An exact DSatur-based algorithm for the Equitable Coloring Problem

Isabel Méndez-Díaz\textsuperscript{2} Graciela Nasini\textsuperscript{1,3} Daniel Severin\textsuperscript{1,3}

\textsuperscript{1} FCEIA, UNR, \{nasini, daniel\}@fceia.unr.edu.ar
\textsuperscript{2} FCEyN, UBA, imendez@dc.uba.ar
\textsuperscript{3} CONICET, Argentina

VII Latin-American Algorithms, Graphs and Optimization Symposium
México, April 2013
The Equitable Coloring Problem (ECP)
  - Basic definitions
  - An example
  - Brief history of the ECP

Classic DSatur

Motivation

EqDSatur: An exact algorithm for the ECP
  - Initial bounds for the ECP
  - Pruning rules for the ECP
  - The algorithm EqDSatur

Computational experiments
An exact DSatur-based algorithm for the Equitable Coloring Problem

Méndez-Díaz, Nasini, Severin

Introduction

Classic DSatur

Motivation

EqDSatur: An exact algorithm for the ECP

Computational experiments

Basic definitions

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

Classic Coloring

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

$$C_1, C_2, \ldots, C_k \leftarrow \text{color classes}$$

Equitable Coloring

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

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

or equivalently:

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

Equitable Chromatic Number

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

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

\[ G = (V, E) \quad V = \{1, \ldots, n\} \]

Classic Coloring

\[ k\text{-coloring} = \text{partition of } V \text{ into } k \text{ non-empty stable sets} \]

\[ C_1, C_2, \ldots, C_k \leftarrow \text{color classes} \]

Equitable Coloring

\[ k\text{-eqcol} = k\text{-coloring such that:} \]

\[ ||C_i| - |C_j|| \leq 1 \quad \forall \ i, j = 1, \ldots, k \]

or equivalently:

\[ \lfloor n/k \rfloor \leq |C_j| \leq \lceil n/k \rceil \quad \forall \ j = 1, \ldots, k \]

Equitable Chromatic Number

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

- ECP consists of finding \( \chi_{eq}(G) \quad \leftarrow \quad \text{NP-Hard} \)
An exact DSatur-based algorithm for the Equitable Coloring Problem
Méndez-Díaz, Nasini, Severin

Introduction

Classic DSatur

Motivation

EqDSatur: An exact algorithm for the ECP

Computational experiments

Basic definitions

\[ G = (V, E) \quad V = \{1, \ldots, n\} \]

Classic Coloring

- \( k \)-coloring = partition of \( V \) into \( k \) non-empty stable sets
- \( C_1, C_2, \ldots, C_k \leftarrow \) color classes

Equitable Coloring

- \( k \)-eqcol = \( k \)-coloring such that:
  - \( ||C_i| - |C_j|| \leq 1 \quad \forall \ i, j = 1, \ldots, k \)

  or equivalently:
  - \( \lfloor n/k \rfloor \leq |C_j| \leq \lceil n/k \rceil \quad \forall \ j = 1, \ldots, k \)

Equitable Chromatic Number

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

- ECP consists of finding \( \chi_{eq}(G) \leftarrow \) NP-Hard
An example

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]
- B&C-LF₂ 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)]
DSatur Branch-and-Bound:

- In each subproblem, the graph is partially colored with \( k \) colors.
- It chooses an uncolored vertex \( u \) with the largest degree saturation.
- 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]
DSatur Branch-and-Bound:

- In each subproblem, the graph is partially colored with \( k \) colors.
- It chooses an uncolored vertex \( u \) with the largest degree saturation.
- 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 degree saturation.
- 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]
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]

 Expedient: 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]

 располагать для себя и своих детей в будущем.
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 $\rightarrow \bar{c}$  
  [Kubale, Furmanczyk 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 $\tilde{\alpha}(S) = \text{partition of } S \text{ in cliques, computed greedily}$
Initial bounds for the ECP

