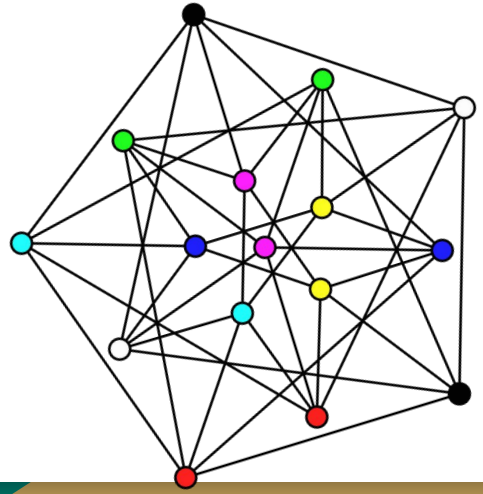
A LINEAR INTEGER PROGRAMMING APPROACH FOR THE EQUITABLE COLORING PROBLEM



This survey was made by:
Apachiței Maria-Luisa
Munteanu Rareș-Costin
Vîrvarei Alexandru

Table of contents

EQUITABLE COLORING PROBLEM

→ $G = (V, E)$ - undirected graph $V = 1, \ldots, n$
→ $E = \{(u, v) | (u, v) \in V^2 \text{ and } u \neq v\}$
→ $C_j \overset{\text{not}}{=}$ the set of vertices painted with color j, $\forall j = 1, \ldots, n$
→ $k - eqcol \overset{\text{not}}{=}$ a k-coloring of G that satisfies the equity constraints:

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

→ A k-coloring is a k-eqcol if and only if:

$$\left\lfloor \frac{n}{k} \right\rfloor \leq |C_j| \leq \left\lceil \frac{n}{k} \right\rceil, \forall j = 1, \ldots, k$$

→ $\chi_{eq}(G) \overset{\text{not}}{=}$ minimum k for which there exists a k-eqcol in$G$

ECP

1

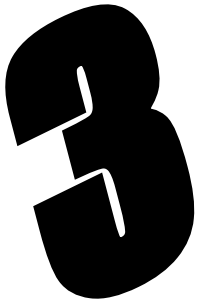
2 PAPER's BASIS and PURPOSE

- algorithms based on **linear integer programming (IP)** have proved great results in solving the traditional coloring problem
- CP -> ECP by adding extra inequalities
- the **Branch-And-Cut algorithm** that's solving CP will be reused for solving ECP

PURPOSE

1. Adapt the CP-model which have proved good performance when using B&C
2. Choose the model with best behavior

MODELS for ECP



The research paper proposes 3 possible linear representations for ECP, named as **M1, M2 and M3**. These have the same basis but the way of modelling the equity constraints is changed. The differences are leaded by the desire of *reducing the number of symmetric feasible solutions* (in our problem: the *symmetrical colorings*).

Let us show and explain each of them…

M1

Notations for $\forall v \in \{1,\dots,n\}$ and $\forall j \in \{1,\dots,n\}$:
→ $x_{vj} = 1$ if and only if color j is assigned to vertex v
→ $w_j = 1$ if and only if color j is used by some vertex

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

minimize the number of colors

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

each vertex has to be painted by an unique color

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

two adjacent vertices can not share the same color

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

use color j + 1 iff color j is used

$$x_{ij} \leq w_j \qquad \forall i \in I, j = 1,\dots,n$$

no isolated vertex can be painted with color j that is NOT used

$$\sum_{v=1}^{n} x_{vj} = w_n + \sum_{k=j}^{n-1} t_k^j (w_k - w_{k+1}) \qquad \forall j = 1,\dots,n-1$$

$$x_{vj}, w_j \in \{0,1\} \qquad \forall v \in V, j = 1,\dots n$$

binary variables

$t_k^j = |C_j|$ with $t_k^j = \left\lfloor \frac{n}{k} \right\rfloor + 1$ if $j \leq p$ and $t_k^j = \left\lceil \frac{n}{k} \right\rceil$ if $p < j \leq k$ and p = n mod k

If we restrict symmetric colorings in a k-eqcol by adding card(j) ≥ card(j+1) and we define ***p = n mod k****, then the k-eqcol has exactly* ***p*** *classes of size* ***⌊n/k⌋ + 1*** and **k-p** classes of size ***⌊n/k⌋*** *(in this order)*

M2

Notations for $\forall v \in \{1,\dots,n\}$ and $\forall j \in \{1,\dots,n\}$:
→ $x_{vj} = 1$ if and only if color j is assigned to vertex v
→ $w_j = 1$ if and only if color j is used by some vertex

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

minimize the number of colors

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

each vertex has to be painted by an unique color

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

two adjacent vertices can not share the same color

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

use color j + 1 iff color j is used

$$x_{ij} \leq w_j \qquad \forall i \in I, j = 1,\dots,n$$

no isolated vertex can be painted with color j that is NOT used

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

To avoid symmetry: vertex v should be painted by a color j iff ***j ≤ v***

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

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

By adding the previous constraint, we need to remodel the condition of equity presented in ***M1*** *by rewriting the* ***k-eqcol condition****:*

