A Branch and Cut Algorithm for the Equitable Graph Coloring Problem

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

INFORMS, Buenos Aires, 2010

Definitions

Classic coloring

Let $G = (V, E)$. A k-coloring is a partition of V in C_1, C_2, ..., C_k such that:

- $uv \in E,\ u \in C_j \implies v \notin C_j$

Equitable coloring

A k-eqcol is a k-coloring such that:

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

Equitable chromatic number

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

- ECP consists in finding $\chi_{eq}(G)$.

Definitions

Classic coloring

Let $G = (V, E)$. A k-coloring is a partition of V in C_1, C_2, ..., C_k such that:

- $uv \in E,\ u \in C_j \implies v \notin C_j$

Equitable coloring

A k-eqcol is a k-coloring such that:

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

Equitable chromatic number

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

- ECP consists in finding $\chi_{eq}(G)$.

Definitions

Classic coloring

Let $G = (V, E)$. A k-coloring is a partition of V in C_1, C_2, ..., C_k such that:

- $uv \in E,\ u \in C_j \implies v \notin C_j$

Equitable coloring

A k-eqcol is a k-coloring such that:

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

Equitable chromatic number

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

- ECP consists in finding $\chi_{eq}(G)$.

A bit of history of ECP...

- Definition and first results [Meyer (1973)]
- Applications:
  - Municipal garbage collection service [Tucker 1973]
  - Traffic signal control [Irani, Leung (1996)]
  - Parallel memory systems [Das, Finocchi, Petreschi (2006)]
- ECP is NP-Hard [Kubale, Furmanczyk (2005)]
- Polynomial algorithm to obtain a $(\Delta + 1)$-eqcol [Kierstead, Kostochka (2008)]
- IP model and C&B [Méndez-Díaz, Nasini, S.- (Alio/Euro 2008)]
- B&C-LF_2 based on *asymmetric representatives* model [Bahiense, Frota, Maculan, Noronha, Ribeiro (LAGOS 2009)]
- IP model breaking more symmetries [Méndez-Díaz, Nasini, S.- (MACI 2009)]

A bit of history of ECP...

- Definition and first results [Meyer (1973)]
- Applications:
  - Municipal garbage collection service [Tucker 1973]
  - Traffic signal control [Irani, Leung (1996)]
  - Parallel memory systems [Das, Finocchi, Petreschi (2006)]
- ECP is NP-Hard [Kubale, Furmanczyk (2005)]
- Polynomial algorithm to obtain a $(\Delta + 1)$-eqcol [Kierstead, Kostochka (2008)]
- IP model and C&B [Méndez-Díaz, Nasini, S.- (Alio/Euro 2008)]
- B&C-LF_2 based on *asymmetric representatives* model [Bahiense, Frota, Maculan, Noronha, Ribeiro (LAGOS 2009)]
- IP model breaking more symmetries [Méndez-Díaz, Nasini, S.- (MACI 2009)]

A bit of history of ECP...

- Definition and first results [Meyer (1973)]
- Applications:
  - Municipal garbage collection service [Tucker 1973]
  - Traffic signal control [Irani, Leung (1996)]
  - Parallel memory systems [Das, Finocchi, Petreschi (2006)]
- ECP is NP-Hard [Kubale, Furmanczyk (2005)]
- Polynomial algorithm to obtain a $(\Delta + 1)$-eqcol [Kierstead, Kostochka (2008)]
- IP model and C&B [Méndez-Díaz, Nasini, S.- (Alio/Euro 2008)]
- B&C-LF_2 based on *asymmetric representatives* model [Bahiense, Frota, Maculan, Noronha, Ribeiro (LAGOS 2009)]
- IP model breaking more symmetries [Méndez-Díaz, Nasini, S.- (MACI 2009)]

A bit of history of ECP...

- Definition and first results [Meyer (1973)]
- Applications:
  - Municipal garbage collection service [Tucker 1973]
  - Traffic signal control [Irani, Leung (1996)]
  - Parallel memory systems [Das, Finocchi, Petreschi (2006)]
- ECP is NP-Hard [Kubale, Furmanczyk (2005)]
- Polynomial algorithm to obtain a $(\Delta + 1)$-eqcol [Kierstead, Kostochka (2008)]
- IP model and C&B [Méndez-Díaz, Nasini, S.- (Alio/Euro 2008)]
- B&C-LF_2 based on *asymmetric representatives* model [Bahiense, Frota, Maculan, Noronha, Ribeiro (LAGOS 2009)]
- IP model breaking more symmetries [Méndez-Díaz, Nasini, S.- (MACI 2009)]

A bit of history of ECP...

- Definition and first results [Meyer (1973)]
- Applications:
  - Municipal garbage collection service [Tucker 1973]
  - Traffic signal control [Irani, Leung (1996)]
  - Parallel memory systems [Das, Finocchi, Petreschi (2006)]
- ECP is NP-Hard [Kubale, Furmanczyk (2005)]
- Polynomial algorithm to obtain a $(\Delta + 1)$-eqcol [Kierstead, Kostochka (2008)]
- IP model and C&B [Méndez-Díaz, Nasini, S.- (Alio/Euro 2008)]
- B&C-LF_2 based on *asymmetric representatives* model [Bahiense, Frota, Maculan, Noronha, Ribeiro (LAGOS 2009)]
- IP model breaking more symmetries [Méndez-Díaz, Nasini, S.- (MACI 2009)]

A bit of history of ECP...