- **Upper bound**: Heuristic NAIVE $\rightarrow \bar{c}$
  
  [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:

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

where $\tilde{\alpha}(S) = \text{partition of } S \text{ in cliques, computed greedily}$
An exact DSatur-based algorithm for the Equitable Coloring Problem

Méndez-Díaz, Nasini, Severin

Introduction

Classic DSatur

Motivation

EqDSatur: An exact algorithm for the ECP

Computational experiments

Initial bounds for the ECP

- **Upper bound**: Heuristic \( \text{NAIVE} \rightarrow \bar{c} \)
  
  [Kubale, Furmańczyk 2005]

- **Lower bound**: Two ways
  
  - A maximal clique \( Q \) of \( G \) computed greedily
    
    [Lih, Chen 1994]

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

We propose:

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

where \( \tilde{\alpha}(S) = \) partition of \( S \) in cliques, computed greedily
An exact DSatur-based algorithm for the Equitable Coloring Problem

Méndez-Díaz, Nasini, Severin

Introduction

Classic DSatur

Motivation

EqDSatur: An exact algorithm for the ECP

Computational experiments

Initial bounds for the ECP

- **Upper bound**: Heuristic NAIVE $\longrightarrow \bar{c}$
  
  [Kubale, Furmańczyk 2005]

- **Lower bound**: Two ways
  - A maximal clique $Q$ of $G$ computed greedily
    - [Lih, Chen 1994]

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

  We propose:

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

  where $\tilde{\alpha}(S) =$ partition of $S$ in cliques, computed greedily
An exact DSatur-based algorithm for the Equitable Coloring Problem

Méndez-Díaz, Nasini, Severin

Introduction

Classic DSatur

Motivation

EqDSatur: An exact algorithm for the ECP

Computational experiments

Initial bounds for the ECP

- **Upper bound**: Heuristic NAIVE $\rightarrow \bar{c}$
  
  [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
Pruning 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 $G$
- $C_{k+1}, \ldots, C_n = \emptyset$
- $U = V \setminus \bigcup_{j=1}^k C_j$ (set of uncolored vertices)
- $F(u) = \{ j : \text{no vertex of } C_j \text{ is adjacent to } u \}$ $\forall u \in U$ (set of feasible colors of $u$)

If $Q = \text{maximal clique of } G$,

- $\Pi_Q = \text{partial coloring such that } Q \text{ is painted with colors } 1, 2, \ldots, |Q|$, 

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

$\Pi_Q$ suitable initial partial coloring
Pruning 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 $G$
- $C_{k+1}, \ldots, C_n = \emptyset$
- $U = V \setminus \bigcup_{j=1}^{k} C_j$ (set of uncolored vertices)
- $F(u) = \{j : \text{no vertex of } C_j \text{ is adjacent to } u\} \forall u \in U$ (set of feasible colors of $u$)

If $Q = \text{maximal clique of } G$,
- $\Pi_Q = \text{partial coloring such that } Q \text{ is painted with colors } 1, 2, \ldots, |Q|$, then $\Pi_Q$ can be extended to a $\chi_{eq}(G)$-eqcol.

$\Pi_Q$ suitable initial partial coloring
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 $G$
- $C_{k+1}, \ldots, C_n = \emptyset$
- $U = V \setminus \bigcup_{j=1}^{k} C_j$ (set of uncolored vertices)
- $F(u) = \{ j : \text{no vertex of } C_j \text{ is adjacent to } u \}$ $\forall u \in U$ (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.

