Coloreo de Grafos

Algoritmos y Estructuras de Datos III

Coloreo de nodos

Definiciones:

- Un **coloreo (válido) de los nodos** de un grafo $G = (V, X)$ es una asignación $f : V \to C$, tal que $f(v) \neq f(u)$ $\forall (u, v) \in E$.
- Los elementos de C son llamados **colores**. Muchas veces los colores son enteros positivos.
- Para todo entero positvo k, un k-**coloreo** de G es un coloreo de los nodos de G que usa exactamente k colores.
- Un grafo G se dice k-**coloreable** si existe un k-coloreo de G.
- El **número cromático** de G, $\chi(G)$, es el menor número de colores necesarios para colorear los nodos de G.
- Un grafo G se dice k-**cromático** si $\chi(G) = k$.

Coloreo de nodos

Ejemplos:

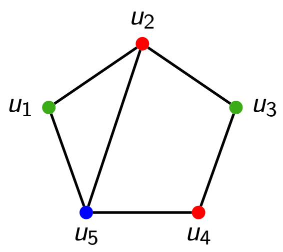
- $\chi(K_n) = n$.
- Si G es un grafo bipartito con $m > 0$, entonces $\chi(G) = 2$.
- Si H_{2k} es un circuito simple par, entonces $\chi(H_{2k}) = 2$.
- Si H_{2k+1} es un circuito simple impar, entonces $\chi(H_{2k+1}) = 3$.
- Si T es un árbol con $n > 1$, entonces $\chi(T) = 2$.

Cotas para χ

Proposición: Si H es un subgrafo de G entonces $\chi(H) \leq \chi(G)$.

Definición: Una clique en un grafo es un subgrafo completo maximal. El número clique $\omega(G)$ de un grafo es el número de nodos de una clique máxima de G.

Proposición: Para cualquier grafo G, $\chi(G) \geq \omega(G)$.



$\chi(G) \leq 3$

$\omega(G) \geq 3$

$\chi(G) = 3$

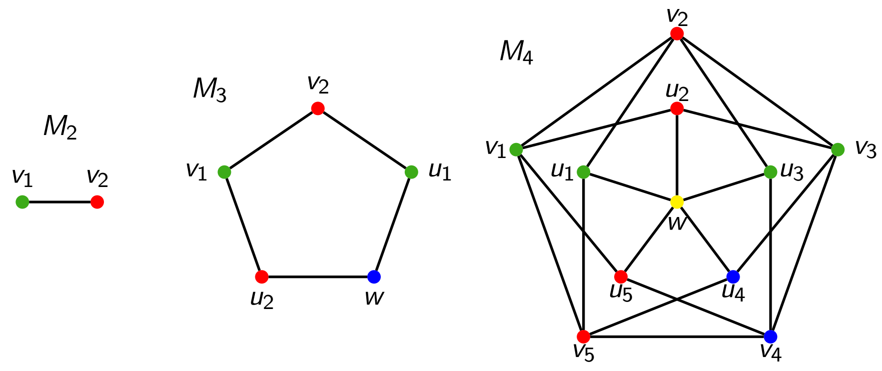
¿Es buena esta cota?

Grafos de Mycielski

Definición (por inducción):

1. $M_1 = K_1$
2. $M_2 = K_2$
3. Para $i \geq 2$, M_{i+1} se construye a partir de M_i de la siguiente forma:
   - Si M_i tiene p nodos, $v_1, \ldots, v_p$, M_{i+1} tendrá $2p+1$ nodos, $v_1, \ldots, v_p, u_1, \ldots, u_p, w$, donde u_i es copia de v_i.
   - El conjunto de aristas de M_{i+1} tendrá todas las aristas de M_i, las aristas uniendo u_i con los vecinos de v_i en M_i y las aristas uniendo w con cada u_i.

Grafos de Mycielski



¿Cuál es el número cromático de M_i? $\qquad \chi(M_i) = i$

¿Cuál es la clique máxima de M_i? $\qquad \omega(M_i) = 2$

Cotas para χ

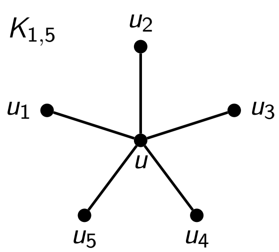
Proposición: Si $\Delta(G)$ es el grado máximo de G entonces

$$\chi(G) \leq \Delta(G) + 1.$$

Teorema (Brooks): Sea G un grafo conexo que no es un circuito impar ni un grafo completo. Entonces

$$\chi(G) \leq \Delta(G).$$

¿Son buenas estas cotas?