- Definition and first results [Meyer (1973)]
- Applications:
  - Municipal garbage collection service [Tucker 1973]
  - Traffic signal control [Irani, Leung (1996)]
  - Parallel memory systems [Das, Finocchi, Petreschi (2006)]
- ECP is NP-Hard [Kubale, Furmanczyk (2005)]
- Polynomial algorithm to obtain a $(\Delta + 1)$-eqcol [Kierstead, Kostochka (2008)]
- IP model and C&B [Méndez-Díaz, Nasini, S.- (Alio/Euro 2008)]
- B&C-LF_2 based on *asymmetric representatives* model [Bahiense, Frota, Maculan, Noronha, Ribeiro (LAGOS 2009)]
- IP model breaking more symmetries [Méndez-Díaz, Nasini, S.- (MACI 2009)]

A bit of history of ECP...

- Definition and first results [Meyer (1973)]
- Applications:
  - Municipal garbage collection service [Tucker 1973]
  - Traffic signal control [Irani, Leung (1996)]
  - Parallel memory systems [Das, Finocchi, Petreschi (2006)]
- ECP is NP-Hard [Kubale, Furmanczyk (2005)]
- Polynomial algorithm to obtain a $(\Delta + 1)$-eqcol [Kierstead, Kostochka (2008)]
- IP model and C&B [Méndez-Díaz, Nasini, S.- (Alio/Euro 2008)]
- B&C-LF_2 based on *asymmetric representatives* model [Bahiense, Frota, Maculan, Noronha, Ribeiro (LAGOS 2009)]
- IP model breaking more symmetries [Méndez-Díaz, Nasini, S.- (MACI 2009)]

In this work...

- TABUEQCOL: A *tabu search* heuristic for the ECP.
- New branching strategy.
- B&C algorithm that uses TABUEQCOL for computing initial good quality bounds, and the branching strategy.
  - Computational results.

In this work...

- TABUEQCOL: A *tabu search* heuristic for the ECP.
- New branching strategy.
- B&C algorithm that uses TABUEQCOL for computing initial good quality bounds, and the branching strategy.
  - Computational results.

In this work...

- TABUEQCOL: A *tabu search* heuristic for the ECP.
- New branching strategy.
- B&C algorithm that uses TABUEQCOL for computing initial good quality bounds, and the branching strategy.
  - Computational results.

Tabu search heuristic

- **TABUCOL:** [Hertz, de Werra (1987)]
  - *Search space*: partition $(V_1, V_2, \dots, V_k)$
  - *Goal*: Find a proper k-coloring.
    - $\sum_{i=1}^{k} |E(V_i)| = 0 \Leftrightarrow (V_1, V_2, \dots, V_k)$ is a k-coloring
    - Minimize $\sum_{i=1}^{k} |E(V_i)|$
  - *Neighborhood of solution*:
    Given $(V_1, \dots, V_i, \dots, V_j, \dots, V_k)$ and $v \in V_i$.
    Look for $(V_1, \dots, V_i', \dots, V_j', \dots, V_k)$ such that

1. v travels from V_i to V_j $\leftarrow \begin{cases} V_i' = V_i \backslash \{v\}, \\ V_j' = V_j \cup \{v\}. \end{cases}$

Tabu search heuristic

- **TABUCOL:** [Hertz, de Werra (1987)]
  - *Search space*: partition $(V_1, V_2, \ldots, V_k)$
  - *Goal*: Find a proper k-coloring.
    - $\sum_{i=1}^{k} |E(V_i)| = 0 \Leftrightarrow (V_1, V_2, \ldots, V_k)$ is a k-coloring
    - Minimize $\sum_{i=1}^{k} |E(V_i)|$
  - *Neighborhood of solution*:
    Given $(V_1, \ldots, V_i, \ldots, V_j, \ldots, V_k)$ and $v \in V_i$.
    Look for $(V_1, \ldots, V_i', \ldots, V_j', \ldots, V_k)$ such that

    1. v travels from V_i to V_j $\leftarrow \begin{cases} V_i' = V_i \backslash \{v\}, \\ V_j' = V_j \cup \{v\}. \end{cases}$

Tabu search heuristic

- **TABUCOL:** [Hertz, de Werra (1987)]
  - *Search space*: partition $(V_1, V_2, \ldots, V_k)$
  - *Goal*: Find a proper k-coloring.
    - $\sum_{i=1}^{k} |E(V_i)| = 0 \Leftrightarrow (V_1, V_2, \ldots, V_k)$ is a k-coloring
    - Minimize $\sum_{i=1}^{k} |E(V_i)|$
  - *Neighborhood of solution*:
    Given $(V_1, \ldots, V_i, \ldots, V_j, \ldots, V_k)$ and $v \in V_i$.
    Look for $(V_1, \ldots, V_i', \ldots, V_j', \ldots, V_k)$ such that

1. v travels from V_i to V_j $\leftarrow \begin{cases} V_i' = V_i \backslash \{v\}, \\ V_j' = V_j \cup \{v\}. \end{cases}$

Tabu search heuristic

- **TABUCOL:** [Hertz, de Werra (1987)]
  - *Search space*: partition $(V_1, V_2, \dots, V_k)$
  - *Goal*: Find a proper k-coloring.
    - $\sum_{i=1}^{k} |E(V_i)| = 0 \;\Leftrightarrow\; (V_1, V_2, \dots, V_k)$ is a k-coloring
    - Minimize $\sum_{i=1}^{k} |E(V_i)|$
  - *Neighborhood of solution*:
    Given $(V_1, \dots, V_i, \dots, V_j, \dots, V_k)$ and $v \in V_i$.
    Look for $(V_1, \dots, V_i', \dots, V_j', \dots, V_k)$ such that

