Un algoritmo exacto para el Problema de Coloreo Equitativo en Grafos

Isabel Méndez-Díaz[2] Graciela Nasini[1,3] Daniel Severin[1,3]

[1] FCEIA, UNR, {nasini, daniel}@fceia.unr.edu.ar

[2] FCEyN, UBA, imendez@dc.uba.ar

[3] CONICET, Argentina

IV Congreso Latinoamericanos de Matemáticos
Córdoba, Agosto de 2012

Resumen

Definiciones básicas

Sea $G = (V, E)$ $\qquad V = \{1, \ldots, n\}$

Coloreo Clásico

k-coloreo = partición de V en k conjuntos estables

$C_1, C_2, \ldots, C_k \longleftarrow$ clases de color

Coloreo Equitativo

k-eqcol = k-coloreo tal que:

- $||C_i| - |C_j|| \leq 1 \qquad \forall\, i, j = 1, \ldots, k$

o equivalentemente:

- $\lfloor n/k \rfloor \leq |C_j| \leq \lceil n/k \rceil \qquad \forall\, j = 1, \ldots, k$

Número cromático equitativo

$\chi_{eq}(G) = \min\{k : G \text{ admite un } k\text{-eqcol}\}$

- PCEG consiste en hallar $\chi_{eq}(G) \longleftarrow$ NP-Hard

Definiciones básicas

Sea $G = (V, E)$ $\qquad V = \{1, \ldots, n\}$

Coloreo Clásico

k-coloreo $=$ partición de V en k conjuntos estables

$C_1, C_2, \ldots, C_k \longleftarrow$ clases de color

Coloreo Equitativo

k-eqcol $=$ k-coloreo tal que:

- $||C_i| - |C_j|| \leq 1 \qquad \forall\, i, j = 1, \ldots, k$

o equivalentemente:

- $\lfloor n/k \rfloor \leq |C_j| \leq \lceil n/k \rceil \qquad \forall\, j = 1, \ldots, k$

Número cromático equitativo

$\chi_{eq}(G) = \min\{k : G \text{ admite un } k\text{-eqcol}\}$

- PCEG consiste en hallar $\chi_{eq}(G) \longleftarrow$ NP-Hard

Definiciones básicas

Sea $G = (V, E)$ $\qquad V = \{1, \dots, n\}$

Coloreo Clásico

k-coloreo $=$ partición de V en k conjuntos estables

$C_1, C_2, \dots, C_k \longleftarrow$ clases de color

Coloreo Equitativo

k-eqcol $=$ k-coloreo tal que:

- $||C_i| - |C_j|| \leq 1 \qquad \forall\ i, j = 1, \dots, k$

o equivalentemente:

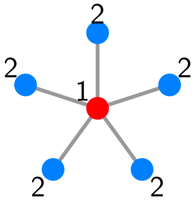
- $\lfloor n/k \rfloor \leq |C_j| \leq \lceil n/k \rceil \qquad \forall\ j = 1, \dots, k$

Número cromático equitativo

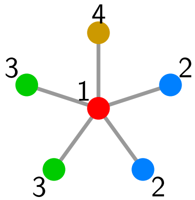
$\chi_{eq}(G) = \min\{k : G \text{ admite un } k\text{-eqcol}\}$

- PCEG consiste en hallar $\chi_{eq}(G) \longleftarrow$ NP-Hard

Ejemplo



$\chi(K_{1,5}) = 2$



$\chi_{eq}(K_{1,5}) = 4$

- En todo coloreo, el vértice central conforma una clase de color de tamaño 1
- En col. equitativo, las otras clases de color tienen tamaño a lo sumo 2

Ejemplo

$\chi(K_{1,5}) = 2$ $\qquad$ $\chi_{eq}(K_{1,5}) = 4$

- En todo coloreo, el vértice central conforma una clase de color de tamaño 1
- En col. equitativo, las otras clases de color tienen tamaño a lo sumo 2

Breve historia del problema

¿Existe algún interés práctico en resolver el PCEG?

- Asignación de aulas a asignaturas [Kubale & Furmańczyk 2005]
- Recolección de residuos [Tucker 1973]
- Esquemas de distribución paralela de datos en memoria [Das, Finocchi & Petreschi 2006]
- Derivación de cotas Chernoff-Hoeffding en suma de variables aleatorias [Pemmaraju 2001]

Breve historia del problema

¿Existe algún interés práctico en resolver el PCEG?

