# An exact DSatur-based algorithm for the Equitable 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

VII Latin-American Algorithms, Graphs and Optimization Symposium
México, April 2013

# Outline

# Basic definitions



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

## Classic Coloring

$k$-coloring = partition of $V$ into $k$ non-empty stable sets

. $C_1, C_2, \ldots, C_k \longleftarrow$ color classes

## Equitable Coloring

$k$-eqcol = $k$-coloring such that:

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

or equivalently:

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

## Equitable Chromatic Number

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

- ECP consists of finding $\chi_{eq}(G) \qquad \longleftarrow$ NP-Hard

# Basic definitions

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

## Classic Coloring

$k$-coloring = partition of $V$ into $k$ non-empty stable sets

$C_1, C_2, \ldots, C_k \longleftarrow$ color classes

## Equitable Coloring

$k$-eqcol = $k$-coloring such that:

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

or equivalently:

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

## Equitable Chromatic Number

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

- ECP consists of finding $\chi_{eq}(G) \qquad \longleftarrow$ NP-Hard

# Basic definitions

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

## Classic Coloring

$k$-coloring $=$ partition of $V$ into $k$ non-empty stable sets

$C_1,\ C_2,\ \ldots,\ C_k \longleftarrow$ color classes

## Equitable Coloring

$k$-eqcol $=$ $k$-coloring such that:

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

or equivalently:

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

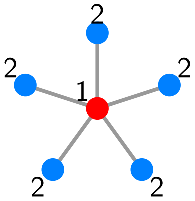
## Equitable Chromatic Number

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

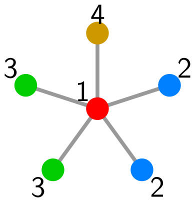
- ECP consists of finding $\chi_{eq}(G)$ $\longleftarrow$ NP-Hard

# An example






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




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

- Since one color class has a single vertex, the remaining color classes can not have more than two vertices in an equitable coloring.

# Brief history about ECP



- Definition and first results [Meyer 1973]
- Assignment of garbage collection routes [Tucker 1973]
- Computational complexity of ECP, applications and heuristics [Kubale, Furmańczyk 2005]
- IP model and C&B [Méndez-Díaz, Nasini, S.- (Alio/Euro 2008)]
- B&C-$LF_2$ based on *asymmetric representative* model [Bahiense, Frota, Maculan, Noronha, Ribeiro (LAGOS 2009)]
- Tabu Search, new branching rule [Méndez-Díaz, Nasini, S.- (INFORMS 2010)]
- New valid inequalities for ECP [Méndez-Díaz, Nasini, S.- (LAGOS 2011)]

# Classic DSatur



DSatur Branch-and-Bound:

- In each subproblem, the graph is partially colored with $k$ colors.
- It chooses an uncolored vertex $u$ with the largest **d**egree **satur**ation.
- In case of tie, it uses an alternative criterion.
- It creates one subproblem per color $j = 1, \ldots, k + 1$, where $u$ is colored with $j$.

Brief history:

- Enumeration scheme [Brown 1972]
- DSatur heuristic and DSatur Branch-and-Bound [Brélaz 1979]
- Improved alternative criterion [Sewell 1996]
- Improved alt. criterion + comparisons with other exact alg. [San Segundo 2012]

# Classic DSatur



DSatur Branch-and-Bound:

- In each subproblem, the graph is partially colored with $k$ colors.
- It chooses an uncolored vertex $u$ with the largest **d**egree **satur**ation.
- In case of tie, it uses an alternative criterion.
- It creates one subproblem per color $j = 1, \ldots, k + 1$, where $u$ is colored with $j$.

Brief history:

- Enumeration scheme [Brown 1972]
- DSatur heuristic and DSatur Branch-and-Bound [Brélaz 1979]
- Improved alternative criterion [Sewell 1996]
- Improved alt. criterion + comparisons with other exact alg. [San Segundo 2012]

# Classic DSatur



DSatur Branch-and-Bound:

- In each subproblem, the graph is partially colored with $k$ colors.
- It chooses an uncolored vertex $u$ with the largest **d**egree **satur**ation.
- In case of tie, it uses an alternative criterion.
- It creates one subproblem per color $j = 1, \ldots, k+1$, where $u$ is colored with $j$.

Brief history:

- Enumeration scheme [Brown 1972]
- DSatur heuristic and DSatur Branch-and-Bound [Brélaz 1979]
- Improved alternative criterion [Sewell 1996]
- Improved alt. criterion + comparisons with other exact alg. [San Segundo 2012]

Since there already exist algorithms for solving the ECP, is it useful to propose a DSatur-based algorithm?

- Simple to implement, they do not require sophisticated optimization engines
- They can be used at some stage in metaheuristics or in more complex exact algorithms
- They can be highly competitive in medium-sized random graphs [San Segundo 2012]

☞ Research on DSatur-based solvers is still important ☺

# Motivation

Since there already exist algorithms for solving the ECP, is it useful to propose a DSatur-based algorithm?

- Simple to implement, they do not require sophisticated optimization engines
- They can be used at some stage in metaheuristics or in more complex exact algorithms
- They can be highly competitive in medium-sized random graphs [San Segundo 2012]

☞ Research on DSatur-based solvers is still important ☺

# Motivation



Since there already exist algorithms for solving the ECP, is it useful to propose a DSatur-based algorithm?

- Simple to implement, they do not require sophisticated optimization engines
- They can be used at some stage in metaheuristics or in more complex exact algorithms
- They can be highly competitive in medium-sized random graphs [San Segundo 2012]

☞ Research on DSatur-based solvers is still important ☺

# Motivation



Since there already exist algorithms for solving the ECP, is it useful to propose a DSatur-based algorithm?

- Simple to implement, they do not require sophisticated optimization engines
- They can be used at some stage in metaheuristics or in more complex exact algorithms
- They can be highly competitive in medium-sized random graphs [San Segundo 2012]

☞ Research on DSatur-based solvers is still important ☺

# Motivation

Since there already exist algorithms for solving the ECP, is it useful to propose a DSatur-based algorithm?