1. v travels from V_i to V_j $\leftarrow \begin{cases} V_i' = V_i \backslash \{v\}, \\ V_j' = V_j \cup \{v\}. \end{cases}$

A Branch and Cut Algorithm for the ECP

Méndez-Díaz, Nasini, Severin

Introduction
TABUEQCOL
Branching strategies
Computational experiments

- **TABUCOL:** [Hertz, de Werra (1987)]
  - *Search space*: partition $(V_1, V_2, \ldots, V_k)$
  - *Goal*: Find a proper k-coloring.
    - $\sum_{i=1}^{k} |E(V_i)| = 0 \iff (V_1, V_2, \ldots, V_k)$ is a k-coloring
    - Minimize $\sum_{i=1}^{k} |E(V_i)|$
  - *Neighborhood of solution*:
    Given $(V_1, \ldots, V_i, \ldots, V_j, \ldots, V_k)$ and $v \in V_i$.
    Look for $(V_1, \ldots, V_i', \ldots, V_j', \ldots, V_k)$ such that

1. v travels from V_i to V_j $\leftarrow$ $\begin{cases} V_i' = V_i \backslash \{v\}, \\ V_j' = V_j \cup \{v\}. \end{cases}$

Tabu search heuristic

- **TABUCOL:** [Hertz, de Werra (1987)]
  - *Search space*: partition $(V_1, V_2, \ldots, V_k)$
  - *Goal*: Find a proper k-coloring.
    - $\sum_{i=1}^{k} |E(V_i)| = 0 \Leftrightarrow (V_1, V_2, \ldots, V_k)$ is a k-coloring
    - Minimize $\sum_{i=1}^{k} |E(V_i)|$
  - *Neighborhood of solution*:
    Given $(V_1, \ldots, V_i, \ldots, V_j, \ldots, V_k)$ and $v \in V_i$.
    Look for $(V_1, \ldots, V_i', \ldots, V_j', \ldots, V_k)$ such that

    1. v travels from V_i to V_j $\leftarrow \begin{cases} V_i' = V_i \backslash \{v\}, \\ V_j' = V_j \cup \{v\}. \end{cases}$

Tabu search heuristic

- **TABUEQCOL:**
  - *Search space*: partition $(V_1, V_2, \ldots, V_k)$ / $||V_i| - |V_j|| \leq 1$
  - *Goal*: Find a proper k-eqcol.
    - $\sum_{i=1}^{k} |E(V_i)| = 0 \Leftrightarrow (V_1, V_2, \ldots, V_k)$ is a k-eqcol
    - Minimize $\sum_{i=1}^{k} |E(V_i)|$
  - *Neighborhood of solution*:
    Given $(V_1, \ldots, V_i, \ldots, V_j, \ldots, V_k)$ and $v \in V_i$.
    Look for $(V_1, \ldots, V_i', \ldots, V_j', \ldots, V_k)$ such that

$$1. \quad v \text{ travels from } V_i \text{ to } V_j \;\leftarrow\; \begin{cases} V_i' = V_i \backslash \{v\}, \\ V_j' = V_j \cup \{v\}. \end{cases}$$

Tabu search heuristic

- **TABUEQCOL:**
  - *Search space*: partition $(V_1, V_2, \ldots, V_k)$ / $||V_i| - |V_j|| \leq 1$
  - *Goal*: Find a proper k-eqcol.
    - $\sum_{i=1}^{k} |E(V_i)| = 0 \;\Leftrightarrow\; (V_1, V_2, \ldots, V_k)$ is a k-eqcol
    - Minimize $\sum_{i=1}^{k} |E(V_i)|$
  - *Neighborhood of solution*:
    Given $(V_1, \ldots, V_i, \ldots, V_j, \ldots, V_k)$ and $v \in V_i$.
    Look for $(V_1, \ldots, V_i', \ldots, V_j', \ldots, V_k)$ such that

1. v travels from V_i to V_j $\leftarrow \begin{cases} |V_i| = |V_j| + 1, \\ V_i' = V_i \backslash \{v\}, \\ V_j' = V_j \cup \{v\}. \end{cases}$

or

2. Swap v and u $\leftarrow \begin{cases} u \in V_j, \\ V_i' = V_i \cup \{u\} \backslash \{v\}, \\ V_j' = V_j \cup \{v\} \backslash \{u\}. \end{cases}$

$n \mid k \implies |V_i| = |V_j| \;\; \forall j \longrightarrow$ only 2 can be applied

Tabu search heuristic

- **TABUEQCOL:**
  - *Search space*: partition $(V_1, V_2, \ldots, V_k)$ / $||V_i| - |V_j|| \leq 1$
  - *Goal*: Find a proper k-eqcol.
    - $\sum_{i=1}^{k} |E(V_i)| = 0 \Leftrightarrow (V_1, V_2, \ldots, V_k)$ is a k-eqcol
    - Minimize $\sum_{i=1}^{k} |E(V_i)|$
  - *Neighborhood of solution*:
    Given $(V_1, \ldots, V_i, \ldots, V_j, \ldots, V_k)$ and $v \in V_i$.
    Look for $(V_1, \ldots, V_i', \ldots, V_j', \ldots, V_k)$ such that

1. v travels from V_i to V_j $\leftarrow \begin{cases} |V_i| = |V_j| + 1, \\ V_i' = V_i \backslash \{v\}, \\ V_j' = V_j \cup \{v\}. \end{cases}$

or

2. Swap v and u $\leftarrow \begin{cases} u \in V_j, \\ V_i' = V_i \cup \{u\} \backslash \{v\}, \\ V_j' = V_j \cup \{v\} \backslash \{u\}. \end{cases}$

$n \mid k \implies |V_i| = |V_j| \ \forall j \longrightarrow$ only 2 can be applied

Tabu search heuristic

- Tabu tenure:
  - v cannot return to V_i for t iterations.