- Asignación de aulas a asignaturas [Kubale & Furmańczyk 2005]
- Recolección de residuos [Tucker 1973]
- Esquemas de distribución paralela de datos en memoria [Das, Finocchi & Petreschi 2006]
- Derivación de cotas Chernoff-Hoeffding en suma de variables aleatorias [Pemmaraju 2001]

Breve historia del problema

- Definición y primeros resultados [Meyer (1973)]
- Complejidad del PCEG [Kubale, Furmanczyk (2005)]
- Modelo IP model y C&B [Méndez-Díaz, Nasini, S.- (Alio/Euro 2008)]
- B&C-LF_2 basado en modelo de *representantes asimétricos* [Bahiense, Frota, Maculan, Noronha, Ribeiro (LAGOS 2009)]
- Búsqueda Tabu y B&C basado en desigualdades clique y nueva estr. de branching [Méndez-Díaz, Nasini, S.- (INFORMS 2010)]
- Nuevas desigualdades válidas para el PCEG [Méndez-Díaz, Nasini, S.- (LAGOS 2011)]
- B&C que incorpora nuevas desig. válidas, heur. primales, enum. implícita, etc - Tesis Doctoral S.- (2012)

Motivación

Si ya existen algoritmos poderosos para resolver el PCEG, ¿Por qué proponer un algoritmo de enumeración tipo DSATUR?

- Sencillez, fácil de implementar, no requiere de motores de optimización
- Al limitarle su tiempo, puede ser útil para hallar una cota superior de calidad
- Puede ser incorporado a un B&C para resolver subproblemas cuando el B&C lo considere necesario

Recientemente se ha comprobado que una variante de DSATUR resultó ser competitiva frente al mejor B&C hasta el momento, en instancias aleatorias de ciertos tamaños [San Segundo 2012]

☞ los algoritmos tipo DSATUR aún tienen vigencia ☺

Motivación

Si ya existen algoritmos poderosos para resolver el PCEG, ¿Por qué proponer un algoritmo de enumeración tipo DSATUR?

- Sencillez, fácil de implementar, no requiere de motores de optimización
- Al limitarle su tiempo, puede ser útil para hallar una cota superior de calidad
- Puede ser incorporado a un B&C para resolver subproblemas cuando el B&C lo considere necesario

Recientemente se ha comprobado que una variante de DSATUR resultó ser competitiva frente al mejor B&C hasta el momento, en instancias aleatorias de ciertos tamaños [San Segundo 2012]

☞ los algoritmos tipo DSATUR aún tienen vigencia ☺

Motivación

Si ya existen algoritmos poderosos para resolver el PCEG, ¿Por qué proponer un algoritmo de enumeración tipo DSATUR?

- Sencillez, fácil de implementar, no requiere de motores de optimización
- Al limitarle su tiempo, puede ser útil para hallar una cota superior de calidad
- Puede ser incorporado a un B&C para resolver subproblemas cuando el B&C lo considere necesario

Recientemente se ha comprobado que una variante de DSATUR resultó ser competitiva frente al mejor B&C hasta el momento, en instancias aleatorias de ciertos tamaños [San Segundo 2012]

☞ los algoritmos tipo DSATUR aún tienen vigencia ☺

Motivación

Si ya existen algoritmos poderosos para resolver el PCEG, ¿Por qué proponer un algoritmo de enumeración tipo DSATUR?

- Sencillez, fácil de implementar, no requiere de motores de optimización
- Al limitarle su tiempo, puede ser útil para hallar una cota superior de calidad
- Puede ser incorporado a un B&C para resolver subproblemas cuando el B&C lo considere necesario

Recientemente se ha comprobado que una variante de DSATUR resultó ser competitiva frente al mejor B&C hasta el momento, en instancias aleatorias de ciertos tamaños [San Segundo 2012]

☞ los algoritmos tipo DSATUR aún tienen vigencia ☺

Motivación

Si ya existen algoritmos poderosos para resolver el PCEG, ¿Por qué proponer un algoritmo de enumeración tipo DSATUR?

- Sencillez, fácil de implementar, no requiere de motores de optimización
- Al limitarle su tiempo, puede ser útil para hallar una cota superior de calidad
- Puede ser incorporado a un B&C para resolver subproblemas cuando el B&C lo considere necesario

Recientemente se ha comprobado que una variante de DSATUR resultó ser competitiva frente al mejor B&C hasta el momento, en instancias aleatorias de ciertos tamaños [San Segundo 2012]

☞ los algoritmos tipo DSATUR aún tienen vigencia ☺

Motivación

Si ya existen algoritmos poderosos para resolver el PCEG, ¿Por qué proponer un algoritmo de enumeración tipo DSATUR?