- Simple to implement, they do not require sophisticated optimization engines
- They can be used at some stage in metaheuristics or in more complex exact algorithms
- They can be highly competitive in medium-sized random graphs [San Segundo 2012]

☞ Research on DSatur-based solvers is still important ☺

# Initial bounds for the ECP



- **Upper bound**: Heuristic NAIVE $\longrightarrow \overline{c}$ [Kubale, Furmańczyk 2005]
- **Lower bound**: Two ways
  - A maximal clique $Q$ of $G$ computed greedily
  - [Lih, Chen 1994]

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

We propose:

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

where $\tilde{\alpha}(S) =$ partition of $S$ in cliques, computed greedily

# Initial bounds for the ECP

- **Upper bound**: Heuristic NAIVE $\longrightarrow \overline{c}$
  [Kubale, Furmańczyk 2005]
- **Lower bound**: Two ways
  - A maximal clique $Q$ of $G$ computed greedily
  - [Lih, Chen 1994]

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

We propose:

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

where $\tilde{\alpha}(S)$ = partition of $S$ in cliques, computed greedily

# Initial bounds for the ECP



- **Upper bound**: Heuristic NAIVE $\longrightarrow \overline{c}$
  [Kubale, Furmańczyk 2005]
- **Lower bound**: Two ways
  - A maximal clique $Q$ of $G$ computed greedily
  - [Lih, Chen 1994]

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

We propose:

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

where $\tilde{\alpha}(S)$ = partition of $S$ in cliques, computed greedily

# Initial bounds for the ECP



- **Upper bound**: Heuristic NAIVE $\longrightarrow \overline{c}$
  [Kubale, Furmańczyk 2005]
- **Lower bound**: Two ways
  - A maximal clique $Q$ of $G$ computed greedily
  - [Lih, Chen 1994]

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

We propose:

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

where $\tilde{\alpha}(S)$ = partition of $S$ in cliques, computed greedily

# Initial bounds for the ECP



- **Upper bound**: Heuristic NAIVE $\longrightarrow \overline{c}$ [Kubale, Furmańczyk 2005]
- **Lower bound**: Two ways
  - A maximal clique $Q$ of $G$ computed greedily
  - [Lih, Chen 1994]

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

We propose:

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

where $\tilde{\alpha}(S)$ = partition of $S$ in cliques, computed greedily

# Prunning rules for the ECP



More definitions:

## Partial Coloring

partial $k$-coloring $\Pi = (k, C_1, \ldots, C_n, U, F)$:

- $C_1, \ldots, C_k$ disjoint stable sets of $G$
- $C_{k+1}, \ldots, C_n = \varnothing$
- $U = V \setminus \bigcup_{j=1}^{k} C_j$ (set of uncolored vertices)
- $F(u) = \{j :$ no vertex of $C_j$ is adjacent to $u\ \} \ \forall u \in U$ (set of feasible colors of $u$)

If
- $Q$ = maximal clique of $G$,
- $\Pi_Q$ = partial coloring such that $Q$ is painted with colors $1, 2, \ldots, |Q|$,

then $\Pi_Q$ can be extended to a $\chi_{eq}(G)$-eqcol.

☞ $\Pi_Q$ suitable initial partial coloring

# Prunning rules for the ECP



More definitions:

## Partial Coloring

partial $k$-coloring $\Pi = (k, C_1, \ldots, C_n, U, F)$:

- $C_1, \ldots, C_k$ disjoint stable sets of $G$
- $C_{k+1}, \ldots, C_n = \varnothing$
- $U = V \setminus \bigcup_{j=1}^{k} C_j$ (set of uncolored vertices)
- $F(u) = \{j :$ no vertex of $C_j$ is adjacent to $u\}$ $\forall u \in U$ (set of feasible colors of $u$)

If
- $Q$ = maximal clique of $G$,
- $\Pi_Q$ = partial coloring such that $Q$ is painted with colors $1, 2, \ldots, |Q|$,

then $\Pi_Q$ can be extended to a $\chi_{eq}(G)$-eqcol.

☞ $\Pi_Q$ suitable initial partial coloring

# Prunning rules for the ECP

More definitions:

## Partial Coloring

partial $k$-coloring $\Pi = (k, C_1, \ldots, C_n, U, F)$:

- $C_1, \ldots, C_k$ disjoint stable sets of $G$
- $C_{k+1}, \ldots, C_n = \varnothing$
- $U = V \setminus \bigcup_{j=1}^{k} C_j$ (set of uncolored vertices)
- $F(u) = \{j :$ no vertex of $C_j$ is adjacent to $u$ $\}$ $\forall u \in U$ (set of feasible colors of $u$)

If
- $Q =$ maximal clique of $G$,
- $\Pi_Q =$ partial coloring such that $Q$ is painted with colors $1, 2, \ldots, |Q|$,

then $\Pi_Q$ can be extended to a $\chi_{eq}(G)$-eqcol.

☞ $\Pi_Q$ suitable initial partial coloring

# Prunning rules for the ECP



More definitions:

## Partial Coloring

partial $k$-coloring $\Pi = (k, C_1, \ldots, C_n, U, F)$:

- $C_1, \ldots, C_k$ disjoint stable sets of $G$
- $C_{k+1}, \ldots, C_n = \varnothing$
- $U = V \setminus \bigcup_{j=1}^{k} C_j$ (set of uncolored vertices)
- $F(u) = \{j :$ no vertex of $C_j$ is adjacent to $u$ $\}$ $\forall u \in U$ (set of feasible colors of $u$)

If
- $Q =$ maximal clique of $G$,
- $\Pi_Q =$ partial coloring such that $Q$ is painted with colors $1, 2, \ldots, |Q|$,

then $\Pi_Q$ can be extended to a $\chi_{eq}(G)$-eqcol.