TABUCOL [Hertz, de Werra (1987)]

- $t = 7$

Dynamic TABUCOL [Galinier, Hao (1999)]

- $t = r + 0.6n_c$
  - r = random number between 0 and 9.
  - n_c = conflictive vertices in current sol.

TABUEQCOL

- $t = 9 + 0.9n_c$
- deterministic

Tabu search heuristic

- Tabu tenure:
  - v cannot return to V_i for t iterations.

TABUCOL [Hertz, de Werra (1987)]

- $t = 7$

Dynamic TABUCOL [Galinier, Hao (1999)]

- $t = r + 0.6 n_c$
  - r = random number between 0 and 9.
  - n_c = conflictive vertices in current sol.

TABUEQCOL

- $t = 9 + 0.9 n_c$
- deterministic

Tabu search heuristic

- Tabu tenure:
  - v cannot return to V_i for t iterations.

TABUCOL [Hertz, de Werra (1987)]

- $t = 7$

Dynamic TABUCOL [Galinier, Hao (1999)]

- $t = r + 0.6n_c$
  - $r =$ random number between 0 and 9.
  - $n_c =$ conflictive vertices in current sol.

TABUEQCOL

- $t = 9 + 0.9n_c$
- deterministic

Tabu search heuristic

- Tabu tenure:
  - v cannot return to V_i for t iterations.

TABUCOL [Hertz, de Werra (1987)]

- $t = 7$

Dynamic TABUCOL [Galinier, Hao (1999)]

- $t = r + 0.6n_c$
  - r = random number between 0 and 9.
  - n_c = conflictive vertices in current sol.

TABUEQCOL

- $t = 9 + 0.9n_c$
- deterministic

Tabu search heuristic

- Tabu tenure:
  - v cannot return to V_i for t iterations.

TABUCOL [Hertz, de Werra (1987)]

- $t = 7$

Dynamic TABUCOL [Galinier, Hao (1999)]

- $t = r + 0.6 n_c$
  - r = random number between 0 and 9.
  - n_c = conflictive vertices in current sol.

TABUEQCOL

- $t = 9 + 0.9 n_c$
- deterministic

Lower and upper bounds from the literature

- Lower bounds of χ_{eq}:
  - Method 1:

$$\chi_{eq}(G) \geq \chi(G) \geq \omega(G)$$

    - Find a maximal clique greedily.
  - Method 2: [Kubale, Furmanczyk (2005)]

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

    - Compute a partition by cliques of $G - N[v]$ $\forall\, v$.
- Upper bounds of χ_{eq}:
  - $\chi_{eq}(G) \leq \Delta(G) + 1$ [Hajnal, Szemerédi (1970)]
  - `Naive` algorithm: SL coloring + balancing procedure [Kubale, Furmanczyk (2005)]

Lower and upper bounds from the literature

- Lower bounds of χ_{eq}:
  - Method 1:

$$\chi_{eq}(G) \geq \chi(G) \geq \omega(G)$$

    - Find a maximal clique greedily.
  - Method 2: [Kubale, Furmanczyk (2005)]

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

    - Compute a partition by cliques of $G - N[v]$ $\forall\, v$.
- Upper bounds of χ_{eq}:
  - $\chi_{eq}(G) \leq \Delta(G) + 1$ [Hajnal, Szemerédi (1970)]
  - Naive algorithm: SL coloring + balancing procedure [Kubale, Furmanczyk (2005)]

Lower and upper bounds from the literature

- Lower bounds of χ_{eq}:
  - Method 1:
    $$\chi_{eq}(G) \geq \chi(G) \geq \omega(G)$$
    - Find a maximal clique greedily.
  - Method 2: [Kubale, Furmanczyk (2005)]
    $$\chi_{eq}(G) \geq \left\lceil \frac{n+1}{\alpha(G - N[v]) + 2} \right\rceil \quad \forall\ v \in V$$
    - Compute a partition by cliques of $G - N[v]$ $\forall\ v$.
- Upper bounds of χ_{eq}:
  - $\chi_{eq}(G) \leq \Delta(G) + 1$ [Hajnal, Szemerédi (1970)]
  - `Naive` algorithm: SL coloring + balancing procedure [Kubale, Furmanczyk (2005)]

Lower and upper bounds from the literature

- Lower bounds of χ_{eq}:
  - Method 1:

$$\chi_{eq}(G) \geq \chi(G) \geq \omega(G)$$

    - Find a maximal clique greedily.
  - Method 2: [Kubale, Furmanczyk (2005)]

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

    - Compute a partition by cliques of $G - N[v] \quad \forall\ v$.
- Upper bounds of χ_{eq}:
  - $\chi_{eq}(G) \leq \Delta(G) + 1$ [Hajnal, Szemerédi (1970)]
  - Naive algorithm: SL coloring + balancing procedure [Kubale, Furmanczyk (2005)]

Lower and upper bounds from the literature

- Lower bounds of χ_{eq}:
  - Method 1:

$$\chi_{eq}(G) \geq \chi(G) \geq \omega(G)$$

    - Find a maximal clique greedily.
  - Method 2: [Kubale, Furmanczyk (2005)]

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

    - Compute a partition by cliques of $G - N[v]$ $\forall\ v$.