- Sencillez, fácil de implementar, no requiere de motores de optimización
- Al limitarle su tiempo, puede ser útil para hallar una cota superior de calidad
- Puede ser incorporado a un B&C para resolver subproblemas cuando el B&C lo considere necesario

Recientemente se ha comprobado que una variante de DSATUR resultó ser competitiva frente al mejor B&C hasta el momento, en instancias aleatorias de ciertos tamaños [San Segundo 2012]

☞ los algoritmos tipo DSATUR aún tienen vigencia ☺

Breve repaso de DSATUR y sus variantes

Más definiciones:

Coloreo Parcial

Π es un k-coloreo-parcial si es una partición de V en $k+1$ conjuntos C_1, C_2, ..., C_k, U, de los cuales los primeros k son estables, y U es el conjunto de *vertices no coloreados*

Grado de saturación

Dados Π k-coloreo-parcial y $u \in U$, $\rho(u)$ es el grado de saturación de u definido como la cantidad de colores distintos asignados a los vértices adyacentes a u

$$\rho(u) = |\{\Pi(v) : v \in N(u) \backslash U\}|$$

Breve repaso de DSATUR y sus variantes

Más definiciones:

Coloreo Parcial

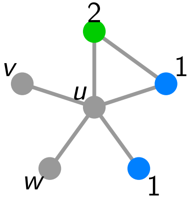
Π es un k-coloreo-parcial si es una partición de V en $k+1$ conjuntos C_1, C_2, ..., C_k, U, de los cuales los primeros k son estables, y U es el conjunto de *vertices no coloreados*

Grado de saturación

Dados Π k-coloreo-parcial y $u \in U$, $\rho(u)$ es el grado de saturación de u definido como la cantidad de colores distintos asignados a los vértices adyacentes a u

$$\rho(u) = |\{\Pi(v) : v \in N(u) \backslash U\}|$$

Breve repaso de DSATUR y sus variantes



$k = 2 \qquad U = \{u, v, w\} \qquad \rho(u) = 2 \qquad \rho(v) = \rho(w) = 0$

Elementos de DSATUR:

- Q = clique maximal de G
- $\overline{c}$ = coloreo inicial de G
- UB = cota superior de $\chi(G)$
  - inicialmente es cant. de colores de $\overline{c}$
- Π k-coloreo parcial
  - $k < UB$
  - vértices de Q siempre son pintados de colores $1, 2, \ldots, |Q|$
- $F(u)$ = colores disponibles para $u \in U$ en Π
  - $\Pi(v) \notin F(u)$ para cada $v \in N(u) \backslash U$
  - $|F(u) \cap \{1, \ldots, UB - 1\}| \geq 1$

Breve repaso de DSATUR y sus variantes

Elementos de DSATUR:

- Q = clique maximal de G
- $\overline{c}$ = coloreo inicial de G
- UB = cota superior de $\chi(G)$
  - inicialmente es cant. de colores de $\overline{c}$
- Π k-coloreo parcial
  - $k < UB$
  - vértices de Q siempre son pintados de colores $1, 2, \ldots, |Q|$
- $F(u)$ = colores disponibles para $u \in U$ en Π
  - $\Pi(v) \notin F(u)$ para cada $v \in N(u) \backslash U$
  - $|F(u) \cap \{1, \ldots, UB-1\}| \geq 1$

Elementos de DSATUR:

- Q = clique maximal de G
- $\overline{c}$ = coloreo inicial de G
- UB = cota superior de $\chi(G)$
  - inicialmente es cant. de colores de $\overline{c}$
- Π k-coloreo parcial
  - $k < UB$
  - vértices de Q siempre son pintados de colores $1, 2, \ldots, |Q|$
- $F(u)$ = colores disponibles para $u \in U$ en Π
  - $\Pi(v) \notin F(u)$ para cada $v \in N(u) \backslash U$
  - $|F(u) \cap \{1, \ldots, UB - 1\}| \geq 1$

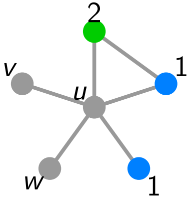
Elementos de DSATUR:

- Q = clique maximal de G
- $\overline{c}$ = coloreo inicial de G
- UB = cota superior de $\chi(G)$
  - inicialmente es cant. de colores de $\overline{c}$
- Π k-coloreo parcial
  - $k < UB$
  - vértices de Q siempre son pintados de colores $1, 2, \ldots, |Q|$
- $F(u)$ = colores disponibles para $u \in U$ en Π
  - $\Pi(v) \notin F(u)$ para cada $v \in N(u) \setminus U$
  - $|F(u) \cap \{1, \ldots, UB - 1\}| \geq 1$

Elementos de DSATUR:

- Q = clique maximal de G
- $\overline{c}$ = coloreo inicial de G
- UB = cota superior de $\chi(G)$
  - inicialmente es cant. de colores de $\overline{c}$
- Π k-coloreo parcial
  - $k < UB$
  - vértices de Q siempre son pintados de colores $1, 2, \ldots, |Q|$
- $F(u)$ = colores disponibles para $u \in U$ en Π
  - $\Pi(v) \notin F(u)$ para cada $v \in N(u) \backslash U$
  - $|F(u) \cap \{1, \ldots, UB - 1\}| \geq 1$