☞ $\Pi_Q$ suitable initial partial coloring

# Prunning rules for the ECP



We wonder when a partial coloring can be extended to an equitable coloring

- $LB$ = initial lower bound
- $UB$ = best solution found so far
- $\Pi = (k, C_1, \ldots, C_n, U, F)$ partial coloring with $k < UB$
- $M$ = size of largest class of $\Pi$

**Lemma 1**

If $\Pi$ can be extended to an $r$-eqcol of $G$ with $r < UB$, then:

$$(P.1) \quad |U| \geq \sum_{t=1}^{k} \left( \max\left\{ M - 1, \left\lfloor \frac{n}{UB - 1} \right\rfloor \right\} - |C_t| \right)^+$$

$$(P.2) \quad M \leq \left\lceil \frac{n}{\max\{k, LB\}} \right\rceil \qquad x^+ = \max\{x, 0\}$$

# Prunning rules for the ECP



We wonder when a partial coloring can be extended to an equitable coloring

- $LB$ = initial lower bound
- $UB$ = best solution found so far
- $\Pi = (k, C_1, \dots, C_n, U, F)$ partial coloring with $k < UB$
- $M$ = size of largest class of $\Pi$

**Lemma 1**

If $\Pi$ can be extended to an $r$-eqcol of $G$ with $r < UB$, then:

$$(P.1) \quad |U| \geq \sum_{t=1}^{k} \left( \max\left\{ M - 1, \left\lfloor \frac{n}{UB - 1} \right\rfloor \right\} - |C_t| \right)^{+}$$

$$(P.2) \quad M \leq \left\lceil \frac{n}{\max\{k, LB\}} \right\rceil \qquad x^{+} = \max\{x, 0\}$$

# Prunning rules for the ECP



We wonder when a partial coloring can be extended to an equitable coloring

- $LB$ = initial lower bound
- $UB$ = best solution found so far
- $\Pi = (k, C_1, \ldots, C_n, U, F)$ partial coloring with $k < UB$
- $M$ = size of largest class of $\Pi$

**Lemma 1**

If $\Pi$ can be extended to an $r$-eqcol of $G$ with $r < UB$, then:

$$(P.1) \quad |U| \geq \sum_{t=1}^{k} \left( \max\left\{ M-1, \left\lfloor \frac{n}{UB-1} \right\rfloor \right\} - |C_t| \right)^{+}$$

$$(P.2) \quad M \leq \left\lceil \frac{n}{\max\{k, LB\}} \right\rceil \qquad x^{+} = \max\{x, 0\}$$

# Prunning rules for the ECP



We wonder when a partial coloring can be extended to an equitable coloring

- $LB$ = initial lower bound
- $UB$ = best solution found so far
- $\Pi = (k, C_1, \ldots, C_n, U, F)$ partial coloring with $k < UB$
- $M$ = size of largest class of $\Pi$

**Lemma 1**

If $\Pi$ can be extended to an $r$-eqcol of $G$ with $r < UB$, then:

$$(P.1) \quad |U| \geq \sum_{t=1}^{k} \left( \max\left\{ M - 1, \left\lfloor \frac{n}{UB - 1} \right\rfloor \right\} - |C_t| \right)^{+}$$

$$(P.2) \quad M \leq \left\lceil \frac{n}{\max\{k, LB\}} \right\rceil \qquad x^{+} = \max\{x, 0\}$$

# Prunning rules for the ECP

**Proof sketch (P.2)**. Let $\Pi^*$ be an $r$-eqcol. Let $M^*$ = size of largest class of $\Pi^*$. Then,

Equity constraint, $r \geq k \longrightarrow M^* \leq \left\lceil \frac{n}{k} \right\rceil$

Equity constraint, $r \geq LB \longrightarrow M^* \leq \left\lceil \frac{n}{LB} \right\rceil$

$\Pi$ can be extended to $\Pi^*$, $\longrightarrow M \leq M^*$

$$\therefore \ (P.2) \quad M \leq \min\left\{ \left\lceil \frac{n}{k} \right\rceil, \left\lceil \frac{n}{LB} \right\rceil \right\} = \left\lceil \frac{n}{\max\{k, LB\}} \right\rceil$$

# Prunning rules for the ECP



**Proof sketch (P.2)**. Let $\Pi^*$ be an $r$-eqcol.
Let $M^* =$ size of largest class of $\Pi^*$. Then,

Equity constraint, $r \geq k \longrightarrow M^* \leq \left\lceil \frac{n}{k} \right\rceil$

Equity constraint, $r \geq LB \longrightarrow M^* \leq \left\lceil \frac{n}{LB} \right\rceil$

$\Pi$ can be extended to $\Pi^*$, $\longrightarrow M \leq M^*$

$$\therefore \; (P.2) \quad M \leq \min\left\{ \left\lceil \frac{n}{k} \right\rceil, \left\lceil \frac{n}{LB} \right\rceil \right\} = \left\lceil \frac{n}{\max\{k, LB\}} \right\rceil$$

# Prunning rules for the ECP

**Proof sketch (P.2)**. Let $\Pi^*$ be an $r$-eqcol.
Let $M^*$ = size of largest class of $\Pi^*$. Then,

Equity constraint, $r \geq k \longrightarrow M^* \leq \left\lceil \frac{n}{k} \right\rceil$

Equity constraint, $r \geq LB \longrightarrow M^* \leq \left\lceil \frac{n}{LB} \right\rceil$

$\Pi$ can be extended to $\Pi^*$, $\longrightarrow M \leq M^*$

$$\therefore \ (P.2) \quad M \leq \min\left\{ \left\lceil \frac{n}{k} \right\rceil, \left\lceil \frac{n}{LB} \right\rceil \right\} = \left\lceil \frac{n}{\max\{k, LB\}} \right\rceil$$

# Prunning rules for the ECP



**Proof sketch (P.2)**. Let $\Pi^*$ be an $r$-eqcol.
Let $M^* =$ size of largest class of $\Pi^*$. Then,

Equity constraint, $r \geq k \longrightarrow M^* \leq \left\lceil \frac{n}{k} \right\rceil$

Equity constraint, $r \geq LB \longrightarrow M^* \leq \left\lceil \frac{n}{LB} \right\rceil$