- Upper bounds of χ_{eq}:
  - $\chi_{eq}(G) \leq \Delta(G) + 1$ [Hajnal, Szemerédi (1970)]
  - Naive algorithm: SL coloring + balancing procedure [Kubale, Furmanczyk (2005)]

Lower and upper bounds from the literature

- Lower bounds of χ_{eq}:
  - Method 1:

$$\chi_{eq}(G) \geq \chi(G) \geq \omega(G)$$

    - Find a maximal clique greedily.
  - Method 2: [Kubale, Furmanczyk (2005)]

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

    - Compute a partition by cliques of $G - N[v]$ $\forall\ v$.

- Upper bounds of χ_{eq}:
  - $\chi_{eq}(G) \leq \Delta(G) + 1$ [Hajnal, Szemerédi (1970)]
  - `Naive` algorithm: SL coloring + balancing procedure [Kubale, Furmanczyk (2005)]

Tabu search heuristic

- Algorithm:
  1. Obtain $\underline{\chi_{eq}}$ with Method 1 and 2
  2. Obtain a k-eqcol with `Naive` algorithm
  3. $k := k - 1$
  4. Find a k-eqcol with TABUEQCOL
  5. Go to step 3 unless:
     - i Reach 10000 iterations
     - ii Reach 20 seconds
     - iii Optimality: $k = \underline{\chi_{eq}}$

- Preliminary results:

Vertices	Average Rel. Gap
60	17.5%
125	26.4%
500	47.1%

$$\text{Rel. Gap} = \frac{k - \underline{\chi_{eq}}}{k}$$

Tabu search heuristic

- Algorithm:
  1. Obtain $\underline{\chi_{eq}}$ with Method 1 and 2
  2. Obtain a k-eqcol with `Naive` algorithm
  3. $k := k - 1$
  4. Find a k-eqcol with TABUEQCOL
  5. Go to step 3 unless:
     i Reach 10000 iterations
     ii Reach 20 seconds
     iii Optimality: $k = \underline{\chi_{eq}}$

- Preliminary results:

Vertices	Average Rel. Gap
60	17.5%
125	26.4%
500	47.1%

$$\text{Rel. Gap} = \frac{k - \underline{\chi_{eq}}}{k}$$

Tabu search heuristic

- Algorithm:
    1. Obtain $\underline{\chi_{eq}}$ with Method 1 and 2
    2. Obtain a k-eqcol with `Naive` algorithm
    3. $k := k - 1$
    4. Find a k-eqcol with TABUEQCOL
    5. Go to step 3 unless:
        - i Reach 10000 iterations
        - ii Reach 20 seconds
        - iii Optimality: $k = \underline{\chi_{eq}}$

- Preliminary results:
    - Graphs of 60 vertices
        - $k = \underline{\chi_{eq}}$ in 16% of instances.
        - $k = \chi_{eq}$ in 88% of instances.

IP model for the ECP [Méndez-Díaz, Nasini, Severin (2009)]

$$x_{vj} = \begin{cases} 1 & \text{color } j \text{ is assigned to } v, \\ 0 & \text{otherwise,} \end{cases} \qquad w_j = \begin{cases} 1 & x_{vj} = 1 \text{ for some } v, \\ 0 & \text{otherwise.} \end{cases}$$

$$\min \sum_{j=1}^{n} w_j$$

$$s.t. \sum_{j=1}^{n} x_{vj} = 1, \qquad \forall\ v \in V$$

$$x_{uj} + x_{vj} \leq w_j, \qquad \forall\ uv \in E,\ j = 1, \ldots, n$$

$$x_{vj} \leq w_j, \qquad \forall\ v \text{ isolated},\ j = 1, \ldots, n$$

$$w_{j+1} \leq w_j, \qquad \forall\ j = 1, \ldots, n-1$$

$$x_{vj} = 0, \qquad \forall\ j \geq v + 1$$

$$\sum_{v \in V} x_{vj} \geq w_n + \sum_{k=j}^{n-1} \left\lfloor \frac{n}{k} \right\rfloor (w_k - w_{k+1}), \qquad \forall\ j = 1, \ldots, n-1$$

$$\sum_{v \in V} x_{vj} \leq w_n + \sum_{k=j}^{n-1} \left\lceil \frac{n}{k} \right\rceil (w_k - w_{k+1}), \qquad \forall\ j = 1, \ldots, n-1$$

IP model for the ECP [Méndez-Díaz, Nasini, Severin (2009)]

$$x_{vj} = \begin{cases} 1 & \text{color } j \text{ is assigned to } v, \\ 0 & \text{otherwise,} \end{cases} \qquad w_j = \begin{cases} 1 & x_{vj} = 1 \text{ for some } v, \\ 0 & \text{otherwise.} \end{cases}$$

$$\min \sum_{j=1}^{n} w_j$$

$$s.t. \sum_{j=1}^{n} x_{vj} = 1, \qquad \forall\, v \in V$$

$$x_{uj} + x_{vj} \leq w_j, \qquad \forall\, uv \in E,\ j = 1, \ldots, n$$

$$x_{vj} \leq w_j, \qquad \forall\, v \text{ isolated},\ j = 1, \ldots, n$$

$$w_{j+1} \leq w_j, \qquad \forall\, j = 1, \ldots, n-1$$

$$x_{vj} = 0, \qquad \forall\, j \geq v + 1$$

$$\sum_{v \in V} x_{vj} \geq w_n + \sum_{k=j}^{n-1} \left\lfloor \frac{n}{k} \right\rfloor (w_k - w_{k+1}), \qquad \forall\, j = 1, \ldots, n-1$$

$$\sum_{v \in V} x_{vj} \leq w_n + \sum_{k=j}^{n-1} \left\lceil \frac{n}{k} \right\rceil (w_k - w_{k+1}), \qquad \forall\, j = 1, \ldots, n-1$$

