imposible para que no terminara esta tesis, porque espero que siga haciendo lo mismo por varias décadas más.
- A mis hermanos, Samanta y Martín, por defenderse mutuamente cuando me peleaba con alguno de ellos.
- A mis viejos, por ayudarme y apoyarme en todas mis decisiones, aún si significaron estar lejos de ellos.
- Mateji, Petri, Anici, in celi moji slovenski družini, ker hvala njim se lahko počutim v tujini (skoraj) doma.
- Al resto de la *Troupe Brullante* (Alita, Victoria, Natalia, Nico, Gabi, Romina, etc., etc.) por... ustedes ya saben.

Índice

Resumen

El problema de k-coloreo en un grafo consiste en asignarle a cada vértice un 'color' o 'número' de forma tal que dos vértices adyacentes no reciban el mismo, y usando no más de k etiquetas distintas. Es un problema NP-completo en el caso general, aunque se puede resolver en tiempo polinomial restringiendo la entrada a ciertas clases de grafos, como por ejemplo los grafos perfectos. El problema de asignación de recursos indistinguibles a tareas es una clara situación que es fácilmente representable con k-coloreo. Cada vértice representa una tarea, y las tareas a las cuales no se les puede asignar el mismo recurso son adyacentes. Los colores representarán en este caso los recursos.

En el problema de coloreo con listas, cada vértice tiene asociada una lista de 'colores' o 'valores' con los cuales puede ser etiquetado. El problema consiste, entonces, en etiquetar todos los vértices del grafo de forma tal que no haya dos vértices adyacentes que utilicen la misma etiqueta y a cada uno de los vértices le sea asignada una etiqueta perteneciente a su propia lista de etiquetas. El problema de coloreo con listas es muy utilizado actualmente para la asignación de radio-frecuencias.

En esta tesis estudiamos una variante nueva del problema de coloreo en grafos, el μ-coloreo, aplicable a problemas de asignación de recursos a tareas en los cuales los recursos son ordenables de modo tal que un recurso puede ser asignado a una tarea si su número de orden es suficientemente bajo. En el problema de

μ-coloreo cada vértice v tiene asociado un número natural $\mu(v)$, que es el máximo 'color' con el que puede ser etiquetado. Eso define una función μ de los vértices del grafo en los naturales.

Notemos que el problema de k-coloreo es un caso particular de μ-coloreo, donde la función μ es constante con valor k, con lo cual μ-coloreo resulta ser NP-completo. A su vez, μ-coloreo es un caso particular del problema de coloreo con listas, donde la lista asociada al vértice v tiene la forma $\{1, \ldots, \mu(v)\}$.

En esta tesis mostramos que, en términos de complejidad, el problema de μ-coloreo se encuentra estrictamente entre el coloreo y el coloreo con listas. Es decir, existe una clase de grafos para la cual μ-coloreo es polinomial y coloreo con listas es NP-completo, y existe otra clase de grafos para la cual k-coloreo es polinomial y μ-coloreo es NP-completo.

Una clique en un grafo es un subgrafo completo maximal. Berge definió en 1960 los grafos perfectos como aquellos G tales que, para todo subgrafo inducido H de G, H es coloreable con una cantidad de colores igual al tamaño de una clique máxima de H. Esto es equivalente a decir que G es perfecto si y sólo si para todo H subgrafo inducido de G, y para todo k, las cliques de H son k-coloreables si y sólo si H es k-coloreable. En esta tesis proyectamos el concepto de grafo perfecto al μ-coloreo, definiendo los grafos M-perfectos: un grafo G es M-perfecto si y sólo si, para todo H subgrafo inducido de G, y toda función μ, las cliques de H son μ-coloreables si y sólo si H es μ-coloreable.

Obtuvimos para los grafos M-perfectos propiedades similares a las de los perfectos, como ser, una caracterización elegante por subgrafos prohibidos, un algoritmo polinomial de reconocimiento y un algoritmo polinomial para μ-coloreo en grafos M-perfectos.

Los resultados de esta tesis aparecen publicados en [3].

Capítulo 1

Introducción

En este capítulo definiremos los conceptos básicos de teoría de grafos y complejidad computacional utilizados a lo largo de la tesis. También presentaremos las propiedades principales de algunas clases de grafos relacionadas con los resultados de la tesis e introduciremos y describiremos las diferentes variantes del problema de coloreo estudiadas en la misma.

1.1 Definiciones básicas

Denotaremos un grafo G por el par $(V(G), E(G))$, donde $V(G)$ es un conjunto finito, el conjunto de *vértices* de G, y $E(G)$ es un conjunto de pares no ordenados de vértices distintos de G, llamados *aristas*. Si $(v, w) \in E(G)$ decimos que los vértices v y w son *adyacentes* en G, o que están unidos por una arista.

Dado un grafo $G = (V, E)$, defininimos su grafo *complemento* $\overline{G} = (V, \overline{E})$, tal que dados dos vértices distintos v y w en V, la arista (v, w) pertenece a $\overline{E}$ si y sólo si no pertenece a E.

Notaremos por $N_G(v)$ al *vecindario abierto* del vértice v en G, es decir, el conjunto de vértices adyacentes a v, y por $N_G[v]$ al *vecindario cerrado* del vértice v en G, es decir, $N(v) \cup \{v\}$.

Un vértice v es *universal* cuando es adyacente a todos los demás, es decir, $N_G[v] = V(G)$.

El *grado* de un vértice v es el cardinal de $N_G(v)$ y se nota $d(v)$.

Un *camino* de longitud t es una sucesión de vértices distintos $v_1, v_2, \ldots, v_t$ tales que $(v_i, v_{i+1}) \in E$ para $1 \leq i \leq t-1$. Un *camino inducido* (o 'path') es un camino sin cuerdas, es decir, no existe $p > 1$ tal que $(v_i, v_{i+p}) \in E$. Notamos como P_t al camino inducido de longitud t.

Figura 1.1: Un camino inducido de longitud t.

Un *ciclo* de longitud t es una sucesión de vértices $v_1, v_2, \ldots, v_t$ tal que $v_1, v_2, \ldots, v_{t-1}$ forman un camino y v_t es adyacente a v_1 y v_{t-1}. Un *ciclo inducido* es un ciclo sin cuerdas, es decir, no hay dos vértices adyacentes que no sean consecutivos en el ciclo. Notamos por C_t al ciclo inducido de longitud t. Un *hole* es un ciclo inducido de longitud mayor o igual que 4.

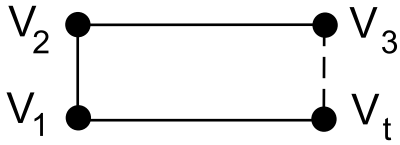

Figura 1.2: Un ciclo inducido de longitud t.

Un grafo H es *subgrafo* de G si y sólo si $V(H) \subseteq V(G)$ y $E(H) \subseteq E(G)$.

Un grafo H es *subgrafo inducido* de G si y sólo si $V(H) \subseteq V(G)$ y además, $(v, w) \in E(H) \Leftrightarrow v, w \in V(H)$ y $(v, w) \in E(G)$.

Una propiedad se dice *hereditaria* si al valer para un grafo, vale también para todos sus subgrafos inducidos.

Un grafo G es un *cografo* si no tiene como subgrafo inducido a P_4. En la Sección 1.3.2 presentaremos las principales propiedades de esta clase de grafos.

Un grafo es *completo* si sus vértices son adyacentes dos a dos. Denotaremos por K_n al grafo completo de n vértices.

Un *completo* H de G es un subconjunto de vértices adyacentes dos a dos.

Una *clique* es un completo maximal de G. Es decir, es un completo de G que no está propiamente incluido en otro completo de G. Llamaremos $\omega(G)$ al tamaño de una clique máxima de G y $\mathcal{C}(G)$ al conjunto de cliques de G.

Un *conjunto independiente* de G es un completo de $\overline{G}$. Llamaremos $\alpha(G)$ al tamaño de un conjunto independiente máximo de G.

Sea G un grafo, U y W subconjuntos disjuntos de $V(G)$. Se dice que U es *completo a* W en G si todos los vértices de U son adyacentes a todos los vértices de W en G. Se dice que U es *anticompleto a* W en G si todos los vértices de U son no adyacentes a todos los vértices de W en G.

Un *k-coloreo* de G es una partición de $V(G)$ en k conjuntos independientes (a cada uno de ellos se le asocia un color, de ahí el nombre "coloreo"). Si G admite un k-coloreo decimos que es *k-coloreable.*

El *número cromático* de G es el menor k para el cual es k-coloreable, y lo denotamos por $\chi(G)$.

Un cubrimiento de vértices por cliques en un grafo G es un subconjunto de las cliques de G tal que todo vértice de G pertenece a al menos una de ellas. Se define $k(G)$ como el tamaño de un cubrimiento mínimo de vértices por cliques en G. Es fácil ver que $k(G) = \chi(\overline{G})$.

Un grafo G es *perfecto* si y sólo si $\chi(H) = \omega(H)$ para todo subgrafo inducido H de G. En la Sección 1.3.1 presentaremos aspectos históricos y las propiedades principales de los grafos perfectos.

Dada una familia finita de conjuntos no vacíos, el *grafo intersección* de esta familia se obtiene representando cada conjunto por un vértice y conectando dos vértices por un arista si y sólo si los correspondientes conjuntos se intersecan.

Un *grafo de intervalos* es el grafo de intersección de intervalos en una recta. En la Sección 1.3.3 mencionaremos algunas propiedades de esta clase de grafos.

Sea $\mathcal{F}$ una familia de grafos. Un grafo G es *$\mathcal{F}$-free* si no contiene como subgrafo inducido a ningún grafo de $\mathcal{F}$. Cuando la familia tiene un solo elemento ($\mathcal{F} = \{H\}$), un grafo $\mathcal{F}$-free se suele notar *H-free*.