$\Pi$ can be extended to $\Pi^*$, $\longrightarrow M \leq M^*$

$$\therefore \; (P.2) \quad M \leq \min\left\{ \left\lceil \frac{n}{k} \right\rceil, \left\lceil \frac{n}{LB} \right\rceil \right\} = \left\lceil \frac{n}{\max\{k, LB\}} \right\rceil$$

**Proof sketch (P.1)**. Let $\Pi^*$ be an $r$-eqcol with $r < UB$. Let $m^*$ = size of smallest class of $\Pi^*$. Then,

Equity constr., $r < UB \longrightarrow m^* \geq \left\lfloor \frac{n}{UB-1} \right\rfloor$

$\Pi$ can be extended to $\Pi^* \longrightarrow M^* \geq M \longrightarrow m^* \geq M - 1$.

$$\therefore \; m^* \geq \max\left\{ M-1, \left\lfloor \frac{n}{UB-1} \right\rfloor \right\}$$

# Prunning rules for the ECP

**Proof sketch (P.1)**. Let $\Pi^*$ be an $r$-eqcol with $r < UB$. Let $m^*$ = size of smallest class of $\Pi^*$. Then,

Equity constr., $r < UB \longrightarrow m^* \geq \left\lfloor \dfrac{n}{UB - 1} \right\rfloor$

$\Pi$ can be extended to $\Pi^* \longrightarrow M^* \geq M \longrightarrow m^* \geq M - 1$.

$$\therefore \; m^* \geq \max\left\{ M - 1, \left\lfloor \frac{n}{UB - 1} \right\rfloor \right\}$$

# Prunning rules for the ECP



**Proof sketch (P.1)**. Let $\Pi^*$ be an $r$-eqcol with $r < UB$. Let $m^* =$ size of smallest class of $\Pi^*$. Then,

$$\text{Equity constr., } r < UB \longrightarrow m^* \geq \left\lfloor \frac{n}{UB - 1} \right\rfloor$$

$$\Pi \text{ can be extended to } \Pi^* \longrightarrow M^* \geq M \longrightarrow m^* \geq M - 1.$$

$$\therefore \; m^* \geq \max\left\{ M - 1, \left\lfloor \frac{n}{UB - 1} \right\rfloor \right\}$$

# Prunning rules for the ECP

**Proof sketch (P.1)**. Let $\Pi^*$ be an $r$-eqcol with $r < UB$. Let $m^* =$ size of smallest class of $\Pi^*$. Then,

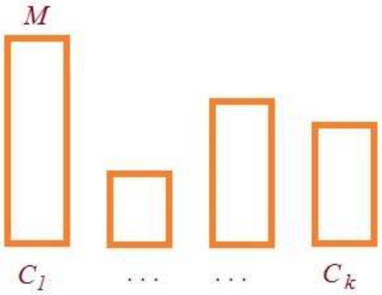
Equity constr., $r < UB \longrightarrow m^* \geq \left\lfloor \dfrac{n}{UB-1} \right\rfloor$

$\Pi$ can be extended to $\Pi^* \longrightarrow M^* \geq M \longrightarrow m^* \geq M - 1$.

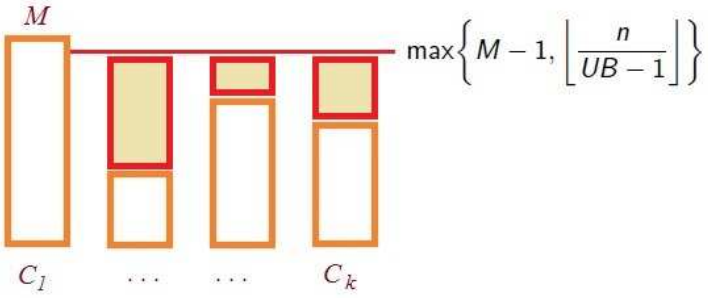
$$\therefore \; m^* \geq \max\left\{ M-1, \left\lfloor \frac{n}{UB-1} \right\rfloor \right\}$$

# Prunning rules for the ECP

**Proof sketch (P.1) (cont.)**. Color classes from $\Pi$ must have at least $\max\left\{M-1, \left\lfloor \frac{n}{UB-1} \right\rfloor\right\}$ vertices.



**Proof sketch (P.1) (cont.).** Color classes from $\Pi$ must have at least $\max\left\{M-1, \left\lfloor \frac{n}{UB-1} \right\rfloor\right\}$ vertices.



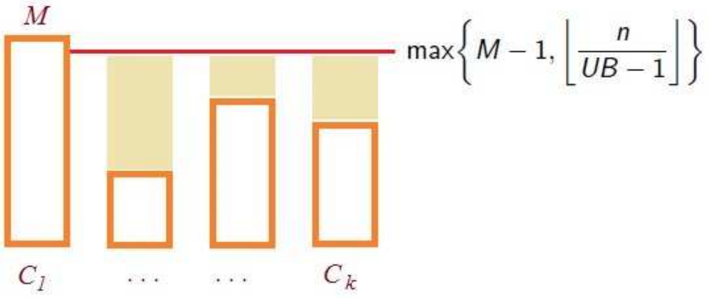

$$\textcolor{red}{\textit{Needed vertices}} = \sum_{t=1}^{k}\left(\max\left\{M-1, \left\lfloor \frac{n}{UB-1} \right\rfloor\right\} - |C_t|\right)^{+}$$

# Prunning rules for the ECP



**Proof sketch (P.1) (cont.)**. Color classes from $\Pi$ must have at least $\max\left\{M-1, \left\lfloor \frac{n}{UB-1} \right\rfloor\right\}$ vertices.

In order to reach these levels, $\Pi$ must have at least the same amount of uncolored vertices.




$$\therefore \quad (P.1) \quad |U| \geq \sum_{t=1}^{k} \left( \max\left\{M-1, \left\lfloor \frac{n}{UB-1} \right\rfloor\right\} - |C_t| \right)^{+}$$

# Prunning rules for the ECP



**Lemma 2**

If $\Pi$ is a partial $k$-coloring satisfying property (P.1) and $U = \varnothing$ then $\Pi$ is a $k$-eqcol.