ียว $\Pi_Q$ suitable initial partial coloring
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 $G$
- $C_{k+1}, \ldots, C_n = \emptyset$
- $U = V \setminus \bigcup_{j=1}^{k} C_j$ (set of uncolored vertices)
- $F(u) = \{j : \text{no vertex of } C_j \text{ is adjacent to } u\}$ $\forall u \in U$ (set of feasible colors of $u$)

If $Q = \text{maximal clique of } G$,

- $\Pi_Q = \text{partial coloring such that } Q \text{ is painted with colors } 1, 2, \ldots, |Q|$, then $\Pi_Q$ can be extended to a $\chi_{eq}(G)$-eqcol.

$\Pi_Q$ suitable initial partial coloring
We wonder when a partial coloring can be extended to an equitable coloring

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

**Lemma 1**

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

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

\[
(P.2) \quad M \leq \left\lfloor \frac{n}{\max\{k, LB\}} \right\rfloor^+
\]
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:

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

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

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

\[
(P.2) \quad M \leq \left\lfloor \frac{n}{\max\{k, LB\}} \right\rfloor \quad x^+ = \max\{x, 0\}
\]
Pruning 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:

\[
(P.1) \quad |U| \geq \sum_{t=1}^{k} \left( \max \left\{ M - 1, \left\lfloor \frac{n}{UB - 1} \right\rfloor \right\} - |C_t| \right)^+ \\
(P.2) \quad M \leq \left\lfloor \frac{n}{\max\{k, LB\}} \right\rfloor \\
x^+ = \max\{x, 0\}
\]
Pruning rules for the ECP

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

Equity constraint, \( r \geq k \rightarrow M^* \leq \left\lfloor \frac{n}{k} \right\rfloor 

Equity constraint, \( r \geq LB \rightarrow M^* \leq \left\lfloor \frac{n}{LB} \right\rfloor 

\Pi \text{ can be extended to } \Pi^*, \rightarrow M \leq M^* 

\therefore (P.2) \ M \leq \min \left\{ \left\lfloor \frac{n}{k} \right\rfloor, \left\lfloor \frac{n}{LB} \right\rfloor \right\} = \left\lfloor \frac{n}{\max\{k, LB\}} \right\rfloor
An exact DSatur-based algorithm for the Equitable Coloring Problem

Méndez-Díaz, Nasini, Severin

Introduction

Classic DSatur

Motivation

EqDSatur: An exact algorithm for the ECP

Computational experiments

Pruning rules for the ECP

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

\[
\text{Equity constraint, } r \geq k \implies M^* \leq \left\lfloor \frac{n}{k} \right\rfloor
\]

\[
\text{Equity constraint, } r \geq LB \implies M^* \leq \left\lfloor \frac{n}{LB} \right\rfloor
\]

\( \Pi \) can be extended to \( \Pi^* \), \( \implies M \leq M^* \)

\[
\therefore (P.2) \quad M \leq \min \left\{ \left\lfloor \frac{n}{k} \right\rfloor, \left\lfloor \frac{n}{LB} \right\rfloor \right\} = \left\lfloor \frac{n}{\max\{k, LB\}} \right\rfloor
\]
Pruning rules for the ECP

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

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

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

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

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

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

Equity constraint, \( r \geq k \) \( \rightarrow M^* \leq \left\lfloor \frac{n}{k} \right\rfloor \)

Equity constraint, \( r \geq \text{LB} \) \( \rightarrow M^* \leq \left\lfloor \frac{n}{\text{LB}} \right\rfloor \)

\( \Pi \) can be extended to \( \Pi^* \), \( \rightarrow M \leq M^* \)

\( \therefore (P.2) \quad M \leq \min \left\{ \left\lfloor \frac{n}{k} \right\rfloor, \left\lfloor \frac{n}{\text{LB}} \right\rfloor \right\} = \left\lfloor \frac{n}{\max\{k, \text{LB}\}} \right\rfloor \)
Pruning 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,

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

\( \Pi \) can be extended to \( \Pi^* \rightarrow M^* \geq M \rightarrow m^* \geq M - 1. \)

\[
\therefore m^* \geq \max\left\{ M - 1, \left\lfloor \frac{n}{UB - 1} \right\rfloor \right\}
\]
Pruning rules for the ECP