IP model for the ECP [Méndez-Díaz, Nasini, Severin (2009)]

$$x_{vj} = \begin{cases} 1 & \text{color } j \text{ is assigned to } v, \\ 0 & \text{otherwise,} \end{cases} \qquad w_j = \begin{cases} 1 & x_{vj} = 1 \text{ for some } v, \\ 0 & \text{otherwise.} \end{cases}$$

$$\min \sum_{j=1}^{n} w_j$$

$$s.t. \sum_{j=1}^{n} x_{vj} = 1, \qquad \forall\, v \in V$$

$$x_{uj} + x_{vj} \leq w_j, \qquad \forall\, uv \in E,\ j = 1, \dots, n$$

$$x_{vj} \leq w_j, \qquad \forall\, v \text{ isolated},\ j = 1, \dots, n$$

$$w_{j+1} \leq w_j, \qquad \forall\, j = 1, \dots, n-1$$

$$x_{vj} = 0, \qquad \forall\, j \geq v + 1$$

$$\sum_{v \in V} x_{vj} \geq w_n + \sum_{k=j}^{n-1} \left\lfloor \frac{n}{k} \right\rfloor (w_k - w_{k+1}), \qquad \forall\, j = 1, \dots, n-1$$

$$\sum_{v \in V} x_{vj} \leq w_n + \sum_{k=j}^{n-1} \left\lceil \frac{n}{k} \right\rceil (w_k - w_{k+1}), \qquad \forall\, j = 1, \dots, n-1$$

IP model for the ECP [Méndez-Díaz, Nasini, Severin (2009)]

$$x_{vj} = \begin{cases} 1 & \text{color } j \text{ is assigned to } v, \\ 0 & \text{otherwise,} \end{cases} \qquad w_j = \begin{cases} 1 & x_{vj} = 1 \text{ for some } v, \\ 0 & \text{otherwise.} \end{cases}$$

$$\min \sum_{j=1}^{n} w_j$$

$$s.t. \sum_{j=1}^{n} x_{vj} = 1, \qquad \forall\ v \in V$$

$$x_{uj} + x_{vj} \leq w_j, \qquad \forall\ uv \in E,\ j = 1, \ldots, n$$

$$x_{vj} \leq w_j, \qquad \forall\ v \text{ isolated},\ j = 1, \ldots, n$$

$$w_{j+1} \leq w_j, \qquad \forall\ j = 1, \ldots, n-1$$

$$x_{vj} = 0, \qquad \forall\ j \geq v + 1$$

$$\sum_{v \in V} x_{vj} \geq w_n + \sum_{k=j}^{n-1} \left\lfloor \frac{n}{k} \right\rfloor (w_k - w_{k+1}), \qquad \forall\ j = 1, \ldots, n-1$$

$$\sum_{v \in V} x_{vj} \leq w_n + \sum_{k=j}^{n-1} \left\lceil \frac{n}{k} \right\rceil (w_k - w_{k+1}), \qquad \forall\ j = 1, \ldots, n-1$$

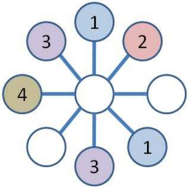
Branching strategies

- *Dichotomic branching*: Compute score for each x_{vj}. Select var. with max. score and generate two subproblems.

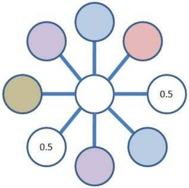
Branching strategies

- *Dichotomic branching*: Compute score for each x_{vj}. Select var. with max. score and generate two subproblems.

- *Score of x_{vj}*:
  - Metric A: $|\{k = 1, \ldots, n \ : \ x^*_{uk} = 1, \ \forall \ u \in N(v)\}|$
    - Number of colors used in neighborhood
  - Metric B: $|\{u \in N(v) \ : \ 0 < x^*_{uj} < 1\}|$
    - Number of fractional var. in neighborhood



Metric A = 4

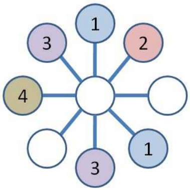


Metric B = 2

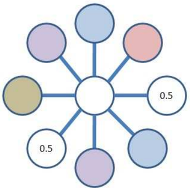
- AB-rule: Use A. In case of tie, use B.
- BA-rule: Use B. In case of tie, use A.