$U = \varnothing \longrightarrow \Pi$ is a $k$-coloring

If $M - 1 \geq \left\lfloor \dfrac{n}{UB - 1} \right\rfloor$

$|U| = 0$, (P.1) $\longrightarrow \sum_{t=1}^{k} \left(M - 1 - |C_t|\right)^{+} = 0$

$\longrightarrow M - 1 \leq |C_t| \leq M \quad \forall\ t \longrightarrow \Pi$ is equitable

# Prunning rules for the ECP



## Lemma 2

If $\Pi$ is a partial $k$-coloring satisfying property (P.1) and $U = \varnothing$ then $\Pi$ is a $k$-eqcol.

$U = \varnothing \longrightarrow \Pi$ is a $k$-coloring

If $M - 1 \geq \left\lfloor \dfrac{n}{UB - 1} \right\rfloor$

$|U| = 0$, (P.1) $\longrightarrow \sum_{t=1}^{k} \left(M - 1 - |C_t|\right)^{+} = 0$

$\longrightarrow M - 1 \leq |C_t| \leq M \quad \forall\, t \longrightarrow \Pi$ is equitable

# Prunning rules for the ECP

## Lemma 2

If $\Pi$ is a partial $k$-coloring satisfying property (P.1) and $U = \emptyset$ then $\Pi$ is a $k$-eqcol.

$U = \emptyset \longrightarrow \Pi$ is a $k$-coloring

If $M - 1 \geq \left\lfloor \dfrac{n}{UB - 1} \right\rfloor$

$|U| = 0$, (P.1) $\longrightarrow \sum_{t=1}^{k} \left(M - 1 - |C_t|\right)^{+} = 0$

$\longrightarrow M - 1 \leq |C_t| \leq M \ \ \forall\ t \longrightarrow \Pi$ is equitable

# Prunning rules for the ECP



**Lemma 2**

If $\Pi$ is a partial $k$-coloring satisfying property (P.1) and $U = \emptyset$ then $\Pi$ is a $k$-eqcol.

$U = \emptyset \longrightarrow \Pi$ is a $k$-coloring

If $M - 1 < \left\lfloor \frac{n}{UB - 1} \right\rfloor$

$$|U| = 0,\ (P.1) \longrightarrow \sum_{t=1}^{k} \left( \left\lfloor \frac{n}{UB - 1} \right\rfloor - |C_t| \right)^{+} = 0$$

$$\longrightarrow \left\lfloor \frac{n}{UB - 1} \right\rfloor \leq |C_t| \leq M \quad \forall\ t$$

# Prunning rules for the ECP



**Lemma 2**

If $\Pi$ is a partial $k$-coloring satisfying property (P.1) and $U = \varnothing$ then $\Pi$ is a $k$-eqcol.

$U = \varnothing \longrightarrow \Pi$ is a $k$-coloring

If $M - 1 < \left\lfloor \frac{n}{UB - 1} \right\rfloor$

$$|U| = 0,\ (P.1) \longrightarrow \sum_{t=1}^{k} \left( \left\lfloor \frac{n}{UB-1} \right\rfloor - |C_t| \right)^{+} = 0$$

$$\longrightarrow \left\lfloor \frac{n}{UB-1} \right\rfloor = |C_t| = M \ \ \forall\ t \longrightarrow \Pi \text{ is equitable}$$

If we use (P.1) as a prunning rule, every time a coloring is reached at a leaf of the search tree, it is already equitable

# Prunning rules for the ECP



## Lemma 2

If $\Pi$ is a partial $k$-coloring satisfying property (P.1) and $U = \varnothing$ then $\Pi$ is a $k$-eqcol.

$U = \varnothing \longrightarrow \Pi$ is a $k$-coloring

If $M - 1 < \left\lfloor \frac{n}{UB - 1} \right\rfloor$

$$|U| = 0,\ (P.1) \longrightarrow \sum_{t=1}^{k} \left( \left\lfloor \frac{n}{UB-1} \right\rfloor - |C_t| \right)^{+} = 0$$

$$\longrightarrow \left\lfloor \frac{n}{UB-1} \right\rfloor = |C_t| = M \quad \forall\ t \longrightarrow \Pi \text{ is equitable}$$

If we use (P.1) as a prunning rule, every time a coloring is reached at a leaf of the search tree, it is already equitable

# The algorithm EqDSatur

<u>INIT.</u>: $G$ graph, $\overline{c}$ $UB$-**eqcol**, **LB lower bound of** $\chi_{eq}(G)$, $Q$ maximal clique of $G$.

<u>NODE($\Pi = (k, C_1, \ldots, C_n, U, F)$)</u>: ($UB$ and $\overline{c}$ global var.)

- *Step 1.* If $U = \varnothing$, set $UB \leftarrow k$, $\overline{c} \leftarrow \Pi$ and return.
- *Step 2.* Select $u \in U$ according to [San Segundo 2012].
- *Step 3.* For each color $1 \leq j \leq \min\{k+1, UB-1\}$ such that $j \in F(u)$, do:
  $\Pi' \leftarrow (\langle u, j \rangle \hookrightarrow \Pi)$
  If $F'(v) \cap \{1, \ldots, UB-1\} \neq \varnothing \; \forall v \in U'$ **and** $\Pi'$ **satisfies P.1 and P.2**, execute NODE($\Pi'$).

**Theorem**

The recursive execution of NODE($\Pi_Q$) gives the value of $\chi_{eq}(G)$ into $UB$ and an optimal eqcol into $\overline{c}$.

# The algorithm EqDSatur

<u>INIT.</u>: $G$ graph, $\overline{c}$ $UB$-**eqcol**, **LB lower bound of** $\chi_{eq}(G)$, $Q$ maximal clique of $G$.

<u>NODE($\Pi = (k, C_1, \ldots, C_n, U, F)$)</u>: ($UB$ and $\overline{c}$ global var.)