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

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

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

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

\[
\text{Equity constr., } r < UB \quad \rightarrow \quad m^* \geq \left\lfloor \frac{n}{UB - 1} \right\rfloor
\]

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

\[
\therefore \quad m^* \geq \max \left\{ M - 1, \left\lfloor \frac{n}{UB - 1} \right\rfloor \right\}
\]
Pruning 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., $r < UB \Rightarrow m^* \geq \left\lfloor \frac{n}{UB - 1} \right\rfloor$

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

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

\[ |U| \geq k \sum_{t=1}^{M} \left( \max\left\{ M - 1, \left\lfloor \frac{n}{UB - 1} \right\rfloor \right\} - |C_t| \right) + \ldots \]
Proof sketch (P.1) (cont.). Color classes from Π must have at least \( \max\left\{ M - 1, \left\lceil \frac{n}{UB - 1} \right\rceil \right\} \) vertices.

\[
\text{Needed vertices} = \sum_{t=1}^{k} \left( \max\left\{ M - 1, \left\lceil \frac{n}{UB - 1} \right\rceil \right\} - |C_t| \right)^+
\]
Pruning 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.

$$\therefore (P.1) \quad |U| \geq \sum_{t=1}^{k} \left( \max\left\{ M - 1, \left\lfloor \frac{n}{UB - 1} \right\rfloor \right\} - |C_t| \right)^+$$
Lemma 2

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

\[
U = \emptyset \rightarrow \Pi \text{ is a } k\text{-coloring}
\]

If \( M - 1 \geq \left\lfloor \frac{n}{UB - 1} \right\rfloor \)

\[
|U| = 0, \text{(P.1)} \rightarrow \sum_{t=1}^{k} (M - 1 - |C_t|)^+ = 0
\]

\( \rightarrow M - 1 \leq |C_t| \leq M \ \forall \ t \rightarrow \Pi \text{ is equitable} \)
Pruning 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.

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

If $M - 1 \geq \left\lceil \frac{n}{UB - 1} \right\rceil$

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

$\implies M - 1 \leq |C_t| \leq M \ \forall \ t \implies \Pi$ is equitable
Lemma 2

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

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

If $M - 1 \geq \left\lceil \frac{n}{UB - 1} \right\rceil$

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

$\rightarrow M - 1 \leq |C_t| \leq M \ \forall \ t \rightarrow \Pi$ is equitable
Lemma 2

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

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

If $M - 1 < \left\lceil \frac{n}{UB - 1} \right\rceil$

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

$\rightarrow \left\lceil \frac{n}{UB - 1} \right\rceil \leq |C_t| \leq M \ \forall \ t$
If \( \Pi \) is a partial \( k \)-coloring satisfying property (P.1) and \( U = \emptyset \) then \( \Pi \) is a \( k \)-eqcol.

\[
U = \emptyset \quad \rightarrow \quad \Pi \text{ is a } k\text{-coloring}
\]

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

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

\[
\rightarrow \quad \left\lfloor \frac{n}{UB - 1} \right\rfloor = |C_t| = M \quad \forall \, t \quad \rightarrow \quad \Pi \text{ is equitable}
\]

If we use (P.1) as a pruning rule, every time a coloring is reached at a leaf of the search tree, it is already equitable.
Lemma 2

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

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

If $M - 1 < \left\lceil \frac{n}{UB - 1} \right\rceil$

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

$\rightarrow \left\lceil \frac{n}{UB - 1} \right\rceil = |C_t| = M \ \forall \ t \rightarrow \Pi$ is equitable

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

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

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

- **Step 1.** If $U = \emptyset$, 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:
  - $\Pi' \leftarrow (\langle u, j \rangle \mapsto \Pi)$
  - If $F'(v) \cap \{1, \ldots, UB - 1\} \neq \emptyset \ \forall v \in U'$ 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