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

$$x_{vj}, w_j \in \{0,1\} \qquad \forall v \in V, j = 1,\dots n$$

binary variables

M3

Notations for $\forall v \in \{1,\dots,n\}$ and $\forall j \in \{1,\dots,n\}$:
→ $x_{vj} = 1$ if and only if color j is assigned to vertex v
→ $w_j = 1$ if and only if color j is used by some vertex

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

minimize the number of colors

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

each vertex has to be painted by an unique color

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

two adjacent vertices can not share the same color

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

use color j + 1 only if color j is used

$$x_{ij} \leq w_j \qquad \forall i \in I, j = 1,\dots,n$$

no isolated vertex can be painted with color j that is NOT used

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

*To avoid symmetry: vertex v should be painted by a color j iff **j ≤ v***

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

***∀ j = 1, … , k,** strengthen the **k-eqcol condition**:*

$$\lfloor n/k \rfloor \leq |C\_j| \leq \lceil n/k \rceil$$

by the disjunction:

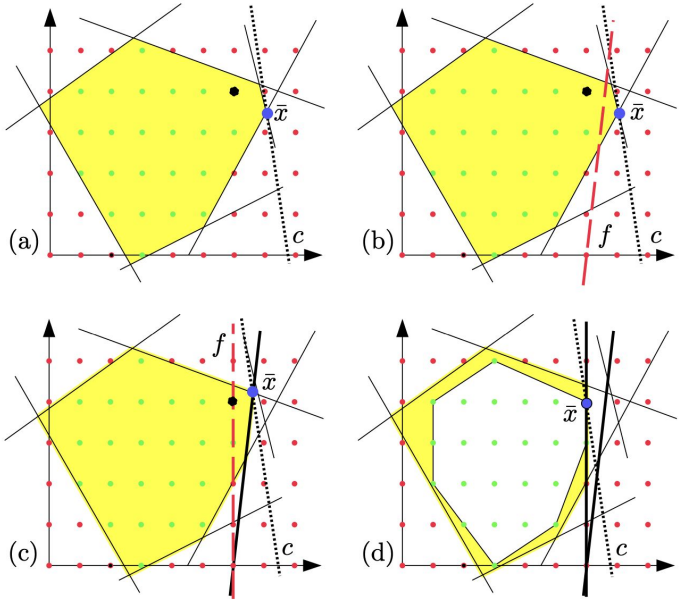
$$\lfloor n/k \rfloor = |C\_j| \vee |C\_j| = \lceil n/k \rceil$$

*and rewrite the equity condition from **M2** by adding the binary variable **y***