- *Step 1.* If $U = \varnothing$, set $UB \leftarrow k$, $\overline{c} \leftarrow \Pi$ and return.
- *Step 2.* Select $u \in U$ according to [San Segundo 2012].
- *Step 3.* For each color $1 \leq j \leq \min\{k+1, UB-1\}$ such that $j \in F(u)$, do:
  $\Pi' \leftarrow (\langle u, j\rangle \hookrightarrow \Pi)$
  If $F'(v) \cap \{1, \ldots, UB-1\} \neq \varnothing$ $\forall v \in U'$ **and** $\Pi'$ **satisfies P.1 and P.2**, execute NODE($\Pi'$).

## Theorem

The recursive execution of NODE($\Pi_Q$) gives the value of $\chi_{eq}(G)$ into $UB$ and an optimal eqcol into $\overline{c}$.

# Computational experiments



We compare:

- CPLEX = CPLEX 12.1 with formulation given in [Méndez-Díaz, Nasini, S.- (DAM 2012)]
- $LF_2$ = Branch-and-Cut given in [Bahiense, Frota, Maculan, Noronha, Ribeiro (DAM 2011)]
- $\text{EqDS}_1$ = EqDSatur
- $\text{EqDS}_2$ = EqDSatur that order color classes according to their size: if $|C_{i_1}| \leq |C_{i_2}| \leq \ldots \leq |C_{i_k}|$, we evaluate first $j = i_1$, then $j = i_2$, ..., $j = i_k$, $j = k + 1$.
  *Advantage*: Tends to increase smallest class first, thus balancing sizes of classes and finding equitable colorings early.
  *Disadvantage*: QuickSort is required at each node evaluation.

# Computational experiments

We compare:

- CPLEX = CPLEX 12.1 with formulation given in [Méndez-Díaz, Nasini, S.- (DAM 2012)]
- $LF_2$ = Branch-and-Cut given in [Bahiense, Frota, Maculan, Noronha, Ribeiro (DAM 2011)]
- $\text{EqDS}_1$ = EqDSatur
- $\text{EqDS}_2$ = EqDSatur that order color classes according to their size: if $|C_{i_1}| \leq |C_{i_2}| \leq \ldots \leq |C_{i_k}|$, we evaluate first $j = i_1$, then $j = i_2$, …, $j = i_k$, $j = k + 1$.
  *Advantage*: Tends to increase smallest class first, thus balancing sizes of classes and finding equitable colorings early.
  *Disadvantage*: QuickSort is required at each node evaluation.

# Computational experiments

We compare:

- CPLEX = CPLEX 12.1 with formulation given in [Méndez-Díaz, Nasini, S.- (DAM 2012)]
- $LF_2$ = Branch-and-Cut given in [Bahiense, Frota, Maculan, Noronha, Ribeiro (DAM 2011)]
- $\textsc{EqDS}_1 = \textsc{EqDSatur}$
- $\textsc{EqDS}_2 = \textsc{EqDSatur}$ that order color classes according to their size: if $|C_{i_1}| \leq |C_{i_2}| \leq \ldots \leq |C_{i_k}|$, we evaluate first $j = i_1$, then $j = i_2$, ..., $j = i_k$, $j = k + 1$.
  *Advantage*: Tends to increase smallest class first, thus balancing sizes of classes and finding equitable colorings early.
  *Disadvantage*: QuickSort is required at each node evaluation.

- 10 random instances per row (total: 100)
- Max. time: 2 horas
- Values of $LF_2$ are taken from [Bahiense, Frota, Maculan, Noronha, Ribeiro (DAM 2011)] (different platform and instances)

| | %Density | % solved inst. | | | % Rel. Gap. | | | Time | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Vertices | Graph | $LF_2$ | EqDS$_1$ | EqDS$_2$ | $LF_2$ | EqDS$_1$ | EqDS$_2$ | $LF_2$ | EqDS$_1$ | EqDS$_2$ |
| 70 | 10 | 100 | 100 | 100 | 0 | 0 | 0 | 109 | 0 | 0 |
| 70 | 30 | 0 | 100 | 100 | 18 | 0 | 0 | — | 0 | 0 |
| 70 | 50 | 0 | 100 | 100 | 8.2 | 0 | 0 | — | 16.2 | 16.1 |
| 70 | 70 | 100 | 100 | 100 | 0 | 0 | 0 | 273 | 31.2 | 32.1 |
| 70 | 90 | 100 | 100 | 100 | 0 | 0 | 0 | 10.4 | 0 | 0 |
| | | CPLEX | EqDS$_1$ | EqDS$_2$ | CPLEX | EqDS$_1$ | EqDS$_2$ | CPLEX | EqDS$_1$ | EqDS$_2$ |
| 80 | 10 | 100 | 100 | 100 | 0 | 0 | 0 | 5.7 | 0 | 0 |
| 80 | 30 | 0 | 100 | 100 | 20 | 0 | 0 | — | 23.7 | 17.6 |
| 80 | 50 | 0 | 100 | 100 | 24 | 0 | 0 | — | 477 | 424 |
| 80 | 70 | 10 | 70 | 70 | 16 | 10 | 10 | 5769 | 1715 | 1653 |
| 80 | 90 | 100 | 100 | 100 | 0 | 0 | 0 | 690 | 12.5 | 12.3 |

☞ EqDSatur highly competitive

☞ Ordering color classes seems to improve performance

# Computational experiments



- 10 random instances per row (total: 100)
- Max. time: 2 horas
- Values of $LF_2$ are taken from [Bahiense, Frota, Maculan, Noronha, Ribeiro (DAM 2011)] (different platform and instances)