Branching strategies

- *Dichotomic branching*: Compute score for each x_{vj}. Select var. with max. score and generate two subproblems.
- *Score of x_{vj}*:
  - Metric A: $|\{k = 1, \ldots, n \ : \ x^*_{uk} = 1, \ \forall \ u \in N(v)\}|$
    - Number of colors used in neighborhood
  - Metric B: $|\{u \in N(v) \ : \ 0 < x^*_{uj} < 1\}|$
    - Number of fractional var. in neighborhood



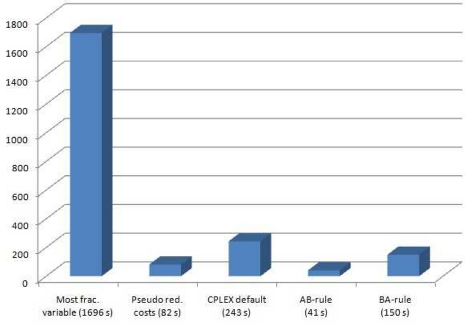
Metric A = 4



Metric B = 2

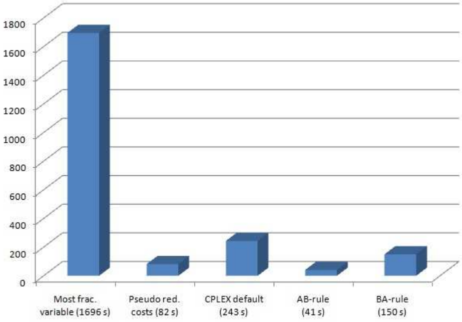
- AB-rule: Use A. In case of tie, use B.
- BA-rule: Use B. In case of tie, use A.

- Graphs of 60 vertices, density: 30% - 70%



- AB-rule is 2 times faster than pseudo reduced costs.

- Graphs of 60 vertices, density: 30% - 70%



- AB-rule is 2 times faster than pseudo reduced costs.

Separation routine

Clique inequalities

Let Q be a maximal clique, and $j = 1, \ldots, n-1$ Then,

$$\sum_{v \in Q} x_{vj} \leq w_j,$$

is a valid inequality of $conv\{(x, w) : (x, w)\ eqcol\}$.

- Separation routine for *clique cuts* are given in [Méndez-Díaz, Zabala (2005)]
  - Good results in a C&B (Alio/Euro 2008)
  - Better results if used in intermediate nodes (MACI 2009)

$$\boxed{\text{ECOPT} = \text{TABUEQCOL} + \text{AB-rule} + \text{Clique cuts}}$$

Separation routine

Clique inequalities

Let Q be a maximal clique, and $j = 1, \ldots, n-1$ Then,

$$\sum_{v \in Q} x_{vj} \leq w_j,$$

is a valid inequality of $conv\{(x, w) : (x, w)\ eqcol\}$.

- Separation routine for *clique cuts* are given in [Méndez-Díaz, Zabala (2005)]
  - Good results in a C&B (Alio/Euro 2008)
  - Better results if used in intermediate nodes (MACI 2009)

ECOPT = TABUEQCOL + AB-rule + Clique cuts

Separation routine

Clique inequalities

Let Q be a maximal clique, and $j = 1, \ldots, n-1$ Then,

$$\sum_{v \in Q} x_{vj} \leq w_j,$$

is a valid inequality of $conv\{(x,w) : (x,w)\ eqcol\}$.

- Separation routine for *clique cuts* are given in [Méndez-Díaz, Zabala (2005)]
    - Good results in a C&B (Alio/Euro 2008)
    - Better results if used in intermediate nodes (MACI 2009)

$$\boxed{\text{ECOPT} = \text{TABUEQCOL} + \text{AB-rule} + \text{Clique cuts}}$$

Computational experiments

- *Testbed*: Random graphs of 70 vertices
- B&C-LF_2 [Ribeiro et al. (LAGOS 2009)]

	B&C-LF_2	
Dens.	% solved	time
10%	100	109
30%	0	–
50%	0	–
70%	100	273
90%	100	11

- Our benchmarks:

	CPLEX 12.1		ECOPT	
Dens.	% solved	time	% solved	time
10%	100	3	100	1
30%	0	–	100	85
50%	0	–	60	3448
70%	0	–	80	1055
90%	10	35	100	3

Computational experiments

- *Testbed*: Random graphs of 70 vertices
- B&C-LF_2 [Ribeiro et al. (LAGOS 2009)]

	B&C-LF_2	
Dens.	% solved	time
10%	100	109
30%	0	–
50%	0	–
70%	100	273
90%	100	11

- Our benchmarks:

	CPLEX 12.1		ECOPT	
Dens.	% solved	time	% solved	time
10%	100	3	100	1
30%	0	–	100	85
50%	0	–	60	3448
70%	0	–	80	1055
90%	10	35	100	3

Computational experiments

- *Testbed*: (easy-medium) DIMACS instances + kneser graphs

Instance	$\|V\|$	$\|E\|$	χ_{eq}	CPLEX LB	CPLEX UB	CPLEX nodes	CPLEX time	ECOPT nodes	ECOPT time	B&C-LF_2 nodes	B&C-LF_2 time
miles750	128	2113	31			0	6	0	1	6	171
miles1000	128	3216	42			0	9	0	2	13	267
miles1500	128	5198	73			0	2	0	2	1	13
zeroin.i.1	211	4100	49			11	400	1	**5**	1	50
zeroin.i.2	211	3541	36	30	36	—	—	655	**17**	23	510
zeroin.i.3	206	3540	36	30	36	—	—	235	**7**	28	491
queen6_6	36	290	7			72	2	0	0	1	1
queen7_7	49	476	7			0	4	☺	0	1	0
queen8_8	64	728	9	9	10	—	—	1151	**26**	297	441
myciel3	11	20	4			4	0	5	0	7	0
myciel4	23	71	5			231	0	151	0	237	5
jean	80	254	10			0	0	☺	0	1	4
anna	138	493	11			0	2	☺	0	2	26
david	87	406	30			0	0	☺	0	1	13
games120	120	638	9			0	1	☺	0	1	30
kneser5_2	10	15	3			0	0	☺	0	1	0
kneser7_2	21	105	6			531	0	461	0	357	6
kneser7_3	35	70	3			0	0	0	0	4	2
kneser9_4	126	315	3			0	0	0	0	4	809
1-FullIns_3	30	100	4			9	0	0	0	34	2
2-FullIns_3	52	201	5			35	2	0	0	84	25
3-FullIns_3	80	346	6			21	7	0	0	38	85
4-FullIns_3	114	541	7			338	164	0	**0**	3	72
5-FullIns_3	154	792	8			30	87	0	**0**	5	268

