

# CI-5651 Proyecto #2

## Algoritmo Branch & Bound para el problema de Coloración de Grafos

Guillermo Palma

Alejandro Flores

### 1. Introducción

El problema de Coloración de Grafos es uno de los problemas combinatorios más estudiados en la literatura. Tiene aplicaciones prácticas en programación de horarios, planificación de actividades y asignación de recursos. El objetivo de este proyecto es la implementación y el estudio de un algoritmo Branch & Bound, para encontrar soluciones exactas al problema de coloración de grafos.

### 2. Planteamiento del problema

El problema de Coloración de Grafos puede definirse informalmente de la siguiente manera: dado un grafo  $G = (V, E)$  no dirigido se quiere colorear los nodos del grafo con la menor cantidad de colores posibles, tal que dos nodos adyacentes no tengan el mismo color. Este valor se conoce como número cromático, denotado como  $\chi(G)$ . Para una definición más formal del problema, debe leer las secciones 1.1 y 1.2 de [6].

Se quiere que se implemente el algoritmo corregido de Brélaz [7] para el problema de Coloración de Grafos. Se trata de un algoritmo que sigue una estrategia Branch & Bound, y asegura la obtención de soluciones óptimas. Este algoritmo es una modificación sobre el algoritmo de Brélaz [4], que corrige dos errores que pueden evitar la optimalidad del algoritmo. A su vez, esta es una modificación sobre el algoritmo propuesto por Brown [5] para el mismo problema.

Las soluciones serán evaluadas usando las instancias publicadas en [3] y [1] que se encuentran en el formato estándar DIMACS. Las soluciones óptimas del problema de coloración sobre estos grafos pueden encontrarse en [2], junto con el orden de magnitud del tiempo necesario para calcular las soluciones óptimas (segundos, minutos, horas, etc.). Las soluciones deben

ser lo más eficiente posible, pues el tiempo de ejecución será un factor de evaluación.

### 3. Entrega

La fecha **máxima** de entrega es el 6 de febrero de 2015. Debe entregarse un archivo .zip con el código fuente y el informe, enviado por correo electrónico a `alejandroflores.af@gmail.com`.

#### 3.1. Implementación

- El algoritmo debe implementarse en alguno de los siguientes lenguajes de programación: C, C++ o JAVA. Incluir un Makefile.
- Deben usarse **solo** librerías estándar.
- La entrada debe leerse desde un archivo (indicado como único argumento de la llamada al programa) y la salida debe imprimirse en la salida estándar. El programa debe ejecutarse desde la terminal, usando la siguiente llamada:  
    `> ./p2 instancia`  
    `> java p2 instancia`
- El programa debe imprimir (en una única línea y separados por un espacio), el límite superior inicial (calculado por una heurística) y el número cromático del grafo, *i.e.*  $\chi(G)$ .

#### 3.2. Informe

Junto con el código, debe incluirse un informe con la siguiente estructura:

- Portada
- Diseño y Detalles de Implementación:
  1. Descripción del modelo utilizado para representar el problema.
  2. Estructuras de datos y algoritmos involucrados en la aplicación. Justificación de cada uno de ellos.
  3. Indicación de los problemas encontrados y la manera en que se resolvieron.
- Conclusiones

## Referencias

- [1] Graph Coloring Benchmark Instances. <http://www.cs.hbg.psu.edu/txn131/graphcoloring.html>.
- [2] Graph Coloring Benchmarks. <https://sites.google.com/site/graphcoloring/vertex-coloring>.
- [3] Graph Coloring Instances. <http://mat.gsia.cmu.edu/COLOR/instances.html>.
- [4] Daniel Brélaz. New methods to color the vertices of a graph. *Communications of the ACM*, 22(4):251–256, 1979.
- [5] J Randall Brown. Chromatic scheduling and the chromatic number problem. *Management Science*, 19(4-part-1):456–463, 1972.
- [6] Assefaw Hadish Gebremedhin. *Parallel graph coloring*. PhD thesis, Citeseer, 1999.
- [7] Jürgen Peemöller. A correction to brelaz’s modification of brown’s coloring algorithm. *Commun. ACM*, 26(8):595–597, August 1983.