Un grafo G es *cordal* si todo ciclo de longitud al menos 4 tiene al menos una cuerda. Son equivalentes a los grafos $\{C_5, hole\}$-free.

Un grafo $G = (V(G), E(G))$ es *de comparabilidad* si es posible direccionar sus aristas de modo de que el grafo dirigido resultante $G' = (V(G), D(G))$ satisfaga: $(u, v) \in D(G), (v, w) \in D(G) \Rightarrow (u, w) \in D(G)$. Un grafo es *de co-comparabilidad* si su complemento es de comparabilidad.

1.2 Complejidad algorítmica

Un problema algorítmico $\pi(I, Q)$ consiste de un conjunto I de todas las posibles entradas para el problema, llamado el conjunto de instancias, y de una pregunta Q sobre esas instancias. Resolver uno de estos problemas consiste en desarrollar un algoritmo cuya entrada es una instancia del problema y cuya salida es una respuesta a la pregunta del problema.

Llamamos a un problema *de decisión* cuando las posibles respuestas a la pregunta son SI y NO.

Un problema es *polinomial* de orden t si existe un algoritmo que lo resuelva en una cantidad de operaciones acotada por un polinomio en n de grado t, donde n es el tamaño de la entrada. Se denota como $O(n^t)$. Cuando $t = 1$ el problema se dice *lineal*. Un problema pertenece a la clase **P** si es polinomial.

Un problema de decisión pertenece a la clase **NP** (no-determinísticamente polinomial) si es resoluble en tiempo polinomial por una máquina de Turing no-determinística. Equivalentemente, un problema de decisión está en NP si cualquier instancia que produce respuesta SI posee una comprobación de correctitud (también llamada certificado) verificable en tiempo polinomial en el tamaño de la entrada.

Claramente, todo problema que pertenece a P, pertenece a NP. En otras palabras, la clase P es una sub-clase de NP. Queda aún abierta la pregunta: ¿Es NP = P?

Sean $\pi_1(I_1, Q_1)$ y $\pi_2(I_2, Q_2)$ dos problemas de decisión. Una *reducción polinomial* de π_1 en π_2 es una función $f : I_1 \to I_2$ tal que se satisfacen las siguientes dos condiciones:

1. f puede computarse en tiempo polinomial.

2. Para toda instancia $D \in I_1$, D produce respuesta SI para π_1 si y sólo si $f(D)$ produce respuesta SI para π_2.

Un problema es *NP-hard* si cualquier problema de la clase NP se puede reducir polinomialmente a él.

Un problema es *NP-completo* si pertenece a NP y es NP-hard.

1.3 Clases de grafos

1.3.1 Grafos perfectos

Mucho se ha trabajado y escrito sobre los grafos perfectos desde que fueran definidos por Berge en 1960 [1], y en años recientes se han logrado importantes avances, como las demostraciones de algunas conjeturas. Gracias a estas demostraciones tenemos varias definiciones equivalentes para estos grafos:

Un grafo G es *perfecto* si y sólo si:

- $\chi(H) = \omega(H)$ para todo subgrafo inducido H de G.
- $\alpha(H) = k(H)$ para todo subgrafo inducido H de G [23].
- $\omega(H).\alpha(H) \geq |H|$ para todo subgrafo inducido H de G [23].
- es un grafo de *Berge*, es decir, no contiene como subgrafo inducido a C_{2k+1} ni a $\overline{C_{2k+1}}$, con $k > 1$ [7].

La equivalencia entre las definiciones dadas en los dos primeros items se conoce como *Teorema de los Grafos Perfectos*, y se puede reformular como “un grafo es perfecto si y sólo si su complemento lo es". Fue conjeturado por Berge en 1960 y demostrado por Lóvasz en 1972.

La caracterización por subgrafos prohibidos dada en el último item se conoce como *Teorema Fuerte de los Grafos Perfectos.* Fue conjeturado por Berge en 1960 (es más fuerte que el resultado anterior) y demostrado por Chudnovsky, Robertson, Seymour y Thomas recién en 2002.

Los grafos perfectos también tienen importancia desde el punto de vista algorítmico, ya que en ellos los problemas de reconocimiento, coloreo, clique máxima, conjunto independiente máximo y cubrimiento mínimo de vértices por cliques son resolubles polinomialmente [6, 14].

El estudio de estos grafos favoreció la proliferación de diversas clases de grafos definidas por subgrafos prohibidos, o X-free, donde X es un grafo, o un conjunto (finito o infinito) de grafos prohibidos.

También se han definido variantes de los grafos perfectos, basadas en la igualdad de otros parámetros de un grafo, como por ejemplo los grafos *clique-perfectos* [15], basados en el conjunto independiente de cliques máximo y el cubrimiento de cliques por vértices mínimo y los grafos de *vecindad perfecta* [22], basados en el conjunto de aristas máximo sin vecinos en común y el cubrimiento mínimo de aristas por vecindades. Algunas otras variantes se basan en parámetros asociados a variantes del problema de coloreo, como los grafos *b-perfectos* [20], basados en el número b-cromático en lugar del número cromático, y los grafos *perfectos circulares* [25], basados en el coloreo circular y las cliques circulares. En esta tesis definiremos también una variante de grafos perfectos: los grafos *M-perfectos.*

1.3.2 Cografos

Recordemos que un grafo G es un *cografo* si no contiene como subgrafo inducido a P_4.

Propiedades [5]: Sea G un cografo.

- Todo subgrafo inducido de G es un cografo.
- El complemento de G es un cografo.
- G no contiene como subgrafo inducido ningún ciclo C_t de longitud $t > 4$, y por lo tanto los cografos son perfectos.
- Si G es conexo, entonces su complemento no lo es.
- Si G es conexo, entonces tiene diámetro menor o igual a 2.
- En cada subgrafo inducido H de G, la intersección de cualquier clique máxima con cualquier conjunto independiente máximo contiene solamente un vértice [9].
- G tiene al menos un par de vértices v, w tales que o bien $N[v] = N[w]$ (si son adyacentes) o $N(v) = N(w)$ (si no lo son).
- Se puede determinar si un grafo es cografo en orden $O(n)$ [9].
- Cualquier cografo puede ser construido a partir de los siguientes axiomas:
  - A_1: El grafo trivial K_1 es un cografo.
  - A_2: Si H y F son cografos con $V(H) \cap V(F) = \emptyset$, entonces $H \cup F$, también lo es, en donde $V(H \cup F) = V(H) \cup V(F)$ y $E(H \cup F) = E(H) \cup E(F)$.
  - A_3: El complemento de un cografo es un cografo.

- Equivalentemente, un grafo G es un cografo si y sólo si admite un árbol de descomposición a partir de los axiomas anteriores: cada nodo del árbol contiene un grafo; la raíz contiene el grafo original; cada nodo que contiene un grafo no trivial tiene como hijos nodos conteniendo las componentes conexas del grafo o su complemento (en un cografo no trivial alguno es disconexo); los nodos que contienen grafos triviales son hojas del árbol de descomposición.

1.3.3 Grafos de intervalos

Recordemos que un grafo $G = (V, E)$ es de *intervalos* si existe un conjunto de intervalos en la recta real cuyo grafo intersección es G. A ese conjunto se lo llama *modelo de intervalos* de G.

Propiedades [5]:

- Un grafo G es un grafo de intervalos si y sólo si es un grafo de co-comparabilidad sin ningún C_4 inducido.
- Un grafo G es un grafo de intervalos si y sólo si es a la vez un grafo de co-comparabilidad y cordal.
- Los grafos de intervalos no pueden contener como subgrafo inducido a un ciclo de longitud mayor o igual a cuatro.
- Se puede determinar si un grafo es de intervalos en orden $O(n)$ [4].
- Los grafos de intervalos son perfectos.

1.3.4 Grafos bipartitos

Un grafo $G = (V, E)$ es *bipartito* si y sólo si sus vértices pueden ser particionados en dos conjuntos independientes.

Propiedades [5]:

- Un grafo G es bipartito si y sólo no contiene ningún ciclo C_t, $t \geq 3$, de longitud impar.
- Un grafo G es bipartito si y sólo si puede ser coloreado con 2 o menos colores.
- El problema de determinar si un grafo es bipartito es lineal.
- Los grafos bipartitos son perfectos.

1.4 Problemas de coloreo de grafos

Los problemas de coloreo de grafos nacieron, al igual que otros problemas de la Teoría de Grafos, como un juego de ingenio. Entre ellos el camino Hamiltoniano (dibujar un grafo sin levantar el lapiz), colorear los grafos con 4 colores, y tantos otros. Pero en la práctica sirven para modelar diversas situaciones de la vida real, de modo que estas puedan luego ser resueltas aplicando algoritmos generales.

Distintas variantes del problema de coloreo aparecen en la literatura. Un resumen bastante completo de ellas puede encontrarse en [24].

En esta tesis pondremos atención a tres de sus variantes: la variante clásica, conocida como *k-coloreo*, una generalización conocida como *coloreo con listas* (o *list-coloring*) y una tercer variante intermedia que surge a partir del problema de asignación de aulas a materias y definiremos como *μ-coloreo*.

1.4.1 k-coloreo

El problema de k-coloreo consiste en asignarle a cada vértice un 'color' o 'número' de forma tal que dos vértices adyacentes no reciban el mismo, y usando no más de k etiquetas distintas.

k-COLOREO
INSTANCIA: Un grafo $G = (V, E)$, un entero positivo $k \leq |V|$.
PREGUNTA: ¿ Existe una función $f : V \to \mathbb{N}$ tal que $f(v) \neq f(w)$ si $(v, w) \in E$ y $f(v) \leq k$ para todo $v \in V$?

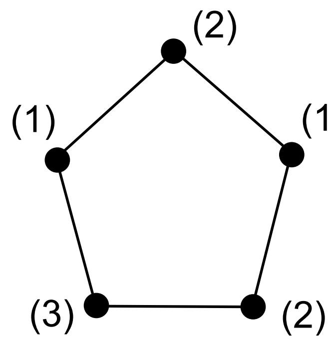

Figura 1.3: C_5 es 3-coloreable, pero no 2-coloreable.