Breve repaso de DSATUR y sus variantes



$k = 2 \quad UB = 4 \quad F(u) = \{3, 4\} \quad F(v) = F(w) = \{1, 2, 3, 4\}$

$\textsc{DSatur}(k, \Pi, F)$:
(UB y $\overline{c}$ variables globales)

- *Paso 1.* Si $U = \varnothing$ entonces Π ya es un k-coloreo: hacer $UB \leftarrow k$, $\overline{c} \leftarrow \Pi$ y salir.
- *Paso 2.* Elegir vértice $u \in U$ de máximo grado de saturación.
- *Paso 3.* Para cada color $1 \leq j \leq UB - 1$ tal que $j \in F(u)$, ejecutar $\textsc{DSatur}(k', \Pi', F')$ donde:
  - $k' \leftarrow \max\{j, k\}$,
  - $C'_j \leftarrow C_j \cup \{u\}$,
  - $C'_r \leftarrow C_r$ for all $r \neq j$,
  - $U' \leftarrow U \backslash \{u\}$,
  - $F'(v) \leftarrow (F(v) \cap \{1, \ldots, UB - 1\}) \backslash \{j\}$ para todo $v \in N(u) \cap U$,
  - $F'(v) \leftarrow F(v) \cap \{1, \ldots, UB - 1\}$ para todo $v \in U \backslash N[u]$.

  La evaluación del subproblema $\textsc{DSatur}(k', \Pi', F')$ sólo es realizada si $F'(v) \neq \varnothing$ para todo $v \in U \backslash \{u\}$.

$\textsc{DSatur}(k, \Pi, F)$:
(UB y $\overline{c}$ variables globales)

- *Paso 1.* Si $U = \varnothing$ entonces Π ya es un k-coloreo: hacer $UB \leftarrow k$, $\overline{c} \leftarrow \Pi$ y salir.
- *Paso 2.* Elegir vértice $u \in U$ de máximo grado de saturación.
- *Paso 3.* Para cada color $1 \leq j \leq UB - 1$ tal que $j \in F(u)$, ejecutar $\textsc{DSatur}(k', \Pi', F')$ donde:
  - $k' \leftarrow \max\{j, k\}$,
  - $C'_j \leftarrow C_j \cup \{u\}$,
  - $C'_r \leftarrow C_r$ for all $r \neq j$,
  - $U' \leftarrow U \backslash \{u\}$,
  - $F'(v) \leftarrow (F(v) \cap \{1, \ldots, UB - 1\}) \backslash \{j\}$ para todo $v \in N(u) \cap U$,
  - $F'(v) \leftarrow F(v) \cap \{1, \ldots, UB - 1\}$ para todo $v \in U \backslash N[u]$.

  La evaluación del subproblema $\textsc{DSatur}(k', \Pi', F')$ sólo es realizada si $F'(v) \neq \varnothing$ para todo $v \in U \backslash \{u\}$.

$\text{DSATUR}(k, \Pi, F)$:
(UB y $\overline{c}$ variables globales)

- *Paso 1.* Si $U = \varnothing$ entonces Π ya es un k-coloreo: hacer $UB \leftarrow k$, $\overline{c} \leftarrow \Pi$ y salir.
- *Paso 2.* Elegir vértice $u \in U$ de máximo grado de saturación.
- *Paso 3.* Para cada color $1 \leq j \leq UB - 1$ tal que $j \in F(u)$, ejecutar $\text{DSATUR}(k', \Pi', F')$ donde:
  - $k' \leftarrow \max\{j, k\}$,
  - $C'_j \leftarrow C_j \cup \{u\}$,
  - $C'_r \leftarrow C_r$ for all $r \neq j$,
  - $U' \leftarrow U \backslash \{u\}$,
  - $F'(v) \leftarrow (F(v) \cap \{1, \ldots, UB - 1\}) \backslash \{j\}$ para todo $v \in N(u) \cap U$,
  - $F'(v) \leftarrow F(v) \cap \{1, \ldots, UB - 1\}$ para todo $v \in U \backslash N[u]$.

  La evaluación del subproblema $\text{DSATUR}(k', \Pi', F')$ sólo es realizada si $F'(v) \neq \varnothing$ para todo $v \in U \backslash \{u\}$.

$\underline{\textsc{DSatur}(k, \Pi, F)}$:
(UB y $\overline{c}$ variables globales)

- *Paso 1.* Si $U = \emptyset$ entonces Π ya es un k-coloreo: hacer $UB \leftarrow k$, $\overline{c} \leftarrow \Pi$ y salir.
- *Paso 2.* Elegir vértice $u \in U$ de máximo grado de saturación.
- *Paso 3.* Para cada color $1 \leq j \leq UB - 1$ tal que $j \in F(u)$, ejecutar $\textsc{DSatur}(k', \Pi', F')$ donde:
  - $k' \leftarrow \max\{j, k\}$,
  - $C'_j \leftarrow C_j \cup \{u\}$,
  - $C'_r \leftarrow C_r$ for all $r \neq j$,
  - $U' \leftarrow U \backslash \{u\}$,
  - $F'(v) \leftarrow (F(v) \cap \{1, \ldots, UB - 1\}) \backslash \{j\}$ para todo $v \in N(u) \cap U$,
  - $F'(v) \leftarrow F(v) \cap \{1, \ldots, UB - 1\}$ para todo $v \in U \backslash N[u]$.