**Init.**: $G$ graph, $c$ $UB$-eqcol, $LB$ lower bound of $\chi_{eq}(G)$, $Q$ maximal clique of $G$.

**Node**($\Pi = (k, C_1, \ldots, C_n, U, F)$): (UB and $c$ global var.)

- **Step 1.** If $U = \emptyset$, set $UB \leftarrow k$, $c \leftarrow \Pi$ 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:

  $\Pi' \leftarrow (\langle u, j \rangle \leftrightarrow \Pi)$

  If $F'(v) \cap \{1, \ldots, UB - 1\} \neq \emptyset \ \forall v \in U'$ 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 $c$. 
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)]
- **\( EqDS_1 = EqDSatur \)**
- **\( EqDS_2 = EqDSatur \)** 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 \( j = i_2 \), \ldots, \( j = i_k \), \( j = k + 1 \).

**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)]
- **EqDS\(_1\)** = EqDSatur
  - **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 \(j = i_2\), \ldots, \(j = i_k\), \(j = k + 1\).

  - **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.
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)]
- **EQDS$_1$** = **EQDSatur**
- **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

- 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)

<table>
<thead>
<tr>
<th>%Density Vertices Graph</th>
<th>% solved inst. $LF_2$ EQDS$_1$ EQDS$_2$</th>
<th>% Rel. Gap. $LF_2$ EQDS$_1$ EQDS$_2$</th>
<th>Time $LF_2$ EQDS$_1$ EQDS$_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>100 100 100</td>
<td>0 0 0</td>
<td>109 0 0</td>
</tr>
<tr>
<td>30</td>
<td>0 100 100</td>
<td>18 0 0</td>
<td>— 0 0</td>
</tr>
<tr>
<td>50</td>
<td>0 100 100</td>
<td>8.2 0 0</td>
<td>— 16.2 16.1</td>
</tr>
<tr>
<td>70</td>
<td>100 100 100</td>
<td>0 0 0</td>
<td>273 31.2 32.1</td>
</tr>
<tr>
<td>90</td>
<td>100 100 100</td>
<td>0 0 0</td>
<td>10.4 0 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>%Density Vertices Graph</th>
<th>% solved inst. $LF_2$ EQDS$_1$ EQDS$_2$</th>
<th>% Rel. Gap. $LF_2$ EQDS$_1$ EQDS$_2$</th>
<th>Time $LF_2$ EQDS$_1$ EQDS$_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>100 100 100</td>
<td>0 0 0</td>
<td>5.7 0 0</td>
</tr>
<tr>
<td>30</td>
<td>0 100 100</td>
<td>20 0 0</td>
<td>— 23.7 17.6</td>
</tr>
<tr>
<td>50</td>
<td>0 100 100</td>
<td>24 0 0</td>
<td>— 477 424</td>
</tr>
<tr>
<td>70</td>
<td>10 70 70</td>
<td>16 10 10</td>
<td>5769 1715 1653</td>
</tr>
<tr>
<td>90</td>
<td>100 100 100</td>
<td>0 0 0</td>
<td>690 12.5 12.3</td>
</tr>
</tbody>
</table>

☞ EQDSatur highly competitive
☞ Ordering color classes seems to improve performance
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)

<table>
<thead>
<tr>
<th>%Density</th>
<th>% solved inst.</th>
<th>% Rel. Gap.</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vertices</td>
<td>Graph</td>
<td>$LF_2$</td>
<td>EqDS$_1$</td>
</tr>
<tr>
<td>70</td>
<td>10</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>70</td>
<td>30</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>70</td>
<td>50</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>70</td>
<td>70</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>70</td>
<td>90</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>80</td>
<td>10</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>80</td>
<td>30</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>80</td>
<td>50</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>80</td>
<td>70</td>
<td>10</td>
<td>70</td>
</tr>
<tr>
<td>80</td>
<td>90</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>