Es un problema NP-completo en el caso general [19], y sigue siendo NP-completo aún restringiendo la entrada a la clase de grafos planares [12], circulares o arco-circulares [11]. Pero se puede resolver en tiempo polinomial restringiendo la entrada a ciertas clases de grafos, como por ejemplo grafos perfectos [14] o grafos sin vértices de grado mayor o igual a 4 [10].

Para el caso general, o las clases en las cuales el problema es NP-completo, suelen utilizarse heurísticas o algoritmos aproximados. En general se trata de algoritmos polinomiales que generan un coloreo válido pero no siempre utilizan la cantidad óptima de colores, aunque habitualmente generan soluciones suficientemente buenas como para ser aplicadas en problemas prácticos. Los algoritmos aproximados garantizan una cierta proximidad al óptimo, por ejemplo “la cantidad de colores utilizados en la solución nunca es más del doble que la óptima”.

El problema de asignación de recursos a tareas es una clara situación que es fácilmente representable con k-coloreo. Cada vértice representa una tarea, y las tareas a las cuales no se les puede asignar el mismo recurso son adyacentes. Los colores representarán en este caso los recursos.

Dentro de este problema general, en algunos casos existen características que restringen la entrada a cierta clase de grafos. Por ejemplo, en el problema de “asignación de retroproyectores a asignaturas” sabemos que el grafo va a ser siempre de intervalos (intersección de intervalos en el tiempo).

El problema de k-coloreo para grafos de intervalos es lineal, y el algoritmo que lo resuelve es un algoritmo goloso muy sencillo: dado un modelo de intervalos del grafo, que se puede obtener en tiempo lineal a partir de la representación por vértices y aristas, basta con recorrer los intervalos en una dirección (ordenados por su extremo izquierdo o derecho). Cuando encontramos el comienzo de un segmento lo pintamos con el menor color posible, cuando nos topamos con el fin de un segmento, liberamos este color. Si se utilizan más de k etiquetas es porque no existe un k-coloreo posible para el grafo dado.

El problema clásico de coloreo tiene, sin embargo, una desventaja muy grande a la hora de modelar una situación real: considera a las instancias de los recursos como equivalentes, cuando muchas veces no es así. Si los recursos que estamos asignando son distinguibles entre sí, entonces no da lo mismo que a un vértice sea asignado el color ‘1’ o ‘2’, aún cuando a cada vértice pueda serle asignado un color menor o igual al número total de

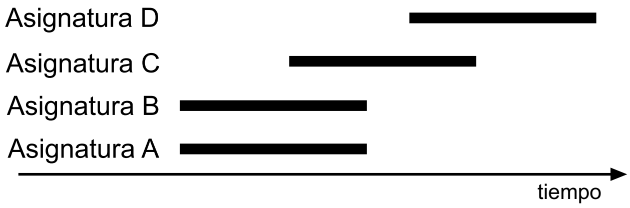

Figura 1.4: Diagrama temporal de duración de asignaturas (algunas superpuestas) que necesitan retroproyector.

recursos disponibles.

Por ejemplo, si debemos nombrar profesores calificados para las asignaturas, de entre los profesores de la facultad, no cualquier profesor puede dictar cualquier asignatura. Hay entonces, para cada asignatura, un conjunto de profesores que pueden dictarla.

Asignatura D	{Perez, Rodriguez}
Asignatura C	{Gonzalez, Rodriguez}
Asignatura B	{Gonzalez, Alvarez, Perez}
Asignatura A	{Gonzalez}

Figura 1.5: Profesores que pueden dictar cada una de las materias.

Este problema, conocido en la literatura como "list-coloring", es mas 'difícil' que el coloreo tradicional dado que es NP-completo para clases de grafos que son k-coloreables en tiempo polinomial.

1.4.2 Coloreo con listas

En el problema de coloreo con listas, cada vértice tiene asociada una lista de 'colores' o 'valores' con los cuales puede ser etiquetado. El problema consiste, entonces, en etiquetar todos los vértices del grafo de forma tal que no haya dos vértices adyacentes que utilicen la misma etiqueta y a cada uno de los vértices le sea asignada una etiqueta perteneciente a su propia lista de etiquetas.

COLOREO CON LISTAS
INSTANCIA: Un grafo $G = (V, E)$, una familia de listas finitas de enteros positivos $\mathbb{L} = \{L(v)\}_{v \in V}$.
PREGUNTA: ¿ Existe una función $f : V \to \mathbb{N}$ tal que $f(v) \neq f(w)$ si $(v, w) \in E$ y $f(v) \in L(v)$ para todo $v \in V$?

El problema de coloreo con listas, muy utilizado actualmente para la asignación de radio-frecuencias (colores) dentro de un esquema de posibles interferencias (aristas) entre células (vértices) [13], como ya hemos mencionado es NP-completo, y se utilizan distintas heurísticas para resolver casos de la vida real.

No siempre se pueden hacer simplificaciones para el problema de asignación de frecuencias, pero generalmente se puede hacer una distribución medianamente inteligente de las células de manera que el grafo resultante sea planar, propiedad utilizada como precondicion por algunas heurísticas que resuelven este problema.

El problema de coloreo con listas sigue siendo NP-completo aún para grafos tales como los planares [21], bipartitos [16], cografos [18] y grafos de intervalos [2]. Y si bien se trabaja mucho actualmente para resolver el problema para otras clases de grafos, sólo se han conseguido resoluciones polinomiales para algunas pocas clases de grafos como ser los árboles y *partial k-trees* [18].

Para nuestro ejemplo anterior de profesores y materias, si un profesor no puede dictar más de una materia por cuatrimestre, el grafo resultante es completo. En ese caso, el problema es polinomial, ya que se reduce a encontrar una correspondencia (matching) máxima en un grafo bipartito. El grafo tiene un vértice por cada materia y uno por cada profesor. El vértice

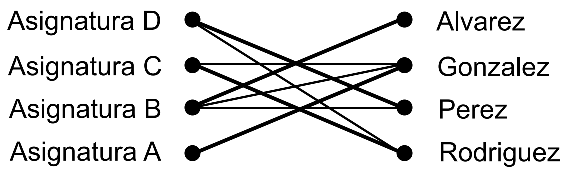

Figura 1.6: Una asignación óptima de profesores a materias corresponde a un matching máximo en el grafo.

correspondiente a la materia M es adyacente al vértice correspondiente al profesor P si P puede dictar M. Una asignación de profesores a materias es entonces un matching en ese grafo, y todas las materias serán dictadas cuando el matching máximo tenga cardinal igual a la cantidad de materias.

1.4.3 μ-coloreo

Dado un grafo G y una función $\mu : V \to \mathbb{N}$ que va de los vértices de G a los naturales, se dice que la función $f : V \to \mathbb{N}$ es un μ-coloreo de G si es un coloreo válido, y además $f(v) \leq \mu(v)$ para todo $v \in V$.

En μ-coloreo, si bien los recursos son distinguibles, tienen la propiedad de ser *ordenables*, de modo tal que si al vértice v puede serle asignado un recurso k, entonces también puede serle asignado el recurso $k - 1$, siempre que $k > 1$. Son entonces indistinguibles para el vértice v dentro del intervalo $[1, \dots, \mu(v)]$.

Es decir, un vértice necesita un recurso de 'valor' menor o igual a una cota particular para cada vértice.

El μ-coloreo describe satisfactoriamente los problemas de asignación de recursos cualificables, en donde unos recursos son 'mejores' que otros, y las asignaciones requieren una 'calidad mínima'.

Un ejemplo claro es el de asignar aulas a las materias de una facultad, donde una materia puede tener mas alumnos que la capacidad de un aula, y por lo tanto no le debería ser asignada en un coloreo, pero si puede serle asignada cualquier aula con la capacidad suficiente.

Este problema puede ser planteado como un problema de coloreo con listas, en donde cada materia tiene la lista de aulas con la capacidad suficiente para albergar a todos sus alumnos.

Pero si tenemos a las aulas ordenadas por tamaño, siendo la 1 la más grande, entonces las listas van a contener todos los elementos del 1 hasta el número del aula más chica que le puede ser asignada (excepto que haya alguna materia con tantos alumnos que no caben en ningún aula, en cuyo caso el problema es trivialmente no factible).

Por lo tanto, se puede 'resumir' esta lista con un sólo número, el del aula más chica capaz de albergar a todos sus alumnos. Llamemos a este número μ, entonces

μ-COLOREO
INSTANCIA: Un grafo $G = (V, E)$, una función $\mu : V \to \mathbb{N}$.
PREGUNTA: ¿ Existe una función $f : V \to \mathbb{N}$ tal que $f(v) \neq f(w)$ si $(v, w) \in E$ y $f(v) \leq \mu(v)$ para todo $v \in V$?

Aula 1	100
Aula 2	60
Aula 3	50
Aula 4	30
Aula 5	30
Aula 6	15

Asignatura D	90	(A 1)
Asignatura C	60	(A 1~2)
Asignatura B	50	(A 1~3)
Asignatura A	20	(A 1~5)

Figura 1.7: μ-coloreo: cantidad de alumnos de las asignaturas, y de asientos de las aulas. En el ejemplo, $\mu(Asignatura_2) = 3$, ya que puede usar las aulas de la 1 a la 3.

Notemos que el problema de k-coloreo es un caso particular de μ-coloreo, donde la función μ es constante con valor k, con lo cual μ-coloreo resulta ser NP-completo. A su vez, μ-coloreo es una subespecificación del problema de coloreo con listas, donde, como ya dijimos, la lista asociada a v tiene la forma $\{1, \ldots, \mu(v)\}$. Desde el punto de vista de la complejidad algorítmica, esto hace que tenga especial sentido estudiar la complejidad del problema en las clases de grafos en las cuales k-coloreo es polinomial y coloreo con listas es NP-completo o tiene complejidad abierta.