  La evaluación del subproblema $\textsc{DSatur}(k', \Pi', F')$ sólo es realizada si $F'(v) \neq \emptyset$ para todo $v \in U \backslash \{u\}$.

$\underline{\text{DSATUR}(k, \Pi, F)}$:

(UB y $\overline{c}$ variables globales)

- *Paso 1.* Si $U = \emptyset$ entonces Π ya es un k-coloreo: hacer $UB \leftarrow k$, $\overline{c} \leftarrow \Pi$ y salir.
- *Paso 2.* Elegir vértice $u \in U$ de máximo grado de saturación.
- *Paso 3.* Para cada color $1 \leq j \leq UB - 1$ tal que $j \in F(u)$ y $j \leq k + 1$, ejecutar DSATUR(k', Π', F') donde:
  - $k' \leftarrow \max\{j, k\}$,
  - $C'_j \leftarrow C_j \cup \{u\}$,
  - $C'_r \leftarrow C_r$ for all $r \neq j$,
  - $U' \leftarrow U \backslash \{u\}$,
  - $F'(v) \leftarrow (F(v) \cap \{1, \ldots, UB - 1\}) \backslash \{j\}$ para todo $v \in N(u) \cap U$,
  - $F'(v) \leftarrow F(v) \cap \{1, \ldots, UB - 1\}$ para todo $v \in U \backslash N[u]$.

  La evaluación del subproblema DSATUR(k', Π', F') sólo es realizada si $F'(v) \neq \emptyset$ para todo $v \in U \backslash \{u\}$.

En cada nivel se incorpora un nuevo color $(k + 1)$

Breve repaso de DSATUR y sus variantes

¿Qué pasa si dos o más vértices tienen el mismo máximo grado de saturación ρ?

Variantes:

- DSATUR original: se desempata elegiendo el vértice u de mayor grado $\delta(u)$. [Brélaz 1979]
- CELIM: desempate con regla más compleja. [Sewell 1996]
- PASS: otras reglas más complejas. [San Segundo 2012]

¿Qué pasa si dos o más vértices tienen el mismo máximo grado de saturación ρ?

Variantes:

- DSATUR original: se desempata eligiendo el vértice u de mayor grado $\delta(u)$. [Brélaz 1979]
- CELIM: desempate con regla más compleja. [Sewell 1996]
- PASS: otras reglas más complejas. [San Segundo 2012]

Desafíos presentes en el PCEG

Desafíos presentes en el PCEG

- La clique maximal Q no es la única fuente para proveer una cota inferior de $\chi_{eq}(G)$:

$$\chi_{eq}(G) \geq \left\lceil \frac{n+1}{\alpha(V \backslash N[v])+2} \right\rceil \quad \forall\ v \in V$$ [*Lih & Chen* 1994]

Proponemos usar:

$$LB = \max \left\{ |Q|, \left\lceil \frac{n+1}{\theta(V \backslash N[v])+2} \right\rceil \quad \forall\ v \in V \right\}$$

donde $\theta(S) =$ partición en cliques de $G[S]$ (goloso).

☞ 1er. desafío: cómo aprovechar LB

- No todos los coloreos encontrados por DSATUR son equitativos

☞ 2do. desafío: cómo podar tempranamente subproblemas que no conduzcan a coloreos equitativos

Desafíos presentes en el PCEG

- La clique maximal Q no es la única fuente para proveer una cota inferior de $\chi_{eq}(G)$:

$$\chi_{eq}(G) \geq \left\lceil \frac{n+1}{\alpha(V \backslash N[v]) + 2} \right\rceil \ \forall\ v \in V$$ [*Lih & Chen* 1994]

Proponemos usar:

$$LB = \max\left\{ |Q|, \left\lceil \frac{n+1}{\theta(V \backslash N[v]) + 2} \right\rceil \ \forall\ v \in V \right\}$$

donde $\theta(S) =$ partición en cliques de $G[S]$ (goloso).

☞ 1er. desafío: cómo aprovechar LB

- No todos los coloreos encontrados por DSATUR son equitativos

☞ 2do. desafío: cómo podar tempranamente subproblemas que no conduzcan a coloreos equitativos

Desafíos presentes en el PCEG

- La clique maximal Q no es la única fuente para proveer una cota inferior de $\chi_{eq}(G)$:

$$\chi_{eq}(G) \geq \left\lceil \frac{n+1}{\alpha(V \backslash N[v])+2} \right\rceil \quad \forall\ v \in V$$ [*Lih & Chen* 1994]

Proponemos usar:

$$LB = \max\left\{ |Q|, \left\lceil \frac{n+1}{\theta(V \backslash N[v])+2} \right\rceil \quad \forall\ v \in V \right\}$$

donde $\theta(S) =$ partición en cliques de $G[S]$ (goloso).