**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)

<table>
<thead>
<tr>
<th>%Density</th>
<th>% solved inst.</th>
<th>% Rel. Gap.</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$LF_2$</td>
<td>EQDS1</td>
<td>EQDS2</td>
</tr>
<tr>
<td>Vertices</td>
<td>Graph</td>
<td></td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>10</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>70</td>
<td>30</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>70</td>
<td>50</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>70</td>
<td>70</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>70</td>
<td>90</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>%Density</th>
<th>% solved inst.</th>
<th>% Rel. Gap.</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CPLEX</td>
<td>EQDS1</td>
<td>EQDS2</td>
</tr>
<tr>
<td>Vertices</td>
<td>Graph</td>
<td></td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>10</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>80</td>
<td>30</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>80</td>
<td>50</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>80</td>
<td>70</td>
<td>10</td>
<td>70</td>
</tr>
<tr>
<td>80</td>
<td>90</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>

**EQDSatur** highly competitive
- Ordering color classes seems to improve performance
**Computational experiments**

**DIMACS COLORLIB instances + Kneser graphs**

<table>
<thead>
<tr>
<th>Name</th>
<th>Vertices</th>
<th>Edges</th>
<th>$\chi_{eq}$</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CPLEX</td>
</tr>
<tr>
<td>1-FullIns_3</td>
<td>30</td>
<td>100</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>2-FullIns_3</td>
<td>52</td>
<td>201</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>anna</td>
<td>138</td>
<td>493</td>
<td>11</td>
<td>0</td>
</tr>
<tr>
<td>david</td>
<td>87</td>
<td>406</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td>games120</td>
<td>120</td>
<td>638</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td>jean</td>
<td>80</td>
<td>254</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>kneser5_2</td>
<td>10</td>
<td>15</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>kneser7_2</td>
<td>21</td>
<td>105</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>kneser7_3</td>
<td>35</td>
<td>70</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>kneser9_4</td>
<td>126</td>
<td>315</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>miles1500</td>
<td>128</td>
<td>5198</td>
<td>73</td>
<td>0</td>
</tr>
<tr>
<td>myciel3</td>
<td>11</td>
<td>20</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>myciel4</td>
<td>23</td>
<td>71</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>queen6_6</td>
<td>36</td>
<td>290</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>queen7_7</td>
<td>49</td>
<td>476</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>zeroin.i.1</td>
<td>211</td>
<td>4100</td>
<td>49</td>
<td>0</td>
</tr>
</tbody>
</table>
### Computational experiments

<table>
<thead>
<tr>
<th>Name</th>
<th>Vertices</th>
<th>Edges</th>
<th>$\chi_{eq}$</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CPLEX</td>
</tr>
<tr>
<td>3-FullIns_3</td>
<td>80</td>
<td>346</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>4-FullIns_3</td>
<td>114</td>
<td>541</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>5-FullIns_3</td>
<td>154</td>
<td>792</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>miles1000</td>
<td>128</td>
<td>3216</td>
<td>42</td>
<td>0</td>
</tr>
<tr>
<td>miles750</td>
<td>128</td>
<td>2113</td>
<td>31</td>
<td>0</td>
</tr>
<tr>
<td>queen8_8</td>
<td>64</td>
<td>728</td>
<td>9</td>
<td>654</td>
</tr>
<tr>
<td>zeroin.i.2</td>
<td>211</td>
<td>3541</td>
<td>36</td>
<td>2</td>
</tr>
<tr>
<td>zeroin.i.3</td>
<td>206</td>
<td>3540</td>
<td>36</td>
<td>5</td>
</tr>
<tr>
<td>1-Insertions_4</td>
<td>67</td>
<td>232</td>
<td>5</td>
<td>–</td>
</tr>
<tr>
<td>myciel5</td>
<td>47</td>
<td>236</td>
<td>6</td>
<td>149</td>
</tr>
<tr>
<td>queen8_12</td>
<td>96</td>
<td>1368</td>
<td>12</td>
<td>5</td>
</tr>
<tr>
<td>queen9_9</td>
<td>81</td>
<td>1056</td>
<td>10</td>
<td>–</td>
</tr>
<tr>
<td>will199GPIA</td>
<td>701</td>
<td>6772</td>
<td>7</td>
<td>–</td>
</tr>
<tr>
<td>ash331GPIA</td>
<td>662</td>
<td>4181</td>
<td>4</td>
<td>–</td>
</tr>
<tr>
<td>kneser11_5</td>
<td>462</td>
<td>1386</td>
<td>3</td>
<td>84</td>
</tr>
<tr>
<td>mulsol.i.1</td>
<td>197</td>
<td>3925</td>
<td>49</td>
<td>1</td>
</tr>
<tr>
<td><strong>Total solved instances</strong></td>
<td></td>
<td></td>
<td></td>
<td>28</td>
</tr>
</tbody>
</table>
### Computational experiments