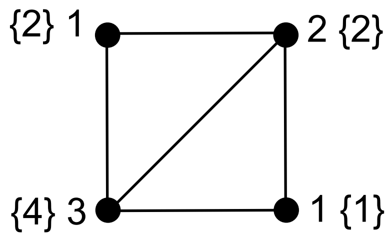

Figura 1.8: Ejemplo de grafo μ-coloreado: Entre parentesis la funcion μ, afuera el valor del μ-coloreo.

Capítulo 2

Grafos M-perfectos

Definiremos en este capítulo una clase de grafos inspirada en los grafos perfectos, pero basada en el μ-coloreo en lugar del k-coloreo. Luego, trataremos de encontrar una caracterización para esta clase.

Sabemos que G es un grafo perfecto si y sólo si $\chi(H) = \omega(H)$ para todo subgrafo inducido H de G. Es decir, el tamaño de la clique máxima es igual a la cantidad mínima de colores necesarios para colorear el grafo. Sabemos que no se puede colorear un grafo con menos colores que el tamaño de la clique máxima, por lo que de no G ser $Perfecto$, un subgrafo H de G debe tener clique máxima de tamaño P y H no ser P-coloreable. Las cliques de H son todas P-coloreables pero H no lo es.

Entonces, podemos traducir la versión anterior de la definición de grafos perfectos de la siguiente forma: Un grafo G es perfecto si y sólo si, para todo H subgrafo inducido de G, y para todo k, las cliques de H son k-coloreables $\Leftrightarrow$ H es k-coloreable.

2.1 De grafos perfectos a M-perfectos

Usando esta definición podemos decir entonces que un grafo G es *M-perfecto* si y sólo si, para todo H subgrafo inducido de G, y toda función $\mu : V \to \mathbb{N}$, las cliques de H son μ-coloreables $\Leftrightarrow$ H es μ-coloreable.

Recordemos que dado un grafo $G = (V, E)$, y una función $\mu : V \to \mathbb{N}$, se dice que G es μ-coloreable si y sólo si existe una función $f : V \to \mathbb{N}$ tal que para todo par de vértices adyacentes v y w, $f(v) \neq f(w)$ (equivalentemente, $f(V)$ es un coloreo válido de G), y además $f(v) \leq \mu(v)$ para todo v.

Veremos que los grafos M-perfectos son exactamente los cografos, o sea aquellos que no contienen como subgrafo inducido a P_4.

Esto hace que tengan algunas propiedades interesantes en común con los grafos perfectos: son una clase auto-complementaria (un grafo es μ-perfecto si y sólo su complemento lo es), tienen reconocimiento polinomial [9] y están caracterizados por subgrafos prohibidos. Además, en el Capítulo 3, mostraremos un algoritmo polinomial para μ-coloreo en grafos M-perfectos.

Llamamos *good* a un grafo G cuando $\chi(G) = \omega(G)$. Los grafos good no son equivalentes a los perfectos, los incluyen en forma propia (un grafo es perfecto cuando todo subgrafo inducido es good). Por ejemplo, el grafo G de la figura 2.1 es good ($\chi(G) = \omega(G) = 3$), pero el subgrafo inducido resultante de eliminar el vértice x no es good ($\chi(G) = 3, \omega(G) = 2$), y por lo tanto G no es perfecto.

De la misma forma, podemos llamar *M-good* a aquellos grafos $G = (V, E)$ tales que para toda función $\mu : V \to \mathbb{N}$, las cliques de G son μ-coloreables $\Leftrightarrow$ G es μ-coloreable. Pero en este caso, veremos que la propiedad de ser M-good es hereditaria y por lo

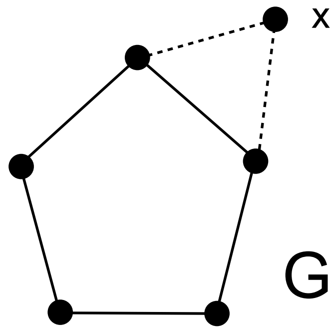

Figura 2.1: Ejemplos: grafo good no perfecto.

tanto los grafos M-good coinciden con los M-perfectos.

Teorema 2.1. *La propiedad de ser M-good es una propiedad hereditaria.*

Demostración: Queremos ver que para cualquier grafo G y cualquier subgrafo inducido H de G, G es M-good $\Rightarrow$ H es M-good.

Supongamos que G es M-good, y sea H un subgrafo inducido de G. Sea $\mu : V(H) \to \mathbb{N}$ una función para la cual todas las cliques de H son μ-coloreables.

Extendamos esta función μ definida para los vértices de H a una función μ' definida en el conjunto de vértices de G, $\mu' : V(G) \to \mathbb{N}$, de la siguiente forma: $\mu'(v) = \mu(v)$ para todo v en H y $\mu'(w) = |V(G)|$ para todo w en $V(G) \setminus V(H)$.

Sea K una clique de G y $K_H = K \cap H$. K_H es un completo de H, por lo tanto está incluido en alguna clique de H y puede ser μ-coloreado y entonces μ'-coloreado. Podemos extender ese μ'-coloreo a K, ya que los vértices de $K \setminus K_H$ tienen como cota $|V(G)|$ que es mayor o igual al tamaño de la clique K.

Luego todas las cliques de G pueden ser μ'-coloreadas, y por lo tanto, como G es M-good, G puede ser μ'-coloreado. Pero ese μ'-coloreo, restringido a H, resulta un μ-coloreo, ya que $\mu'|_H = \mu$. Luego H es M-good. □

2.2 Propiedades de M-perfección

2.2.1 M-perfectos no triviales

Ser M-perfecto no es una propiedad trivial. Existen los grafos M-perfectos, y no todos los grafos lo son. Por ejemplo, el grafo C_4 es M-perfecto mientras que el P_4 no lo es (Figura 2.2).

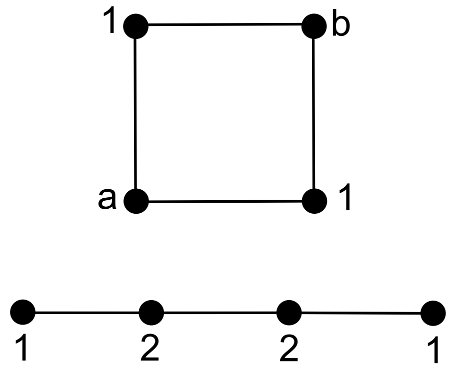

Figura 2.2: Ejemplos: un grafo M-perfecto (C_4) y un grafo no M-perfecto (P_4).

Proposición 2.1. *El grafo C_4 es M-perfecto.*

Demostración: Supongamos que no es cierto. Entonces tiene que existir una función μ para la cual las cliques de C_4 son μ-coloreables pero C_4 no lo es. La función μ no puede valer 1 para dos vértices adyacentes, por lo que, en el peor de los casos, para dos vértices no adyacentes valdrá 1, y para los otros dos valdrá a y b respectivamente, con $a, b > 1$ (Ver Figura 2.2). Pero en ese caso el grafo μ-coloreable, ya que, al ser $2 \leq a \leq b$, basta colorear a y b con 2 para obtener un μ-coloreo del mismo. Llegamos así a un absurdo que proviene de suponer que C_4 no es M-perfecto. $\square$

Proposición 2.2. *El grafo P_4 no es M-perfecto.*

Demostración: Sea μ definida como en la Figura 2.2. Tratemos de μ-colorear P_4 desde uno de los extremos. El único color que podemos asignarle al extremo es 1. Al siguiente vértice, solamente podemos asignarle el 2 (ya que tenemos al extremo de vecino, que ya está usando el 1). Al tercer vértice, solamente podemos asignarle el 1, ya que el 2 está siendo usado por su vecino. Pero nos encontramos con que no podemos asignarle ningún color al último vértice, ya que solamente podríamos asignarle el 1, pero este está siendo usado por su único vecino. Sin embargo, no es difícil ver que las cliques del grafo son μ-coloreables. $\square$

Observación 2.1. Observemos que, por definición, un grafo M-perfecto es también perfecto. Pero un grafo perfecto no es necesariamente M-perfecto, ya que P_4 es perfecto pero no es M-perfecto.

2.2.2 Algunos lemas base

Lema 2.2. *Dados $G = (V, E)$ y $\mu : V \to \mathbb{N}$ tal que $\mu(v) \neq \mu(w)$ $\forall v \neq w$. Entonces G es μ-coloreable.*

Demostración: Basta con colorear cada vértice con el color máximo, $f(v) = \mu(v)$ $\forall v \in V$. $\square$

Lema 2.3. *Dados $G = K_n$ y $\mu : V \to \mathbb{N}$ tal que $\mu(v) < n\ \forall v$, entonces G no es μ-coloreable.*

Demostración: Basta con ver que tenemos menos colores disponibles que vértices, por lo que 2 vértices tendrían que tener el mismo color, lo cual no es un coloreo válido ya que es un completo en donde todos son vecinos. □

Lema 2.4. *Dados $G = (V, E)$, $\mu : V \to \mathbb{N}$, y un orden de los vértices $v_1, \dots, v_n$ tal que $i \leq \mu(v_i)\ \forall 1 \leq i \leq n$, entonces G es μ-coloreable.*

Demostración: Basta con colorear cada vértice con el índice del ordenamiento, $f(v_i) = i\ \forall 1 \leq i \leq n$. □

Lema 2.5. *Dado $G = K_n$, $\mu : V \to \mathbb{N}$, y un orden de los vértices $v_1, \dots, v_n$ tal que $\mu(v_i) \leq \mu(v_j)$ para $i \leq j$, entonces G es μ-coloreable $\Leftrightarrow i \leq \mu(v_i)\ \forall 1 \leq i \leq n$.*

Demostración: Si $i \leq \mu(v_i)\ \forall 1 \leq i \leq n$, entonces G es coloreable con $F(v_i) = i$ □

Supongamos tengo un orden como el descripto, y al colorear los v_i de menor a mayor con $F(v_i) = i$ me encuentro con que no puedo asignar $f(v_j) = j$. Eso quiere decir que hay al menos $\mu(v_j)$ vértices ya coloreados, para los cuales la función μ tiene asiciados valores mas chicos que $\mu(v_j)$, lo cual es un completo de tamaño $\mu(v_j) + 1$ para el cual tengo sólo $\mu(v_j)$ colores. Absurdo de suponer G no μ-coloreable. □

$G = K_n \Rightarrow f(v) \neq f(w)\ \forall v \neq w$ y por lo tanto, f define una relación de orden total y estricto. □

Lema 2.6. *Dado $G = K_n$, $\mu : V \to \mathbb{N}$, y un μ-coloreo válido $f : V \to \mathbb{N}$, entonces podemos crear otro μ-coloreo válido f' intercambiando $f'(v) = f(w)$ y $f'(w) = f(v)$ para cualesquiera dos vértices v y w de G tales que $\mu(v) \geq f(w)$ y $\mu(w) \geq f(v)$.*

Demostración: Para todo par de vértices v y w en V, $N[v] = N[w]$, lo que quiere decir que todo vecino de w es vecino de v, y no hay ningún vecino de uno que no lo sea del otro. Sabemos que $f'(v) \neq f'(w)$, sino existiría ese conflicto en el coloreo original $(f(w) \neq f(v))$.