☞ 1er. desafío: cómo aprovechar LB

- No todos los coloreos encontrados por DSATUR son equitativos

☞ 2do. desafío: cómo podar tempranamente subproblemas que no conduzcan a coloreos equitativos

Desafíos presentes en el PCEG

- La clique maximal Q no es la única fuente para proveer una cota inferior de $\chi_{eq}(G)$:

$$\chi_{eq}(G) \geq \left\lceil \frac{n+1}{\alpha(V \backslash N[v]) + 2} \right\rceil \quad \forall\, v \in V \quad [\textit{Lih \& Chen } 1994]$$

  Proponemos usar:

$$LB = \max\left\{ |Q|, \left\lceil \frac{n+1}{\theta(V \backslash N[v]) + 2} \right\rceil \;\; \forall\, v \in V \right\}$$

  donde $\theta(S) =$ partición en cliques de $G[S]$ (goloso).

  ☞ 1er. desafío: cómo aprovechar LB

- No todos los coloreos encontrados por DSATUR son equitativos

  ☞ 2do. desafío: cómo podar tempranamente subproblemas que no conduzcan a coloreos equitativos

Desafíos presentes en el PCEG

- La clique maximal Q no es la única fuente para proveer una cota inferior de $\chi_{eq}(G)$:

$$\chi_{eq}(G) \geq \left\lceil \frac{n+1}{\alpha(V \setminus N[v]) + 2} \right\rceil \quad \forall\ v \in V \quad [\textit{Lih \& Chen } 1994]$$

  Proponemos usar:

$$LB = \text{máx}\left\{ |Q|, \left\lceil \frac{n+1}{\theta(V \setminus N[v]) + 2} \right\rceil \ \ \forall\ v \in V \right\}$$

  donde $\theta(S) =$ partición en cliques de $G[S]$ (goloso).

  ☞ 1er. desafío: cómo aprovechar LB

- No todos los coloreos encontrados por DSATUR son equitativos

  ☞ 2do. desafío: cómo podar tempranamente subproblemas que no conduzcan a coloreos equitativos

Nuevos criterios de poda

$\underline{\textsc{EqDSatur}(k, \Pi, F)}$:
(UB, LB y $\overline{c}$ variables globales)

- *Paso 1.* Si $U \neq \varnothing$ ir al Paso 3.
- *Paso 2.* Chequear si Π es un k-eqcol ($\lfloor n/k \rfloor \leq |C_j| \leq \lceil n/k \rceil$ para todo $1 \leq j \leq k$). En caso de serlo, hacer $UB \leftarrow k$, $\overline{c} \leftarrow \Pi$ y salir. Si no, salir por *infactibilidad*.
- *Paso 3.* Elegir vértice $u \in U$ de máximo grado de saturación.
- *Paso 4.* Para cada color $1 \leq j \leq UB - 1$ tal que $j \in F(u)$ y $j \leq k + 1$, considere los siguientes casos:
  - *Caso $j \leq k$.* Si
  $$|C_j| + 1 \leq \left\lceil \frac{n}{\max\{k, LB\}} \right\rceil$$
  entonces ejecutar $\textsc{EqDSatur}(k', \Pi', F')$.
  - *Caso $j = k + 1$.* Si $k + 1 \leq LB$ o
  $$|C_i| \leq \left\lceil \frac{n}{k+1} \right\rceil \quad \text{para todo } 1 \leq i \leq k$$
  entonces ejecutar $\textsc{EqDSatur}(k', \Pi', F')$.

Nuevos criterios de poda

Otro criterio de poda:

- Sea $M_j =$ mínimo número de vértices necesarios a agregar en C_j de Π para alcanzar coloreo equitativo.
- Se necesitan al menos $\sum_{j=1}^{k} M_j$ vértices no coloreados.
- Como no es posible saber M_j, podemos estimar una cota inferior:

$$\tilde{M}_j = \text{máx}\left\{0, \left\lfloor \frac{n}{UB-1} \right\rfloor - |C_j|\right\}$$

- Para implementar este criterio, agregamos un paso más: *Paso* $2\frac{1}{2}$. Si $|U| < \sum_{j=1}^{k} \tilde{M}_j$, salir por *infactibilidad*.