$$x_{vj}, w_j, y_j \in \{0,1\} \qquad \forall v \in V, j = 1,\dots n$$

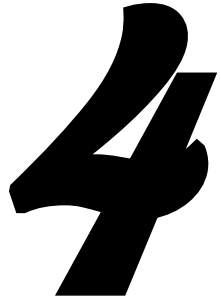
binary variables

Branch and cut algorithm

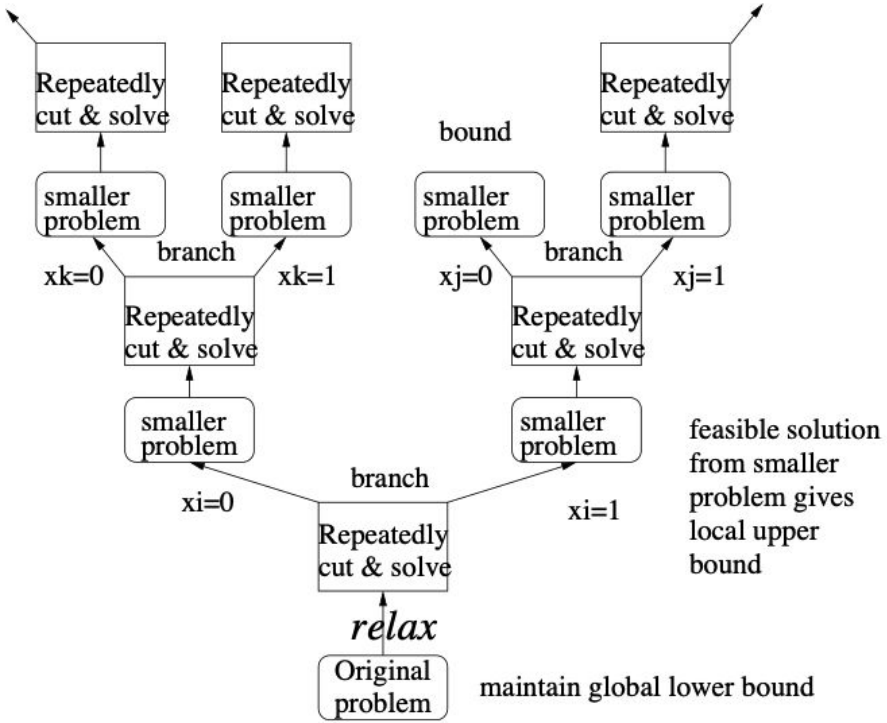


source: Freie Universität Berlin

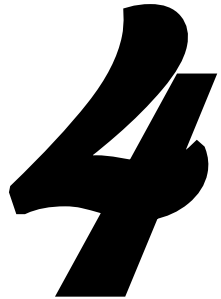
Given an LP, consider the yellow polyhedron (a) is the feasible region for the problem. In this case, the feasible region for the ILP is the convex hull of the green points inside the yellow polyhedron, forming the white polyhedron (d).
If the solution $\bar{x}$ of the LP-relaxation is integral, it corresponds to a feasible incidence vector that represents an optimal solution. Otherwise, we search for a valid inequality $f_x < f_0$ that "cuts off" the solution $\bar{x}$, i.e "a cutting plane"



Branch and cut algorithm - general structure



source: Freie Universität Berlin



RESULTS

5

- we’ve performed tests by **running all the models** proposed by the reference paper
- we will refer to these implementations and their results as **B&C_M1, B&C_M2 and B&C_M3** (meaning: applying Branch and Cut to solve the M_i algorithm)
- the instances and the target values are collected from the following source: Graph coloring Instances
- the comparison between these three models will be made based on the **Number of Evaluated Nodes** and the **Time of Evaluation** (presented in seconds)

Results

5

Name	n	χeq	Evaluated nodes			Time in sec.		
			B&C-M1	B&C-M2	B&C-M3	B&C-M1	B&C-M2	B&C-M3
queen6-6	36	7	46	0	0	13	2	3
queen7-7	49	7	0	0	0	23	10	20
myciel3	11	4	2	0	0	0	0	0
myciel4	23	5	722	14	24	4	1	1
jean	80	10	4	0	0	117	9	13
1-FullIns-3	30	4	2	0	0	3	1	1
2-FullIns-3	52	5	60	16	66	36	5	9

Table 1: Results for each problem : B&C-M1, B&C-M2, B&C-M3

RESULTS

5

The research paper did the same test on random generated instances and they concluded the same thing that can be appreciated from our records too: **B&C-M3 and B&C_M2 implementations outperforms B&C-M1 both in nodes in time** (this is because a larger number of nodes implies a larger number of subproblems to be resolve, so much CPU time allocated for processing). But the more general tests performed on random instances of various sizes guide us to choose **B&C-M3**. (assuming a little risk of a larger computational time in some cases)

So we can now do comparisons between our results (note: **B&C_M3_p our implementation**) and the ones recorded by our research paper and their reference paper (the alternative implementation is noted as **B&C_LF2**).

6

Comparison

Name	n	χeq	Evaluated nodes			Time in sec.		
			B&C-M3	B&C-LF2	B&C-M3$_p$	B&C-M3	B&C-LF2	B&C-M3$_p$
queen6-6	36	7	205	1	0	13	1	3
queen7-7	49	7	2	1	0	4	0	20
myciel3	11	4	0	7	0	0	0	0
myciel4	23	5	179	237	24	0	5	1
jean	80	10	0	1	0	0	4	13
1-FullIns-3	30	4	0	34	0	0	2	1
2-FullIns-3	52	5	0	84	66	0	25	9

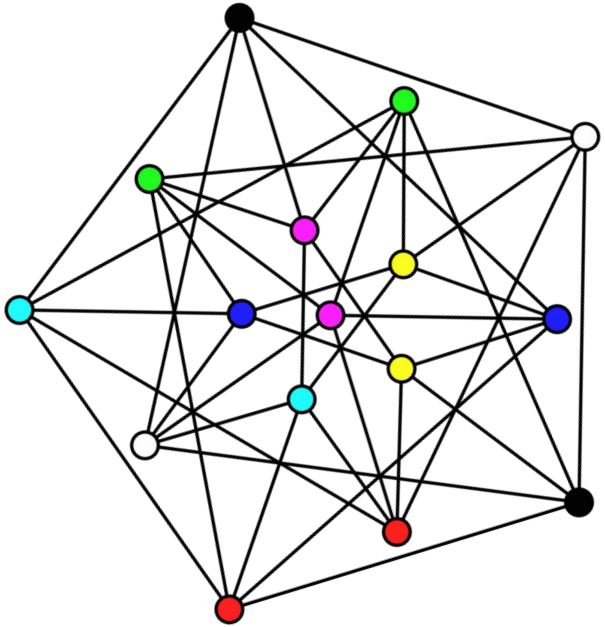
Table 2: Comparing B&C-M3, B&C-LF2 with B&C-M3$_p$

6

Comparison

The previous table shows that both B&C-M3 and B&C_M3p evaluates fewer nodes than B&C-LF2, and consume less time.

Overmore, there is a case where B&C_M3 underperforms regarding time and the evaluated nodes the B&C_LF2 model - *queen6-6* - but our B&C_M3p remains stable.



Hope we attracted interest on our subject!
Thank you for attention!