Supongamos ahora que al intercambiar los colores aparece un conflicto cromático entre v e y, $y \neq v$, $y \neq w$, es decir $f'(v) = f'(y)$. Pero entonces, como $f'(v) = f(w) = f(y) = f'(y)$, el conflicto existía ya entre w e y. Por lo tanto, el intercambio tiene que ser válido. □

2.3 Caracterización de grafos M-perfectos

Queremos ver que los grafos M-perfectos son exactamente los cografos. Para eso necesitamos el siguiente resultado previo.

Definición 1. Dados un grafo $G = (V, E)$, una función $\mu : V \to \mathbb{N}$ y un μ-coloreo $f : V \to \mathbb{N}$ válido para G, decimos que f es un *μ-coloreo minimal* de G, si y sólo si para cada vértice $v \in V$ y para cada entero positivo $j < f(v)$, existe $w \in N(v)$ tal que $f(w) = j$.

Lema 2.7. *Sea $G = (V, E)$ un grafo y $\mu : V \to \mathbb{N}$. Si G es μ-coloreable entonces admite un μ-coloreo minimal.*

Demostración: Sea $G = (V, E)$ un grafo, $\mu : V \to \mathbb{N}$ tal que G es μ-coloreable y f un μ-coloreo válido de G. Entonces podemos transformarlo en uno minimal. Si hay un vértice v para el cual su vecindad no usa todos los colores entre 1 y $f(v)$, entonces simplemente cambiamos la función f, asignándole a v el menor color disponible entre sus vecinos. De esta manera obtenemos otro μ-coloreo válido de G. Repetimos este proceso mientras exista un vértice en esas condiciones, si no existe tal vértice, entonces f es, por definición, minimal. Notemos que en cada paso la suma total de colores usados disminuye y es un entero positivo, con lo cual el proceso no se puede repetir indefinidamente, y por lo tanto en un número finito de pasos se llega a un μ-coloreo minimal. □

Teorema 2.8. *Sea G un grafo P_4-free y supongamos que existe un vértice x de G tal que $G \setminus x$ es M-perfecto. Sea μ una función tal que las cliques de G son μ-coloreables, y sea f un μ-coloreo minimal de $G \setminus x$. Entonces que f no pueda extenderse a G coloreando x con un color menor o igual a T implica que existe un completo $H \subseteq N(x)$ de tamaño T y tal que $f(H) = \{1, \ldots, T\}$.*

Demostración: Sea G un grafo P_4-free y supongamos que existe un vértice x de G tal que $G \setminus x$ es M-perfecto. Sea μ una función tal que las cliques de G son μ-coloreables, y sea f un μ-coloreo minimal de $G \setminus x$.

Por inducción en T. Si f no puede extenderse a G coloreando x con color 1, entonces existe $v \in N(x)$ tal que $f(v) = 1$. En ese caso H es el subgrafo de G inducido por v.

Supongamos que vale para $T = s - 1$ y veamos qué pasa con $T = s$, $s \geq 2$. Si f no puede extenderse a G coloreando x con color menor o igual a s, en particular no puede extenderse a G coloreando x con color menor o igual a $s - 1$, por lo que, por hipótesis inductiva, tenemos un completo H de tamaño $s - 1$ en la vecindad de x que utiliza los colores de 1 a $s-1$. Por otro lado, como x no puede usar el color s, tiene que haber un vértice v en $N(x)$ tal que $f(v)=s$. Consideremos el subgrafo de $G\setminus x$ inducido por $\{w\in G\setminus x : f(w)\leq s-1\}\cup\{v\}$, llamémoslo $\widetilde{G}$, y llamemos $\widetilde{f}$ al coloreo f restringido a $\widetilde{G}\setminus v$. Por la minimalidad de f resulta que $\widetilde{f}$ es minimal y no puede extenderse a $\widetilde{G}$ coloreando v con color menor o igual a $s-1$, luego, por hipótesis inductiva, tenemos un completo F de tamaño $s-1$ en la vecindad de v que utiliza los colores de 1 a $s-1$.

Si $H=F$ entonces $H\cup\{v\}$ es un completo en la vecindad de x que utiliza los colores de 1 a s. Supongamos que no son iguales. Entonces $F\setminus H$ y $H\setminus F$ tienen el mismo cardinal y utilizan los mismos colores. Sea v_H en $H\setminus F$, y sea v_F en $F\setminus H$ tal que $f(v_F)=f(v_H)$. Como f es un μ-coloreo válido de $G\setminus x$, v_F y v_H no son adyacentes. Como G es P_4-free, v_H,x,v,v_F no inducen un P_4, luego x es adyacente a v_F o v es adyacente a v_H.

Si todos los vértices de $H\setminus F$ son adyacentes a v, entonces $H\cup\{v\}$ es un completo de tamaño s en la vecindad de x que usa los colores $1,\ldots,s$.

Supongamos entonces que el conjunto $H_v=\{w\in H : (w,v)\notin E(G)\}$ es no vacío, y definamos $F_v=\{z\in F : \exists z_H\in H_v$ con $f(z)=f(z_H)\}$. Notemos que F_v y H_v tienen el mismo cardinal y usan los mismos colores.

Como H_v es anticompleto a v, resulta que F_v debe ser completo a x, sino tendríamos subgrafos P_4 de la forma h_v,x,v,f_v, $h_V\in H_v, f_v\in F_v$.

Si $H\setminus H_v$ es vacío, entonces $F=F_v$ es completo a x y $F\cup\{v\}$ es un completo de tamaño s en la vecindad de x que usa los colores $1,\ldots,s$.

Supongamos que $H\setminus H_v$ es no vacío, y veamos que F_v es completo a $H \setminus H_v$.

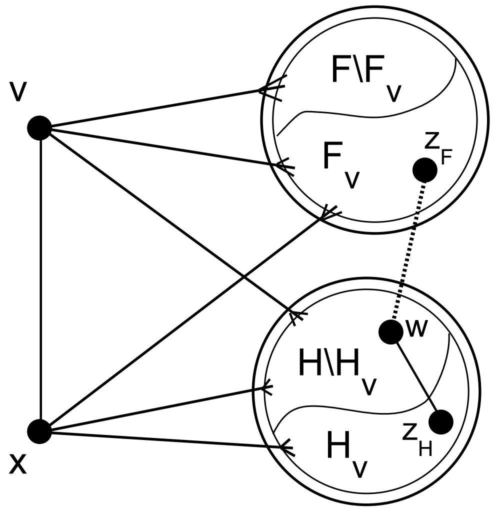

Figura 2.3: F_v tiene que ser completo a $H \setminus H_v$ para evitar $P_4 = \{z_H, w, v, z_F\}$.

Sean $z_F \in F_v$ y $w \in H \setminus H_v$. Sea $z_H \in H_v$ tal que $f(z_H) = f(z_F)$. Luego z_F no es adyacente a z_H, y por ser H completo, z_H y w son adyacentes. Además, w es adyacente a v por estar en $H \setminus H_v$, pero z_H no lo es por estar en H_v. Como z_H, w, v, z_F no inducen un P_4, debe ser w adyacente a z_F. Por lo tanto F_v es completo a $H \setminus H_v$ (ver Figura 2.3).

Entonces $(H \cup F_v \cup \{v\}) \setminus H_v$ es un completo en $N(x)$ de tamaño s y que utiliza los colores de 1 a s en el coloreo f. $\square$

Teorema 2.9. *Sea G un grafo. G es M-perfecto $\Leftrightarrow$ G es P_4-free.*

Demostración: $\Rightarrow$) Vimos que P_4 no es M-perfecto, y como la clase es hereditaria, un grafo M-perfecto debe ser P_4-free.

$\Leftarrow$) Supongamos que existe un grafo P_4-free y no M-perfecto. Sea G uno minimal (G es P_4-free, no es M-perfecto, pero para todo vértice x, $G \setminus x$ es M-perfecto).

Sea $\mu : V(G) \to \mathbb{N}$ tal que las cliques de G son μ-coloreables pero G no lo es. Sea x un vértice de G con $\mu(x)$ máxima. $G \setminus x$ es M-perfecto, y si las cliques de G son μ-coloreables, también lo son las de $G \setminus x$, luego $G \setminus x$ es μ-coloreable. Sea f un μ-coloreo de $G \setminus x$.

Como G no es μ-coloreable, f no se puede extender a un μ-coloreo de G. Luego por el Teorema 2.8, $N(x)$ contiene un completo de tamaño $\mu(x)$. Pero entonces G contiene un completo de tamaño $\mu(x) + 1$ en el cual las cotas de todos sus vértices no superan $\mu(x)$ (ya que lo habíamos elegido con μ máxima). Eso se contradice con la hipótesis de que todas las cliques de G sean μ-coloreables.