Nuevos criterios de poda

Otro criterio de poda:

- Sea M_j = mínimo número de vértices necesarios a agregar en C_j de Π para alcanzar coloreo equitativo.
- Se necesitan al menos $\sum_{j=1}^{k} M_j$ vértices no coloreados.
- Como no es posible saber M_j, podemos estimar una cota inferior:

$$\tilde{M}_j = \max\left\{0, \left\lfloor \frac{n}{UB-1} \right\rfloor - |C_j|\right\}$$

- Para implementar este criterio, agregamos un paso más: *Paso* $2\frac{1}{2}$. Si $|U| < \sum_{j=1}^{k} \tilde{M}_j$, salir por *infactibilidad*.

Nuevos criterios de poda

Otro criterio de poda:

- Sea $M_j =$ mínimo número de vértices necesarios a agregar en C_j de Π para alcanzar coloreo equitativo.
- Se necesitan al menos $\sum_{j=1}^{k} M_j$ vértices no coloreados.
- Como no es posible saber M_j, podemos estimar una cota inferior:
$$\tilde{M}_j = \max\left\{0, \left\lfloor \frac{n}{UB-1} \right\rfloor - |C_j|\right\}$$
- Para implementar este criterio, agregamos un paso más: *Paso $2\frac{1}{2}$*. Si $|U| < \sum_{j=1}^{k} \tilde{M}_j$, salir por *infactibilidad*.

Nuevos criterios de poda

- 30 instancias aleatorias por renglón
- Tiempo máximo: 2 horas

vert.	% d.	% inst. resueltas: sin poda	% inst. resueltas: crit. rojo	% inst. resueltas: rojo y verde	Promedio nodos: sin poda	Promedio nodos: crit. rojo	Promedio nodos: rojo y verde
60	10	100	100	100	394555	**359**	**359**
60	30	100	100	100	19966262	12732	**12676**
60	50	73	**93**	**93**	12093071	278270	**277514**
60	70	87	**93**	**93**	17490967	99492278	**7116094**
60	90	100	100	100	1924812	47579	**14871**
70	10	100	100	100	54591690	226	**220**
70	30	73	**100**	**100**	4064611768	443591	**420352**
70	50	73	**93**	**93**	1220779213	6420622	**6130077**
70	70	90	**93**	**93**	451376116	26222998	**25202357**
70	90	100	100	100	24015409	547187	**396705**

Nuevos criterios de poda

- 30 instancias aleatorias por renglón
- Tiempo máximo: 2 horas

		% inst. resueltas			Prom. Tiempo		
vert.	% d.	sin poda	crit. rojo	rojo y verde	sin poda	crit. rojo	rojo y verde
60	10	100	100	100	0	0	0
60	30	100	100	100	4,1	**0**	**0**
60	50	73	**93**	**93**	3	**0,1**	**0,1**
60	70	87	**93**	**93**	6,9	40	**6,6**
60	90	100	100	100	0,7	**0**	**0**
70	10	100	100	100	5,7	**0**	**0**
70	30	73	**100**	**100**	629	**0,1**	**0,1**
70	50	73	**93**	**93**	249	**7,1**	**7,1**
70	70	90	**93**	**93**	138	**34**	**34**
70	90	100	100	100	8,6	**0,3**	**0,2**

Estrategia de selección de vértices

- 30 instancias aleatorias por renglón
- Tiempo máximo: 2 horas
- (*) = se justifica aumento de promedio debido a que resuelve más instancias

vert.	% d.	% inst. resueltas DSATUR	CELIM	PASS	Prom. Tiempo DSATUR	CELIM	PASS
70	10	100	100	100	0	0	0
70	30	100	100	100	0,1	0,2	0,1
70	50	93	97	**100**	7,1	8,8	7,7(*)
70	70	93	**97**	**97**	34	**27**	**27**
70	90	100	100	100	0,2	0,2	0,1
80	10	100	100	100	0	0	0
80	30	100	100	100	150	**29**	43
80	50	90	83	**93**	753	735	**498**
80	70	40	43	**57**	2339	1739	1922(*)
80	90	97	**100**	**100**	11	13	12

Comparación con CPLEX

- 30 instancias aleatorias por renglón
- Tiempo máximo: 2 horas
- CPLEX utiliza modelo presentado en [Méndez-Díaz, Nasini, S.- (LAGOS 2011)] (con mismas cotas que EQDSATUR y clique maximal coloreada)

	%	% inst. res.		Prom. % Gap Rel.		Prom. Tiempo	
vert.	d.	CPLEX	EQDS	CPLEX	EQDS	CPLEX	EQDS
80	10	100	100	0	0	11	**0**
80	30	0	**100**	22	**0**	–	**43**
80	50	0	**93**	24,6	**2,9**	–	**498**
80	70	3	**57**	16,8	**14,8**	5769	**1922**
80	90	93	**100**	0,2	**0**	752	**12**