<table>
<thead>
<tr>
<th>Name</th>
<th>Vertices</th>
<th>Edges</th>
<th>$\chi_{eq}$</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>CPLEX</td>
<td>$EQDS_1$</td>
</tr>
<tr>
<td>3-FullIns_3</td>
<td>80</td>
<td>346</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>4-FullIns_3</td>
<td>114</td>
<td>541</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>5-FullIns_3</td>
<td>154</td>
<td>792</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>miles1000</td>
<td>128</td>
<td>3216</td>
<td>42</td>
<td>0</td>
</tr>
<tr>
<td>miles750</td>
<td>128</td>
<td>2113</td>
<td>31</td>
<td>0</td>
</tr>
<tr>
<td>queen8_8</td>
<td>64</td>
<td>728</td>
<td>9</td>
<td>654</td>
</tr>
<tr>
<td>zeroin.i.2</td>
<td>211</td>
<td>3541</td>
<td>36</td>
<td>2</td>
</tr>
<tr>
<td>zeroin.i.3</td>
<td>206</td>
<td>3540</td>
<td>36</td>
<td>5</td>
</tr>
<tr>
<td>1-Insertions_4</td>
<td>67</td>
<td>232</td>
<td>5</td>
<td>–</td>
</tr>
<tr>
<td>myciel5</td>
<td>47</td>
<td>236</td>
<td>6</td>
<td>149</td>
</tr>
<tr>
<td>queen8_12</td>
<td>96</td>
<td>1368</td>
<td>12</td>
<td>5</td>
</tr>
<tr>
<td>queen9_9</td>
<td>81</td>
<td>1056</td>
<td>10</td>
<td>–</td>
</tr>
<tr>
<td>will199GPIA</td>
<td>701</td>
<td>6772</td>
<td>7</td>
<td>–</td>
</tr>
<tr>
<td>ash331GPIA</td>
<td>662</td>
<td>4181</td>
<td>4</td>
<td>–</td>
</tr>
<tr>
<td>kneser11_5</td>
<td>462</td>
<td>1386</td>
<td>3</td>
<td>84</td>
</tr>
<tr>
<td>mulsol.i.1</td>
<td>197</td>
<td>3925</td>
<td>49</td>
<td>1</td>
</tr>
<tr>
<td><strong>Total solved instances</strong></td>
<td></td>
<td></td>
<td>28</td>
<td>21</td>
</tr>
</tbody>
</table>
### Computational experiments