Luego no existe un grafo P_4-free y no M-perfecto minimal, que es lo que queríamos demostrar. $\square$

Capítulo 3

Algoritmos y complejidades de μ-coloreo

Sabemos que μ-coloreo es NP-completo para el caso general, ya que k-coloreo es NP-completo y podemos transformar una instancia de k-coloreo a una de μ-coloreo en tiempo polinomial poniendo simplemente $\mu(v) = k$ para todo $v \in V$. También es fácil ver que una solución del problema puede ser chequeada en tiempo polinomial: dando como certificado el coloreo, basta chequear que se trata de un coloreo válido y que para todo $v \in V$, el color de v es menor o igual a $\mu(v)$.

En este capítulo estudiaremos la complejidad computacional del problema en grafos bipartitos y cografos. Tiene sentido estudiar estas clases de grafos porque en ellas se sabe que k-coloreo es polinomial y coloreo con listas es NP-completo (o sea que la complejidad difiere, salvo que P=NP).

El resultado principal de este capítulo es que para cografos μ-coloreo es polinomial mientras que para grafos bipartitos es NP-completo, con lo cual el problema es realmente un problema nuevo estrictamente intermedio entre k-coloreo y coloreo con listas, es decir, no es equivalente a ninguno de ellos salvo que

P=NP (al menos no es equivalente para cualquier clase de grafos, sí para la clase de grafos general, ya que los tres son problemas NP-completos).

3.1 Cografos

El algoritmo secuencial de coloreo es un algoritmo goloso que consiste en dar un orden a los vértices del grafo y luego colorear cada vértice con el menor color que no esté siendo utilizado ya por uno de sus vecinos.

El algoritmo secuencial no resuelve el problema de coloreo en el caso general. Sin embargo, para cualquier grafo existe un orden de los vértices tal que dicho algoritmo colorea en forma óptima el grafo. El problema es que no se conoce una forma de hallar ese orden en tiempo polinomial para un grafo general.

A partir del Teorema 2.8 podemos demostrar el siguiente resultado.

Teorema 3.1. *El algoritmo secuencial de coloreo aplicado a un cografo G con sus vértices ordenados de forma no decreciente por el valor de μ, da un μ-coloreo, si G es μ-coloreable. Además, este coloreo es minimal y utiliza sólo los primeros $\chi(G)$ colores.*

Demostración: Por inducción en la cantidad de vértices del grafo. El teorema vale para $|V(G)| = 1$. Supongamos que vale para todo G tal que $|V(G)| < n$. Sea G un cografo de n vértices y la función μ tal que G sea μ-coloreable. Tomemos entonces los vértices de G ordenados de forma no decreciente con respecto al valor de la función μ, para obtener la secuencia $S = \{v_1, \ldots, v_n\}$.

Al ser G es μ-coloreable, $G \setminus \{v_n\}$ también lo es. Sea f el μ-coloreo válido para $G \setminus \{v_n\}$ obtenido por el algoritmo aplicado a los vértices $\{v_1, \ldots, v_{n-1}\}$, el cual sabemos que existe por hipótesis inductiva.

Por el Teorema 2.8, sabemos que si no se puede extender el coloreo f a v_n utilizando un color menor o igual a $\mu(v_n)$ tiene que existir un completo K de tamaño $\mu(v_n)$ en la vecindad de v_n.

Pero como tomamos los vértices ordenados de forma tal que $\mu(v_i) \leq \mu(v_j)$ para $i < j$, entonces tiene que haber un completo $K \cup \{v_n\}$ de tamaño $\mu(v_n) + 1$, formado por vértices que no pueden usar colores mayores a $\mu(v_n)$, por lo que nopuede ser μ-coloreable.

Llegamos así a un absurdo que proviene de suponer que no se podía μ-colorear G con el algoritmo goloso propuesto.

Por la forma en que se define el color de cada vértice, el coloreo obtenido resulta minimal. Veamos que utiliza sólo los colores de 1 hasta $\chi(G)$ inclusive.

Por la observación anterior, si el algoritmo utiliza el color k para pintar un vértice significa que el grafo contiene un completo de tamaño k, con lo cual no es $(k-1)$-coloreable. Por lo tanto el algoritmo utiliza solamente los primeros $\chi(G)$ colores. □

Un corolario interesante de este teorema es el siguiente resultado, demostrado anteriormente por Chvátal en [8].

Corolario 3.1.1. *El algoritmo secuencial de coloreo resuelve el problema clásico de coloreo (pintar un grafo utilizando la menor cantidad posible de colores) para cografos, independientemente del orden de los vértices.*

Demostración: Un coloreo válido para un grafo puede verse como un μ-coloreo donde $\mu(v) = n$ para todo vértice v (y n es la cantidad de vértices del grafo). Como los valores de la función μ son todos iguales, cualquier orden de los vértices respeta las precondiciones del Teorema 3.1, y por lo tanto el algoritmo colorea un cografo G con $\chi(G)$ colores. $\square$

Teorema 3.2. *El algoritmo de μ-coloreo descrito en el Teorema 3.1 tiene complejidad $O(n \log n + m)$, donde n es la cantidad de vértices del grafo y m la cantidad de aristas.*

Demostración: Ordenar los vértices por μ de menor a mayor tiene orden $O(n \log n)$. Luego, coloreamos cada vértice v con el color más pequeño posible. Encontrar ese color tiene orden $O(d(v))$, ya que hay que buscar en el vecindario del vértice. Por ejemplo, se puede ir marcando en un vector los colores utilizados y luego buscar el menor no marcado (siempre va a ser menor o igual a $d(v)+1$). Como $\sum_{v \in V(G)} d(v) = 2m$ la complejidad total del algoritmo es $O(n \log n + m)$. $\square$

En la práctica, si al modelar niestro problema nos encontramos con un cografo que no es *mu*-coloreable, nos interesará saber donde esta el problema para, de ser posible, corregirlo. Por ejemplo, si estamos asignando profesores a materias como en el ejemplo del capítulo 1, podríamos contratar un profesor con experiencia en el area de las materias que pertenecen al completo no μ-coloreable que encontremos.

Utilizando las ideas de la demostración del Teorema 2.8, el algoritmo anterior puede ser mejorado de manera tal que encuentre un completo no μ-coloreable si el grafo no es μ-coloreable. Es decir, convertirlo en un algoritmo robusto. La segunda parte del algoritmo, al encontrarse con un vértice v_j para el cual sus vecinos han sido coloreados con todos los colores del 1 al $\mu(v_j)$, y por ende no puede ser μ-coloreado, sería:

$K := \{v_j\}$;

$L := \{v_i \in N(v_j) : i < j\}$;

`para` c `desde` $\mu(v_j)$ `hasta` 1 `hacer`

buscar v_k en L tal que $f(v_k) = c$;

$K := K \cup \{v_k\}$;

$L := L \cap N(v_k)$;

`fin para`

`devolver` K;

Teorema 3.3. *El algoritmo robusto es correcto y tiene complejidad $O(n \log n + m)$, donde m es la cantidad de aristas del grafo.*

Demostración: Por el Teorema 2.8, sabemos que existe un completo de tamaño $\mu(v_j)$ en la vecindad de v_j, ya coloreado, y que utiliza en f los colores de 1 a $\mu(v_j)$. Los candidatos a formar parte de ese completo son los vértices de $\{v_1, \ldots, v_{j-1}\}$ que son vecinos de v_j, y esos son inicialmente los elementos del conjunto L. Luego, para cada color c desde $\mu(v_j)$ descendiendo hasta 1, el algoritmo va a buscar un vértice w_c (igual a v_k para algún $1 \leq k < j$) tal que $f(w_c) = c$ y w_c sea adyacente a $w_{c+1}, \ldots, w_{\mu(v_j)}, v_j$. Está claro que si en cada paso existe ese vértice, el algoritmo termina devolviendo el subgrafo completo buscado.

Por lo tanto, sólo resta demostrar la existencia de ese vértice en cada paso. Vamos a demostrar por inducción en la cantidad de pasos i que si $0 \leq i \leq \mu(v_j) - 1$ y tenemos un completo $w_{\mu(v_j)}, w_{\mu(v_j)-1}, \ldots, w_{\mu(v_j)-i+1}$ entonces para todo $1 \leq r \leq \mu(v_j) - i$ existe un vértice v_k con $k < j$, adyacente a todos ellos y a v_j, y tal que $f(v_k) = r$.

Para $i = 0$, dado que v_j no puede colorearse con un color menor o igual a $\mu(v_j)$, es claro que para todo $1 \leq r \leq \mu(v_j)$ existe un vértice v_k con $k < j$, adyacente a v_j, y tal que $f(v_k) = r$.

Supongamos que vale para $i = s$ y veamos que vale para $i = s+1$, suponiendo $s + 1 \leq \mu(v_j) - 1$. Tenemos un completo $w_{\mu(v_j)}, w_{\mu(v_j)-1}, \ldots, w_{\mu(v_j)-s}$ y sea $1 \leq r \leq \mu(v_j) - s - 1$. Por hipótesis inductiva, existe un vértice v_k con $k < j$, adyacente a $v_j, w_{\mu(v_j)}, w_{\mu(v_j)-1}, \ldots, w_{\mu(v_j)-s+1}$ y tal que $f(v_k) = r$. Si v_k es adyacente a $w_{\mu(v_j)-s}$, entoces los w junto con v_k son el completo que buscaba. Si no, dado que $w_{\mu(v_j)-s}$ fue coloreado con el menor color disponible y ese fue $\mu(v_j) - s$, resulta que debe existir un $v_{k'}$ con $k < j$, adyacente a $w_{\mu(v_j)-s}$ y tal que $f(v_{k'}) = r$. Como el grafo es un cografo, $v_{k'} w_{\mu(v_j)-s} w v_k$ no pueden inducir un P_4 para ningún $w \in \{v_j, w_{\mu(v_j)}, w_{\mu(v_j)-1}, \ldots, w_{\mu(v_j)-s+1}\}$. Como v_k y $v_{k'}$ no pueden ser adyacentes ya que utilizan el mismo color y estamos suponiendo que v_k no es adyacente a $w_{\mu(v_j)-s}$, debe ser $v_{k'}$ adyacente a w. Entonces $v_{k'}$ es el vértice buscado, lo cual concluye la demostración de correctitud del algoritmo.