| Vertices | %Density Graph | % solved inst. $LF_2$ | % solved inst. EqDS$_1$ | % solved inst. EqDS$_2$ | % Rel. Gap. $LF_2$ | % Rel. Gap. EqDS$_1$ | % Rel. Gap. EqDS$_2$ | Time $LF_2$ | Time EqDS$_1$ | Time EqDS$_2$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 70 | 10 | 100 | 100 | 100 | 0 | 0 | 0 | 109 | 0 | 0 |
| 70 | 30 | 0 | 100 | 100 | 18 | 0 | 0 | — | 0 | 0 |
| 70 | 50 | 0 | 100 | 100 | 8.2 | 0 | 0 | — | 16.2 | 16.1 |
| 70 | 70 | 100 | 100 | 100 | 0 | 0 | 0 | 273 | 31.2 | 32.1 |
| 70 | 90 | 100 | 100 | 100 | 0 | 0 | 0 | 10.4 | 0 | 0 |
| | | CPLEX | EqDS$_1$ | EqDS$_2$ | CPLEX | EqDS$_1$ | EqDS$_2$ | CPLEX | EqDS$_1$ | EqDS$_2$ |
| 80 | 10 | 100 | 100 | 100 | 0 | 0 | 0 | 5.7 | 0 | 0 |
| 80 | 30 | 0 | 100 | 100 | 20 | 0 | 0 | — | 23.7 | 17.6 |
| 80 | 50 | 0 | 100 | 100 | 24 | 0 | 0 | — | 477 | 424 |
| 80 | 70 | 10 | 70 | 70 | 16 | 10 | 10 | 5769 | 1715 | 1653 |
| 80 | 90 | 100 | 100 | 100 | 0 | 0 | 0 | 690 | 12.5 | 12.3 |

- ☞ EqDSatur highly competitive
- ☞ Ordering color classes seems to improve performance

# Computational experiments

- 10 random instances per row (total: 100)
- Max. time: 2 horas
- Values of $LF_2$ are taken from [Bahiense, Frota, Maculan, Noronha, Ribeiro (DAM 2011)] (different platform and instances)

| Vertices | %Density Graph | % solved inst. $LF_2$ | EqDS$_1$ | EqDS$_2$ | % Rel. Gap. $LF_2$ | EqDS$_1$ | EqDS$_2$ | Time $LF_2$ | EqDS$_1$ | EqDS$_2$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 70 | 10 | 100 | 100 | 100 | 0 | 0 | 0 | 109 | 0 | 0 |
| 70 | 30 | 0 | 100 | 100 | 18 | 0 | 0 | — | 0 | 0 |
| 70 | 50 | 0 | 100 | 100 | 8.2 | 0 | 0 | — | 16.2 | 16.1 |
| 70 | 70 | 100 | 100 | 100 | 0 | 0 | 0 | 273 | 31.2 | 32.1 |
| 70 | 90 | 100 | 100 | 100 | 0 | 0 | 0 | 10.4 | 0 | 0 |
| | | CPLEX | EqDS$_1$ | EqDS$_2$ | CPLEX | EqDS$_1$ | EqDS$_2$ | CPLEX | EqDS$_1$ | EqDS$_2$ |
| 80 | 10 | 100 | 100 | 100 | 0 | 0 | 0 | 5.7 | 0 | 0 |
| 80 | 30 | 0 | 100 | 100 | 20 | 0 | 0 | — | 23.7 | 17.6 |
| 80 | 50 | 0 | 100 | 100 | 24 | 0 | 0 | — | 477 | 424 |
| 80 | 70 | 10 | 70 | 70 | 16 | 10 | 10 | 5769 | 1715 | 1653 |
| 80 | 90 | 100 | 100 | 100 | 0 | 0 | 0 | 690 | 12.5 | 12.3 |

☞ EqDSatur highly competitive

☞ Ordering color classes seems to improve performance

# Computational experiments

- DIMACS COLORLIB instances + Kneser graphs

| Name | Vertices | Edges | $\chi_{eq}$ | Time CPLEX | Time $\text{EqDS}_1$ | Time $\text{EqDS}_2$ | Time $LF_2$ |
|---|---|---|---|---|---|---|---|
| 1-FullIns_3 | 30 | 100 | 4 | 0 | 0 | 0 | 2 |
| 2-FullIns_3 | 52 | 201 | 5 | 0 | 1 | 1 | 25 |
| anna | 138 | 493 | 11 | 0 | 0 | 0 | 26 |
| david | 87 | 406 | 30 | 0 | 0 | 0 | 13 |
| games120 | 120 | 638 | 9 | 0 | 0 | 0 | 30 |
| jean | 80 | 254 | 10 | 0 | 0 | 0 | 4 |
| kneser5_2 | 10 | 15 | 3 | 0 | 0 | 0 | 0 |
| kneser7_2 | 21 | 105 | 6 | 0 | 0 | 0 | 6 |
| kneser7_3 | 35 | 70 | 3 | 0 | 0 | 0 | 2 |
| kneser9_4 | 126 | 315 | 3 | 0 | 0 | 0 | 809 |
| miles1500 | 128 | 5198 | 73 | 0 | 0 | 0 | 13 |
| myciel3 | 11 | 20 | 4 | 0 | 0 | 0 | 0 |
| myciel4 | 23 | 71 | 5 | 0 | 0 | 0 | 5 |
| queen6_6 | 36 | 290 | 7 | 1 | 0 | 0 | 1 |
| queen7_7 | 49 | 476 | 7 | 0 | 0 | 0 | 0 |
| zeroin.i.1 | 211 | 4100 | 49 | 0 | 0 | 0 | 50 |

# Computational experiments



| Name | Vertices | Edges | $\chi_{eq}$ | Time CPLEX | $\mathrm{EQDS}_1$ | $\mathrm{EQDS}_2$ | $LF_2$ |
|---|---|---|---|---|---|---|---|
| 3-FullIns_3 | 80 | 346 | 6 | 0 | – | – | 85 |
| 4-FullIns_3 | 114 | 541 | 7 | 0 | – | – | 72 |
| 5-FullIns_3 | 154 | 792 | 8 | 0 | – | – | 268 |
| miles1000 | 128 | 3216 | 42 | 0 | – | 0 | 267 |
| miles750 | 128 | 2113 | 31 | 0 | – | 0 | 171 |
| queen8_8 | 64 | 728 | 9 | 654 | 7.5 | 1.1 | 441 |
| zeroin.i.2 | 211 | 3541 | 36 | 2 | – | – | 510 |
| zeroin.i.3 | 206 | 3540 | 36 | 5 | – | – | 491 |
| 1-Insertions_4 | 67 | 232 | 5 | – | 1468 | 1540 | |
| myciel5 | 47 | 236 | 6 | 149 | 0 | 0 | |
| queen8_12 | 96 | 1368 | 12 | 5 | – | – | |
| queen9_9 | 81 | 1056 | 10 | – | 719 | 578 | |
| will199GPIA | 701 | 6772 | 7 | – | – | 2 | |
| ash331GPIA | 662 | 4181 | 4 | – | – | 2 | |
| kneser11_5 | 462 | 1386 | 3 | 84 | 2973 | – | |
| mulsol.i.1 | 197 | 3925 | 49 | 1 | – | 0 | |
| Total solved instances | | | | 28 | 21 | 25 | $\geq$24 |