Comparación con algoritmos existentes

- 36 instancias DIMACS

Inst.	vert.	edg.	χ_{eq}	% Gap Rel. CPLEX	% Gap Rel. EQDS	Tiempo CPLEX	Tiempo EQDS
1-FullIns_4	93	593	5	0	0	**28**	1915
3-FullIns_3	80	346	6	**0**	28	**0**	–
4-FullIns_3	114	541	7	**0**	50	**0**	–
1-Insertions_4	67	232	5	20	**0**	–	**1468**
4-Insertions_3	79	156	4	0	0	**836**	2250
ash331GPIA	662	4181	4	**50**	62	–	–
DSJC125.1	125	736	5	0	0	214	**0**
DSJC125.5	125	3891	?	**51**	52	–	–
DSJC125.9	125	6961	?	**8**	10	–	–
fpsol2.i.1	496	11654	65	**0**	23	**11**	–
fpsol2.i.2	451	8691	47	**24**	51	–	–
fpsol2.i.3	425	8688	55	**31**	62	–	–
myciel5	47	236	6	0	0	149	**0**
myciel6	95	755	?	**14**	57	–	–
queen8_12	96	1368	12	**0**	40	**5**	–
queen8_8	64	728	9	0	0	654	**10**
queen9_9	81	1056	10	18	**0**	–	**719**
queen10_10	100	1470	?	**16**	23	–	–

Comparación con algoritmos existentes

- 36 instancias DIMACS

Inst.	vert.	edg.	χ_{eq}	% Gap Rel. CPLEX	% Gap Rel. EQDS	Tiempo CPLEX	Tiempo EQDS
le450_15a	450	8168	15	16	**11**	–	–
le450_15b	450	8169	15	11	**6**	–	–
le450_5a	450	5714	5	**0**	58	**4558**	–
le450_5b	450	5734	5	**0**	58	**4305**	–
miles1000	128	3216	42	**0**	14	**0**	–
miles750	128	2113	31	**0**	9	**0**	–
mug100_1	100	166	4	0	0	**1**	6382
mug100_25	100	166	4	**0**	25	**1**	–
mug88_1	88	146	4	0	0	**1**	156
mug88_25	88	146	4	0	0	**0**	80
mulsol.i.1	197	3925	49	**0**	22	**1**	–
mulsol.i.2	188	3885	?	**12**	46	–	–
will199GPIA	701	6772	7	**22**	33	–	–
inithx.i.1	864	18707	54	**0**	22	**63**	–
school1_nsh	352	14612	14	**0**	65	**1840**	–
zeroin.i.1	211	4100	49	0	0	0	0
zeroin.i.2	211	3541	36	**0**	41	**2**	–
zeroin.i.3	206	3540	36	**0**	38	**5**	–

Conclusiones

- Criterio rojo mejora drásticamente la enumeración.
- Criterio verde reduce cantidad de nodos y ayuda un poco más.
- Al igual que en coloreo clásico, es preferible aplicar selección de vértices PASS en PCEG.
- EqDSatur competitivo en instancias aleatorias de medio tamaño. No así en las DIMACS de mayor tamaño.
  ☞ Cplex resolvió 23, EqDSatur sólo 11
- Trabajo Futuro: Probar otras estrategias de selección de colores además de la habitual donde se barren colores en orden ascendente.

Conclusiones

- Criterio rojo mejora drásticamente la enumeración.
- Criterio verde reduce cantidad de nodos y ayuda un poco más.
- Al igual que en coloreo clásico, es preferible aplicar selección de vértices PASS en PCEG.
- EqDSatur competitivo en instancias aleatorias de medio tamaño. No así en las DIMACS de mayor tamaño.
  - ☞ Cplex resolvió 23, EqDSatur sólo 11
- Trabajo Futuro: Probar otras estrategias de selección de colores además de la habitual donde se barren colores en orden ascendente.

Conclusiones

- Criterio rojo mejora drásticamente la enumeración.
- Criterio verde reduce cantidad de nodos y ayuda un poco más.
- Al igual que en coloreo clásico, es preferible aplicar selección de vértices PASS en PCEG.
- EqDSatur competitivo en instancias aleatorias de medio tamaño. No así en las DIMACS de mayor tamaño.
  ☞ Cplex resolvió 23, EqDSatur sólo 11
- Trabajo Futuro: Probar otras estrategias de selección de colores además de la habitual donde se barren colores en orden ascendente.

Conclusiones

- Criterio rojo mejora drásticamente la enumeración.
- Criterio verde reduce cantidad de nodos y ayuda un poco más.
- Al igual que en coloreo clásico, es preferible aplicar selección de vértices PASS en PCEG.
- EqDSatur competitivo en instancias aleatorias de medio tamaño. No así en las DIMACS de mayor tamaño.
  ☞ Cplex resolvió 23, EqDSatur sólo 11
- Trabajo Futuro: Probar otras estrategias de selección de colores además de la habitual donde se barren colores en orden ascendente.

Gracias!