Si el grafo viene dado por listas de adyacencia ordenadas, la representación del conjunto L se puede implementar con una lista ordenada, con lo cual en cada paso la búsqueda tendrá orden $O(|L|)$, la intersección con la vecindad del vértice v tendrá orden $O(|L|+|N(v)|)$, y la lista resultante para el siguiente paso tendrá un tamaño $O(|N(v)|)$. Si los vértices del completo obtenido son $w_1, \ldots, w_s$ entonces la complejidad de esta parte del algoritmo robusto es $O(\sum_{i=1}^{s} d(w_i))$ que es $O(m)$. Luego la complejidad total del algoritmo robusto es $O(n \log n + m)$. $\square$

Jansen y Scheffler [18] demostraron que coloreo con listas es NP-completo para cografos, por lo tanto μ-coloreo es "más fácil" que coloreo con listas, salvo que P=NP.

3.2 Grafos bipartitos

De los resultados anteriores podemos observar que si un cografo G es μ-coloreable entonces puede ser μ-coloreado usando los primeros $\chi(G)$ colores. Esto no ocurre para grafos bipartitos, ni siquiera para árboles.

Teorema 3.4. *Definamos la familia de árboles $\{T_n\}_{n\in\mathbb{N}}$, y su respectiva familia de funciones $\{\mu_n\}_{n\in\mathbb{N}}$ de la siguiente forma: $T_1 = \{v\}$ es un árbol trivial, con $\mu_1(v) = 1$. El árbol T_{n+1} se obtiene conectando las raíces de los árboles $T_1, \ldots, T_n$ con una nueva raíz w. La función μ_{n+1} extiende a $\mu_1, \ldots, \mu_n$ y se define $\mu_{n+1}(w) = n + 1$. El árbol T_n necesita n colores para ser μ_n-coloreado, y cada vértice v debe ser coloreado con el color $\mu_n(v)$. Además, T_n contiene 2^{n-1} vértices.*

Demostración: Demostrémoslo por inducción en n. El árbol T_1 tiene $2^0 = 1$ vértice, y necesita un color para ser μ_1-coloreado. Su único vértice v tiene $\mu_1(v) = 1$, por lo tanto la afirmación vale para $n = 1$.

Supongamos ahora que vale para $n \leq k$ y veamos qué ocurre con T_{k+1}. Como T_{k+1} se obtiene conectando las raíces de los árboles $T_1, \ldots, T_k$ con una nueva raíz w y μ_{k+1} extiende a $\mu_1, \ldots, \mu_k$, por hipótesis inductiva cada vértice v de $T_1, \ldots, T_k$ debe utilizar color $\mu_{k+1}(v)$. En particular, la raíz del árbol T_i utiliza color i, para $1 \leq i \leq k$, y por lo tanto w debe utilizar color $k + 1 = \mu_{k+1}(w)$. Falta ver que $|V(T_{k+1})| = 2^k$. Por hipótesis inductiva $|V(T_i)| = 2^{i-1}$, para $1 \leq i \leq k$. Así podemos calcular $|V(T_{k+1})| = \sum_{1\leq i\leq k} 2^{i-1} + 1 = 2^k - 1 + 1 = 2^k$. □

Esta familia es óptima en cuanto a cantidad de vértices. De hecho, vale la siguiente propiedad.

Teorema 3.5. *Sea T un árbol y μ una función tal que T es μ-coloreable. Entonces T puede ser μ-coloreado con los primeros a lo sumo $log_2(|V(T)|) + 1$ colores.*

Demostración: Dado un árbol T, y una función μ, probaremos que si un μ-coloreo minimal de T utiliza r colores, entonces T tiene al menos 2^{r-1} vértices.

Demostrémoslo por inducción en r. Vale obviamente para $r = 1$. Supongamos ahora que vale para $r \leq k$ y veamos qué ocurre con $r = k + 1$.

Tomemos un μ-coloreo de T minimal. Si hay un vértice v que utiliza color $k + 1$, entonces v tiene entre sus vecinos k que utilizan los colores $1, \ldots, k$. En $T \setminus \{v\}$ hay entonces k árboles disjuntos $T_1, \ldots, T_k$ y cada árbol T_i está coloreado en forma minimal y contiene un vértice que utiliza color i. Por hipótesis inductiva $|V(T_i)| \geq 2^{i-1}$ y por lo tanto $|V(T)| \geq \sum_{1 \leq i \leq k} |V(T_i)| + 1 \geq \sum_{1 \leq i \leq k} 2^{i-1} + 1 = 2^k$.

Tomemos un μ-coloreo de T que minimice $\max_{v \in V(T)} f(v)$. Ese coloreo va a ser obviamente minimal. Luego si utiliza r colores, entonces $|V(T)| \geq 2^{r-1}$ y por lo tanto $r \leq log_2(|V(T)|) + 1$. □

Para grafos bipartitos se puede obtener un resultado similar, pero ahora el tamaño de los grafos es lineal en n.

Teorema 3.6. *Definamos la familia de grafos bipartitos $\{B_n\}_{n \in \mathbb{N}}$ y su correspondiente familia de funciones $\{\mu_n\}_{n \in \mathbb{N}}$ de la siguiente forma: $B_1 = \{v\}$ un grafo trivial con $\mu_1(v) = 1$. El grafo bipartito $B_{n+1} = (V, W, E)$ con $V = \{v_1, \ldots, v_n\}$, $W = \{w_1, \ldots, w_n\}$; v_i adyacente a w_j para todo $i \neq j$; v_n adyacente a w_n, y v_i no adyacente a w_i para $i < n$; $\mu_{n+1}(v_i) = \mu_{n+1}(w_i) = i$ para los $i < n$; $\mu_{n+1}(v_n) = n$ y $\mu_{n+1}(w_n) = n + 1$. El grafo bipartito*

B_n necesita n colores para ser μ_n-coloreado, y contiene $2n-2$ vértices (para $n \geq 2$).

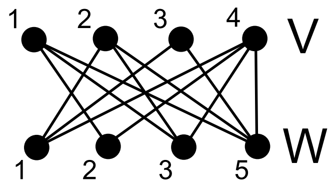

Figura 3.1: B_5 y los valores de μ (y de coloreo necesario) para cada vértice.

Demostración: Veamos que hay una sola forma f de μ_{n+1}-colorear B_{n+1}, y que ésta usa $n+1$ colores, ya que $f(v) = \mu_{n+1}(v)$ para todo vértice v de B_{n+1}.

Primero probemos que $f(v_i) = f(w_i) = i$ para $i < n$, por inducción en i. Es cierto para $i = 1$, ahora supongamos que vale para $i \leq k$ y veamos qué pasa para $i = k+1$. El vértice v_{k+1} es adyacente a todos los w_j con $j \leq k$, los cuales, por hipótesis inductiva, usan los colores de 1 a k. Pero $\mu(v_{k+1}) = k+1$, por lo tanto sólo puede ser coloreado con $k+1$. De forma análoga $f(w_{k+1}) = k+1$.

Por último, v_n es adyacente a todos los w_j con $j < n$, los cuales, como acabamos de probar, usan los colores de 1 a $n-1$. Y como $\mu(v_n) = n$, tiene que ser $f(v_n) = n$. Ahora, w_n es adyacente a todos los v_j con $j \leq n$, que utilizan los colores de 1 a n, y como $\mu(w_n) = n+1$, sólo puede ser $f(w_n) = n+1$. □

Análogamente, vale la siguiente propiedad.

Teorema 3.7. *Sea B un grafo bipartito, y μ una función tal que B es μ-coloreable. Entonces B puede ser μ-coloreado usando a lo sumo los primeros $\frac{(|V(B)|+2)}{2}$ colores.*

Demostración: Sea $B = (V, W, E)$ y tomemos un μ-coloreo f de B que minimice $\sum_{v \in V(B)} f(v)$. Ese coloreo va a ser obviamente minimal. Supongamos que hay un vértice v en V que utiliza color k. Por ser f minimal, v tiene que tener al menos $k-1$ vecinos $w_1, \ldots, w_{k-1}$ en W tales que $f(w_i) = i$. Por lo tanto, $|W| \geq k-1$. Por otra parte, $f(w_{k-1}) = k-1$ y por ser f minimal, w_{k-1} debe tener al menos $k-2$ vecinos en V que utilicen los colores de 1 a $k-2$. Como $f(v) = k$, todos ellos son distintos de v y por lo tanto $|V| \geq k-2+1 = k-1$, con lo cual $|V(B)| \geq 2(k-1)$ y por lo tanto $k \leq \frac{(|V(B)|+2)}{2}$. □

Hujter y Tuza [16] demostraron que coloreo con listas es NP-completo para grafos bipartitos. Veamos que lo mismo ocurre para μ-coloreo.