# Computational experiments

| Name | Vertices | Edges | $\chi_{eq}$ | Time | | | |
|---|---|---|---|---|---|---|---|
| | | | | CPLEX | EqDS$_1$ | EqDS$_2$ | $LF_2$ |
| **3-FullIns_3** | 80 | 346 | 6 | 0 | – | – | 85 |
| **4-FullIns_3** | 114 | 541 | 7 | 0 | – | – | 72 |
| **5-FullIns_3** | 154 | 792 | 8 | 0 | – | – | 268 |
| miles1000 | 128 | 3216 | 42 | 0 | – | 0 | 267 |
| miles750 | 128 | 2113 | 31 | 0 | – | 0 | 171 |
| queen8_8 | 64 | 728 | 9 | 654 | 7.5 | 1.1 | 441 |
| zeroin.i.2 | 211 | 3541 | 36 | 2 | – | – | 510 |
| zeroin.i.3 | 206 | 3540 | 36 | 5 | – | – | 491 |
| 1-Insertions_4 | 67 | 232 | 5 | – | 1468 | 1540 | |
| myciel5 | 47 | 236 | 6 | 149 | 0 | 0 | |
| queen8_12 | 96 | 1368 | 12 | 5 | – | – | |
| queen9_9 | 81 | 1056 | 10 | – | 719 | 578 | |
| will199GPIA | 701 | 6772 | 7 | – | – | 2 | |
| ash331GPIA | 662 | 4181 | 4 | – | – | 2 | |
| kneser11_5 | 462 | 1386 | 3 | 84 | 2973 | – | |
| mulsol.i.1 | 197 | 3925 | 49 | 1 | – | 0 | |
| Total solved instances | | | | 28 | 21 | 25 | ≥24 |

# Computational experiments

| | | | | Time | | | |
|---|---|---|---|---|---|---|---|
| Name | Vertices | Edges | $\chi_{eq}$ | CPLEX | $\text{EqDS}_1$ | $\text{EqDS}_2$ | $LF_2$ |
| 3-FullIns_3 | 80 | 346 | 6 | 0 | – | – | 85 |
| 4-FullIns_3 | 114 | 541 | 7 | 0 | – | – | 72 |
| 5-FullIns_3 | 154 | 792 | 8 | 0 | – | – | 268 |
| miles1000 | 128 | 3216 | 42 | 0 | – | 0 | 267 |
| miles750 | 128 | 2113 | 31 | 0 | – | 0 | 171 |
| **queen8_8** | 64 | 728 | 9 | 654 | **7.5** | **1.1** | 441 |
| zeroin.i.2 | 211 | 3541 | 36 | 2 | – | – | 510 |
| zeroin.i.3 | 206 | 3540 | 36 | 5 | – | – | 491 |
| 1-Insertions_4 | 67 | 232 | 5 | – | 1468 | 1540 | |
| **myciel5** | 47 | 236 | 6 | 149 | **0** | **0** | |
| queen8_12 | 96 | 1368 | 12 | 5 | – | – | |
| **queen9_9** | 81 | 1056 | 10 | – | **719** | **578** | |
| will199GPIA | 701 | 6772 | 7 | – | – | 2 | |
| ash331GPIA | 662 | 4181 | 4 | – | – | 2 | |
| kneser11_5 | 462 | 1386 | 3 | 84 | 2973 | – | |
| mulsol.i.1 | 197 | 3925 | 49 | 1 | – | 0 | |
| Total solved instances | | | | 28 | 21 | 25 | $\geq$24 |

☞ Not bad for an enumerative scheme, right? ☺

# Computational experiments



| Name | Vertices | Edges | $\chi_{eq}$ | Time CPLEX | Time $\text{EqDS}_1$ | Time $\text{EqDS}_2$ | Time $LF_2$ |
|---|---|---|---|---|---|---|---|
| 3-FullIns_3 | 80 | 346 | 6 | 0 | – | – | 85 |
| 4-FullIns_3 | 114 | 541 | 7 | 0 | – | – | 72 |
| 5-FullIns_3 | 154 | 792 | 8 | 0 | – | – | 268 |
| miles1000 | 128 | 3216 | 42 | 0 | – | 0 | 267 |
| miles750 | 128 | 2113 | 31 | 0 | – | 0 | 171 |
| **queen8_8** | 64 | 728 | 9 | 654 | **7.5** | **1.1** | 441 |
| zeroin.i.2 | 211 | 3541 | 36 | 2 | – | – | 510 |
| zeroin.i.3 | 206 | 3540 | 36 | 5 | – | – | 491 |
| 1-Insertions_4 | 67 | 232 | 5 | – | 1468 | 1540 | |
| **myciel5** | 47 | 236 | 6 | 149 | **0** | **0** | |
| queen8_12 | 96 | 1368 | 12 | 5 | – | – | |
| **queen9_9** | 81 | 1056 | 10 | – | **719** | **578** | |
| will199GPIA | 701 | 6772 | 7 | – | – | 2 | |
| ash331GPIA | 662 | 4181 | 4 | – | – | 2 | |
| kneser11_5 | 462 | 1386 | 3 | 84 | 2973 | – | |
| mulsol.i.1 | 197 | 3925 | 49 | 1 | – | 0 | |
| Total solved instances | | | | 28 | 21 | 25 | ≥24 |

☞ Not bad for an enumerative scheme, right? ☺


Thanks!