$\chi(K_{1,n}) = 2$

$\Delta(K_{1,n}) = n$

Problema de los cuatro colores

Teorema de los 4 colores (Appel, Haken, 1976): Si G es un grafo planar, entonces

$$\chi(G) \leq 4.$$

Teorema (Heawood, 1890): Si G es un grafo planar, entonces

$$\chi(G) \leq 5.$$

Algoritmos para coloreo de grafos

- Problema *difícil*, computacionalmente no resuelto.
- No se conocen algoritmos polinomiales para calcular $\chi(G)$ dado un grafo general G.
- Existen muchos enfoques algorítmicos para este problema:
  - Heurísticas y metaheurísticas.
  - Algoritmos basados en backtracking (por ejemplo: DSATUR, Brelaz, 1979).
  - Algoritmos exactos basados en programación lineal entera.

Algoritmo (heurística) secuencial (S)

Dado un orden $v_1, \ldots, v_n$ de V, asignar en el paso i el menor color posible en $\mathbb{N}$ a v_i, para $i = 1, \ldots, n$.

Entrada: Un grafo G con un orden en los nodos $v_1, \ldots, v_n$.

$f(v_1) := 1$

para $i = 2, 3, \ldots, n$ **hacer**

$f(v_i) = \text{mín}\{h \ / \ h \geq 1 \ \texttt{y}$

$f(v_j) \neq h \ \forall (v_j, v_i) \in E, 1 \leq j \leq i-1\}$

retornar `coloreo definido por` f

Algoritmo secuencial (S)

Definimos

$$u_S(G, v_1, v_2, \ldots, v_n) = \max_{1 \leq i \leq n} \text{mín}\{i, d(v_i) + 1\}.$$

Proposición: Si $\chi_S(G)$ es el número de colores usado por el algoritmo secuencial para colorear G cuando los nodos son considerados en el orden $v_1, \ldots, v_n$, entonces

$$\chi(G) \leq \chi_S(G) \leq u_S(G, v_1, v_2, \ldots, v_n).$$

¿Importa el orden en que se colorean los nodos con el algoritmo secuencial?

Algoritmo secuencial (LFS)

Orden Largest First (LF): los nodos son ordenados de mayor grado a menor grado, $d(u_1) \geq d(u_2) \geq \ldots \geq d(u_n)$.

Proposición: Si $u_{LF}(G) = u_S(G, u_1, u_2, \ldots, u_n)$ donde $u_1, u_2, \ldots, u_n$ están ordenados según LF. Entonces

$$u_{LF}(G) \leq \text{mín}\, u_S(G, v_1, v_2, \ldots, v_n)$$

donde el mínimo está tomado sobre todos los ordenes posibles, $v_1, \ldots, v_n$.

¿Esto implica que siempre el algoritmo secuencial da un resultado mejor si se usa LF?

Algoritmo secuencial

Otra cota (mejor) para el número de colores usados por el algoritmo secuencial es:

$$u'_S(G, v_1, v_2, \ldots, v_n) = 1 + \max_{1 \leq i \leq n} \{d_{G_i}(v_i)\}$$

donde $d_{G_i}(v_i)$ es el grado del nodo v_i en el grafo inducido por $v_1, v_2, \ldots, v_i$.

Esto sugiere el siguiente orden.

Algoritmo secuencial (SLS)

Orden Smallest Last (SL):

1. poner como v_n el nodo de mínimo grado de G.
2. para $i = n-1, \ldots, 1$ poner como v_i el nodo de grado mínimo en el subgrafo de G inducido por $V \setminus \{v_n, v_{n-1}, \ldots, v_{i+1}\}$.

Definimos

$$u_{SL}(G) = 1 + \max_{1 \leq i \leq n} \min_{1 \leq j \leq i} \{d_{G_i}(v_j)\}$$

donde $d_{G_i}(v_j)$ es el grado del nodo v_j en el grafo inducido por $V \setminus \{v_n, v_{n-1}, \ldots, v_{i+1}\}$.

Algoritmo secuencial - Cotas

Se puede demostrar (ejercicio) que:

- $\chi_{SLS}(G) \leq u_{SL}(G)$.
- $u_{SL}(G) \leq u_{LF}(G)$.
- SLS colorea un grafo planar con 6 colores o menos.

Algoritmo secuencial con intercambio (SI)

- Supongamos que tenemos un coloreo parcial de G, donde los nodos $v_1, \ldots, v_{i-1}$ ya han sido coloredos y es el turno de colorear a v_i. Si todos los colores ya utilizados están en la vecindad de v_i, será necesario utilizar un nuevo color.
- Si existen p y q dos colores utilizados en el coloreo parcial, tal que en todas las componenetes conexas de H_{pq} los nodos adyacentes a v_i tienen el mismo color, podemos intercambiar los colores p y q en las componentes de H_{pq} con nodos adyacentes a v_i con color p.
- De esta manera, obtendremos un coloreo parcial de G con el color p no utilizado en la vecindad de v_i.
- Este procedimiento se llama p, q-intercambio.