Teorema 3.8. *El problema de μ-coloreo es NP-completo para grafos bipartitos.*

Demostración: Sea $\langle G, \mathbb{L} \rangle$ una instancia de coloreo con listas en grafos bipartitos, es decir, $G = (X, Y, E)$ y para cada vértice $v \in V(G)$, tenemos una lista finita $L(v) \subseteq \mathbb{N}$ de colores posibles para v. Sea $k = |\bigcup_{v \in V(G)} L(v)|$. Sin pérdida de generalidad podemos suponer que $L(v) \subseteq \{1, \ldots, k\}$ para todo v. Construimos G' agregandole a G dos conjuntos de k vértices, $X' = \{x'_1, \ldots, x'_k\}$ e $Y' = \{y'_1, \ldots, y'_k\}$ tales que X, Y, X', Y' sean disjuntos dos a dos. Tomamos la bipartición $(X \cup X', Y \cup Y')$ del nuevo grafo G', y para cada $x \in X$, $y \in Y$, e $i, j \in \{1, \ldots, k\}$, definimos las siguientes relaciones de adyacencia: x'_i es adyacente a y'_j si y sólo si $i \neq j$; x'_i es adyacente a y si y sólo si $i \notin L(y)$; y'_i es adyacente a x si y sólo si $i \notin L(x)$. Definimos $\mu(x'_i) = \mu(y'_i) = i$ y $\mu(x) = \mu(y) = k$. Entonces G es $\mathbb{L}$-coloreable si y sólo si G' es μ-coloreable. La transformación puede realizarse en tiempo polinomial, completando la demostración. □

El problema de k-coloreo es trivialmente polinomial (lineal) para grafos bipartitos (son todos 2-coloreables), con lo cual μ-coloreo es "más difícil" que k-coloreo, salvo que P=NP.

Capítulo 4

Conclusiones y trabajo futuro

El objetivo de esta tesis era estudiar el problema de μ-coloreo, y tratar de definir y estudiar la noción de perfección aplicada a esta nueva variante del problema de coloreo.

Una duda que surgió al estudiar el problema desde el punto de vista algorítmico fue la siguiente: a simple vista el problema de μ-coloreo es más general que k-coloreo y es un caso particular de coloreo por listas, pero, en términos de complejidad, ¿está realmente entre ambos o es equivalente a alguno de ellos?

Con respecto al concepto de perfección aplicado al μ-coloreo, definimos los grafos M-perfectos como aquellos tales que para toda función μ, el grafo es μ-coloreable si y sólo si todas sus cliques lo son. Los resultados principales en este sentido fueron la equivalencia entre grafos M-perfectos y cografos (grafos P_4-free) y la existencia de un algoritmo polinomial $O(n \log n + m)$ de μ-coloreo para esta clase de grafos.

Este resultado es bastante curioso, y nos hace pensar que hay una relación intrínseca entre la polinomialidad del problema de coloreo y la definición de perfección. Como tercer ejemplo, consideremos la definición de perfección aplicada al coloreo con listas: un grafo es L-perfecto si para toda familia de listas $\mathbb{L}$, el grafo es $\mathbb{L}$-coloreable si y sólo si todas sus cliques lo son. Es fácil ver que si tenemos un $P_3 = v_1v_2v_3$ y asignamos a v_1 la lista $\{1\}$, a v_2 la lista $\{1,2\}$, y a v_3 la lista $\{2\}$, entonces ambas cliques son coloreables pero el grafo completo no lo es. Por lo tanto, resulta que los únicos grafos L-perfectos son los P_3-free, o sea los grafos tales que cada componente conexa es completa, conocidos como grafos *cluster*. Para estos grafos el problema de coloreo con listas es polinomial, ya que se reduce fácilmente a hallar un matching máximo en un grafo bipartito.

Con respecto a la pregunta sobre la complejidad del problema de μ-coloreo con respecto a la de sus parientes más conocidos, la existencia de un algoritmo polinomial de μ-coloreo para cografos implica que μ-coloreo es más fácil que coloreo con listas, salvo que P=NP, ya que coloreo por listas es NP-completo para cografos. Cerrando la pregunta, demostramos que μ-coloreo es NP-completo en grafos bipartitos, para los cuales k-coloreo es trivialmente polinomial, por lo tanto eso implica que μ-coloreo es más difícil que k-coloreo, salvo que P=NP.

Por último, para el caso de grafos bipartitos, árboles y cografos, hallamos cotas superiores para la máxima cantidad de colores que puede requerir μ-colorear un grafo. Para árboles la cota es logarítmica con respecto a la cantidad de vértices, para grafos bipartitos es lineal con respecto a la cantidad de vértices y para cografos coincide con el número cromático del grafo.

Como trabajo futuro, sería interesante estudiar el problema desde el punto de vista algorítmico para otras clases de grafos, en particular para grafos de intervalos y arco-circulares, dada su aplicación a problemas de asignación temporal (o cíclicamente temporal) de recursos a tareas (por ejemplo, el problema de asignación de aulas a materias).

También se podrían estudiar las cotas superiores para la máxima cantidad de colores que puede requerir μ-colorear un grafo (en algún sentido, una definición análoga a la del número cromático) para otras clases de grafos además de las abordadas en esta tesis. Sería interesante también estudiar las propiedades que deben cumplir una función μ y un grafo para que éste sea μ-coloreable, y buscar algoritmos polinomiales para resolver el problema en ciertas clases de grafos imponiendo condiciones a la función μ.

Quedan también por estudiarse algoritmos heurísticos o aproximados de μ-coloreo para las clases de grafos para las cuales no se conoce un algoritmo polinomial, en particular para aquellas en las que resulta NP-completo.

Así como se definieron los grafos M-perfectos a partir de los perfectos usando la analogía entre coloreo y μ-coloreo, se podría extender esta analogía a otras definiciones o propiedades del coloreo, tales como coloreo de aristas (edge coloring), o extensión de precoloreo (precoloring extension). También se puede hacer esta analogía con problemas de coloreo con listas, como por ejemplo elegibilidad (choosability).

Por último, los problemas de μ-coloreo y extensión de precoloreo parecerían estar emparentados de alguna forma, ya que ambos son polinomiales en Cografos [17] y NP en grafos Bipartitos[16]. Extensión de precoloreo NP para grafos de intervalos[2], lo que nos lleva a suponer que μ coloreo también lo es.

Bibliografía

[1] C. Berge, Les problemes de colorations en théorie des graphes, *Publications de l'Institut de Statistique de l'Université de Paris* **9** (1960), 123–160.

[2] M. Biro, M. Hujter, and Z. Tuza, Precoloring extension. I. Interval graphs, *Discrete Mathematics* **100**(1–3) (1992), 267–279.

[3] F. Bonomo and M. Cecowski, Between coloring and list-coloring: μ-coloring, *Electronic Notes in Discrete Mathematics* (2004), to appear.

[4] K. Booth and G.S. Lueker, Linear algorithms to recognize interval graphs and test for the consecutive ones property, *Proceedings of the 7th Annual ACM Symposium on Theory of Computing*, Las Vegas, 1975, pp. 255–265.

[5] A. Brandstädt, V. Le, and J. Spinrad, *Graph Classes: A Survey*, SIAM, Philadelphia, 1999.

[6] M. Chudnovsky, G. Cornuéjols, X. Liu, P. Seymour, and K. Vušković, Recognizing Berge Graphs, *Combinatorica*, to appear.

[7] M. Chudnovsky, N. Robertson, P. Seymour, and R. Thomas, The Strong Perfect Graph Theorem, *Annals of Mathematics*, to appear.

[8] V. Chvátal, Perfectly ordered graphs, *Annals of Discrete Mathematics* **21** (1984), 63–65.

[9] D. Corneil, Y. Perl, and L. Stewart, Cographs: recognition, applications and algorithms, *Congressus Numerantium* **43** (1984), 249–258.

[10] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman and Company, San Francisco, 1979.

[11] M. Garey, D. Johnson, G. Miller, and C. Papadimitriou, The complexity of coloring circular arcs and chords, *SIAM Journal on Algebraic and Discrete Methods* **1** (1980), 216–227.

[12] M. Garey, D. Johnson, and L. Stockmeyer, Some simplified NP-complete graph problems, *Theoretical Computer Science* **1**(3) (1976), 237–267.

[13] N. Garg, M. Papatriantafilou, and P. Tsigas, Distributed List Coloring: How to Dynamically Allocate Frequencies to Mobile Base Stations, *Proceedings of the 8th IEEE Symposium on Parallel and Distributed Processing*, 1996, p. 18.

[14] M. Grötschel, L. Lovász, and A. Schrijver, The ellipsoid method and its consequences in combinatorial optimization, *Combinatorica* **1** (1981), 169–197.

[15] V. Guruswami and C. Pandu Rangan, Algorithmic aspects of clique-transversal and clique-independent sets, *Discrete Applied Mathematics* **100** (2000), 183–202.

[16] M. Hujter and Z. Tuza, Precoloring extension. II. Graph classes related to bipartite graphs, *Acta Mathematica Universitatis Comenianae* **62**(1) (1993), 1–11.

[17] M. Hujter and Z. Tuza, Precoloring extension. III. Clases of perfect graphs. *Combin. Probab. Comput.*, **5**(1) (1996), 35-56,

[18] K. Jansen and P. Scheffler, Generalized coloring for tree-like graphs, *Discrete Applied Mathematics* **75** (1997), 135–155.

[19] R. Karp, Reducibility among combinatorial problems, In: *Complexity of Computer Computations* (R. Miller and J. Thatcher, eds.), Plenum Press, New York, 1972, pp. 85–103.

[20] S. Klein and M. Kouider, On *b*-perfect graphs, *Annals of the XII Latin-Ibero-American Congress on Operations Research*, Havanna, Cuba, October 2004.

[21] J. Kratochvíl and Z. Tuza, Algorithmic complexity of list colorings, *Discrete Applied Mathematics* **50** (1994), 297–302.

[22] J. Lehel and Z. Tuza, Neighborhood perfect graphs, *Discrete Mathematics* **61** (1986), 93–101.

[23] L. Lovász, A characterization of perfect graphs and the perfect graph conjecture, *Journal of Combinatorial Theory. Series B* **132** (1972), 95–98.

[24] Z. Tuza, Graph colorings with local constraints – a survey, *Discussiones Mathematicae. Graph Theory* **17** (1997), 161–228.

[25] X. Zhu, Circular perfect graphs, *Journal of Graph Theory*, to appear.