☺ Instance solved by TABUEQCOL

Computational experiments

- *Testbed*: (easy-medium) DIMACS instances + kneser graphs

Instance	$\|V\|$	$\|E\|$	χ_{eq}	CPLEX LB	CPLEX UB	CPLEX nodes	CPLEX time	ECOPT nodes	ECOPT time
1-FullIns_4	93	593	5			2380	1999	4287	**38**
2-Insertions_3	37	72	4			382	0	473	0
3-Insertions_3	56	110	4			31373	18	6845	5
le450_25a	450	8260	25	2	∞	–	–	☺	**0**
le450_25b	450	8263	25	2	∞	–	–	☺	**0**
homer	561	1628	13			0	642	☺	**0**
huck	74	301	11			0	0	☺	0
inithx.i.1	864	18707	54	2	∞	–	–	0	**49**
mug100_1	100	166	4			979	3	2419	2
mug100_25	100	166	4			1002	1	2865	2
mug88_1	88	146	4			129	0	329	0
mug88_25	88	146	4			348	0	1875	1
mulsol.i.1	197	3925	49			0	24	1	3
kneser11_5	462	1386	3			2016	5081	550	**290**
queen8_12	96	1368	12			70	443	☺	**0**
school1	385	19095	15	2	∞	–	–	0	**49**
school1_nsh	352	14612	14	2	∞	–	–	☺	**13**

☺ Instance solved by TABUEQCOL

- *Testbed*: (hard) DIMACS instances

				CPLEX				ECOPT			
Instance	$\|V\|$	$\|E\|$	χ_{eq}	*LB*	*UB*	nodes	time	*LB*	*UB*	nodes	time
1-Insertions_4	67	232	5			2253369	3376			2243965	3384
4-Insertions_3	79	156	4			5867112	4385			490051	3024
fpsol2.i.1	496	11654	65	63	65	—	—			1	**15**
fpsol2.i.2	451	8691	47	30	∞	—	—			2	**96**
fpsol2.i.3	425	8688	55	32	∞	—	—			557	**195**
DSJC125.1	125	736	5	5	6	—	—			0	**7**
DSJC125.5	125	3891	?	12	∞	—	—	**13**	**19**	—	—
DSJC125.9	125	6961	?	43	∞	—	—	43	**45**	—	—
le450_15a	450	8168	?	2	∞	—	—	**15**	**16**	—	—
le450_15b	450	8169	15	2	∞	—	—			210	**438**
le450_5a	450	5714	5	4	∞	—	—			213	**5313**
le450_5b	450	5734	?	4	∞	—	—	**5**	**7**	—	—
le450_5c	450	9803	5	4	∞	—	—			7	**964**
le450_5d	450	9757	5	2	∞	—	—			7	**1194**
myciel5	47	236	6			129444	2224			27251	**45**
myciel6	95	755	?	5	7	—	—	**6**	7	—	—
queen9_9	81	1056	?	9	11	—	—	9	**10**	—	—

Further work

- Improve performance in graphs with 50%-70% density.
- Separation routines for other valid inequalities.
  - 2-rank: S / $\alpha(S) = 2$, Q / Q clique.

$$\sum_{v \in S} x_{vj} + 2 \sum_{v \in Q} x_{vj} \leq 2 w_j$$

    (seems to work as well as cliques)

  - Some polynomial families identified.
- Primal heuristics.

Further work

- Improve performance in graphs with 50%-70% density.
- Separation routines for other valid inequalities.
  - 2-rank: $S \ / \ \alpha(S) = 2, \quad Q \ / \ Q$ clique.

    $$\sum_{v \in S} x_{vj} + 2 \sum_{v \in Q} x_{vj} \leq 2w_j$$

    (seems to work as well as cliques)
  - Some polynomial families identified.
- Primal heuristics.

Further work

- Improve performance in graphs with 50%-70% density.
- Separation routines for other valid inequalities.
  - 2-rank: $S \ / \ \alpha(S) = 2, \ \ Q \ / \ Q$ clique.

$$\sum_{v \in S} x_{vj} + 2 \sum_{v \in Q} x_{vj} \leq 2w_j$$

(seems to work as well as cliques)

  - Some polynomial families identified.
- Primal heuristics.

Further work

- Improve performance in graphs with 50%-70% density.
- Separation routines for other valid inequalities.
  - 2-rank: $S \; / \; \alpha(S) = 2, \;\; Q \; / \; Q$ clique.

$$\sum_{v \in S} x_{vj} + 2 \sum_{v \in Q} x_{vj} \leq 2w_j$$

    (seems to work as well as cliques)

  - Some polynomial families identified.
- Primal heuristics.

Further work

- Improve performance in graphs with 50%-70% density.
- Separation routines for other valid inequalities.
  - 2-rank: S / $\alpha(S) = 2$, Q / Q clique.

$$\sum_{v \in S} x_{vj} + 2 \sum_{v \in Q} x_{vj} \leq 2w_j$$

    (seems to work as well as cliques)

  - Some polynomial families identified.
- Primal heuristics.

THANKS!