Algoritmo secuencial con intercambio (SI)

$f(v_1) := 1$, $k := 1$
para $i = 2, 3, \ldots, n$ **hacer**
  $g := \text{mín}\{h/h \geq 1$ y
    $f(v_j) \neq h \;\; \forall (v_j, v_i) \in E, \;\; 1 \leq j \leq i-1\}$
  si $g \leq k$ **entonces**
    $f(v_i) := g$
  sino
    si `existen` $1 \leq p < q \leq k$, `tales que`
     `un` p, q`-intercambio` *libera* p **entonces**
      `realizar el` p, q`-intercambio`
      $f(v_i) := p$
    sino
      $f(v_i) := g$, $k := k + 1$

Algoritmo secuencial con intercambio (SI)

- No siempre es mejor el algoritmo SI que el algoritmo S.
- Se puede demostrar que:
  - SI colorea un grafo bipartito con 2 colores (ejercicio).
  - SI con el ordenamiento SL colorea un grafo planar con 5 colores como máximo.

Algoritmo secuencial con bracktracking (exacto)

- Se hace una búsqueda exhaustiva. En el árbol de enumeración, cada nodo de nivel i corresponde a un coloreo de $v_1, \ldots, v_{i-1}$.
- Se avanza por las ramas coloreando los siguientes vértices hasta que ocurre alguna de las siguientes situaciones:
  1. Se llegó a un vértice sin colores disponibles: se hace backtracking a partir de v_{i-1} (nodo anterior).
  2. Se coloreó v_n: se encontró un nuevo coloreo del grafo, actualizamos el mejor número de colores q y se continúa con backtracking.
  3. Se llega a un coloreo parcial con más de q colores: este coloreo seguro no será mejor que el actual, se hace backtracking.

Algoritmo secuencial con bracktracking con poda (exacto)

- U_i = conjunto de colores posibles para el nodo v_i, una vez que han sido coloreados $v_1, v_2 \ldots\ldots v_{i-1}$.
- Si l_{i-1} es el máximo color usado para $v_1, \ldots, v_{i-1}$ y sólo buscamos coloreos óptimos, evitando coloreos equivalentes, $\forall j \in U_i$ se verifica que:
  - j no es color asignado a un vecino de v_i ya coloreado
  - $j \leq d(v_i) + 1$
  - $1 \leq j \leq l_{i-1} + 1$
  - si ya se encontró un coloreo del grafo con q colores entonces $j \leq q - 1$

Algoritmo secuencial con bracktracking (exacto)

- q: cantidad de colores usados en la mejor solución encontrada hasta el momento.
- k: nodo siendo considerado.
- l: cantidad de colores utilizados en la solución parcial actual.
- l_k: l para el nodo v_k.
- $cotaInf$: cota inferior para el número cromático del grafo.

Algoritmo secuencial con bracktracking (exacto)

$f(v_1) := 1,\ q := n+1,\ k := 1,\ l := 1$
avanzar := `VERDADERO`
repetir
  si *avanzar*
    $k := k+1,\ l_k := l,$ `determinar` U_k
  si $U_k = \emptyset$
    avanzar := `FALSO`, $k := k-1,\ l := l_k$
  sino
    $j := \text{mín}\, U_k,\ U_k := U_k \setminus \{j\},\ f(v_k) := j$
    si $j > l$ **entonces** $l := l+1$
    si $k < n$ **entonces** *avanzar* := `VERDADERO`
    sino
      `almacenar la nueva solución`
      `encontrar el menor` i `tal que` $f(v_i) = l$
      `borrar` $l, l+1, \ldots, q-1$ `de` $U_1, \ldots, U_{i-1}$
      $q := l,\ l := q-1,\ k := i-1$
      avanzar := `FALSO`
hasta $k = 1$ **o** $q = cotaInf$

Coloreo de aristas

Definiciones:

- Un coloreo válido de las aristas de un grafo G es un asignación de colores a las mismas en la cual dos aristas que tienen un nodo en común no tengan el mismo color.
- El índice cromático $\chi'(G)$ de un grafo G es el menor número de colores con que se pueden colorear las aristas de un grafo.

Teorema de Vizing: Para todo grafo G se verifica que

$$\Delta(G) \leq \chi'(G) \leq \Delta(G) + 1.$$