<table>
<thead>
<tr>
<th>Name</th>
<th>Vertices</th>
<th>Edges</th>
<th>$\chi_{eq}$</th>
<th>Time</th>
<th>CPLEX</th>
<th>EQDS₁</th>
<th>EQDS₂</th>
<th>LF₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-FullIns_3</td>
<td>80</td>
<td>346</td>
<td>6</td>
<td></td>
<td>0</td>
<td>–</td>
<td>–</td>
<td>85</td>
</tr>
<tr>
<td>4-FullIns_3</td>
<td>114</td>
<td>541</td>
<td>7</td>
<td></td>
<td>0</td>
<td>–</td>
<td>–</td>
<td>72</td>
</tr>
<tr>
<td>5-FullIns_3</td>
<td>154</td>
<td>792</td>
<td>8</td>
<td></td>
<td>0</td>
<td>–</td>
<td>–</td>
<td>268</td>
</tr>
<tr>
<td>miles1000</td>
<td>128</td>
<td>3216</td>
<td>42</td>
<td></td>
<td>0</td>
<td>–</td>
<td>0</td>
<td>267</td>
</tr>
<tr>
<td>miles750</td>
<td>128</td>
<td>2113</td>
<td>31</td>
<td></td>
<td>0</td>
<td>–</td>
<td>0</td>
<td>171</td>
</tr>
<tr>
<td><strong>queen8_8</strong></td>
<td>64</td>
<td>728</td>
<td>9</td>
<td>654</td>
<td>7.5</td>
<td>1.1</td>
<td>–</td>
<td>441</td>
</tr>
<tr>
<td>zeroIn.i.2</td>
<td>211</td>
<td>3541</td>
<td>36</td>
<td>2</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>510</td>
</tr>
<tr>
<td>zeroIn.i.3</td>
<td>206</td>
<td>3540</td>
<td>36</td>
<td>5</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>491</td>
</tr>
<tr>
<td>1-Insertions_4</td>
<td>67</td>
<td>232</td>
<td>5</td>
<td></td>
<td>–</td>
<td>1468</td>
<td>1540</td>
<td></td>
</tr>
<tr>
<td><strong>myciel5</strong></td>
<td>47</td>
<td>236</td>
<td>6</td>
<td>149</td>
<td>0</td>
<td>0</td>
<td>–</td>
<td></td>
</tr>
<tr>
<td>queen8_12</td>
<td>96</td>
<td>1368</td>
<td>12</td>
<td>5</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td></td>
</tr>
<tr>
<td><strong>queen9_9</strong></td>
<td>81</td>
<td>1056</td>
<td>10</td>
<td>–</td>
<td>719</td>
<td>578</td>
<td>–</td>
<td></td>
</tr>
<tr>
<td>will199GPIA</td>
<td>701</td>
<td>6772</td>
<td>7</td>
<td>–</td>
<td>–</td>
<td>2</td>
<td>–</td>
<td></td>
</tr>
<tr>
<td>ash331GPIA</td>
<td>662</td>
<td>4181</td>
<td>4</td>
<td>–</td>
<td>–</td>
<td>2</td>
<td>–</td>
<td></td>
</tr>
<tr>
<td>kneser11_5</td>
<td>462</td>
<td>1386</td>
<td>3</td>
<td>84</td>
<td>2973</td>
<td>–</td>
<td>–</td>
<td></td>
</tr>
<tr>
<td>mulsol.i.1</td>
<td>197</td>
<td>3925</td>
<td>49</td>
<td>1</td>
<td>–</td>
<td>0</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td><strong>Total solved instances</strong></td>
<td>28</td>
<td>21</td>
<td>25</td>
<td>28</td>
<td>21</td>
<td>25</td>
<td>24</td>
<td></td>
</tr>
</tbody>
</table>

Not bad for an enumerative scheme, right? 😊
## Computational experiments

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

Total solved instances: 28, 21, 25, $\geq$24.

Not bad for an enumerative scheme, right? 😊
An exact DSatur-based algorithm for the Equitable Coloring Problem

Méndez-Díaz, Nasini, Severin

Introduction

Classic DSatur

Motivation

EqDSatur: An exact algorithm for the ECP

Computational experiments

Thanks!