

Matheuristic Variants of DSATUR for the Vertex Coloring Problem

Federico Bruzzone,¹ PhD Candidate

Milan, Italy – 4 December 2025



¹ADAPT Lab – Università degli Studi di Milano,

Website: federicobruzzone.github.io,

Github: github.com/FedericoBruzzone,

Email: federico.bruzzone@unimi.it

Slides: TODO

Notation

- $G = (V, E)$ is an undirected graph with vertex set $V = \{v_1, \dots, v_n\}$ and edge set E , and denote $I = \llbracket 1; n \rrbracket = [1; n] \cap \mathbb{Z}$.
- An edge $e = (v_i, v_j) \in E$ with $i < j$ links the underlying vertices (*for VCP there is no sense to consider loop/multiple edges*).
- For $i \in I$, δ_i denotes the set of neighbors of vertex v_i , and $d_i = |\delta_i| = |\{j \in I \mid \text{ngb}(v_i, v_j) = 1\}|$, where $\text{ngb}(v_i, v_j) = 1$ iff $(v_i, v_j) \in E$.²

²ngb stands for “neighbor”.

A k -coloring of G

- It is an assignment of colors to vertices such that no two adjacent vertices share the same color.
- **VCP** consists in finding a k -coloring of G using the minimum number of colors k (the *chromatic number* $\chi(G)$).
- A valid k -coloring (c) fulfills: $\forall i < j, (v_i, v_j) \in E \implies c_i \neq c_j$ where $c_i = \llbracket 1; k \rrbracket$ is the color of v_i .

A partial k -coloring of G

- If a vertex may not be colored, we set $c_i = -1$ s.t. $c_i \in \llbracket 1; k \rrbracket \cup \{-1\}$
- A partial k -coloring (c) is *feasible* if $\forall i < j, (v_i, v_j) \in E \implies c_i \neq c_j \vee c_i = c_j = -1$.
- Given v_i the **saturation table** S_i is the set of colors assigned to its colored neighbors: $S_i = \bigcup_{j \in \delta_i}^n \{c_j\} \setminus \{-1\}$, and $s_i = |S_i|$ is the **saturation degree**.
- A total order \succsim over V is defined as: $v_i \succsim v_j \iff s_i > s_j \vee (s_i = s_j \wedge d_i \geq d_j)$

Compact ILP Formulations³, feasible *iff* $\chi(G) \leq k$

$z_{i,c} \in \{0, 1\}$ indicates if vertex v_i is assigned color c .	$\min \sum_{c=1}^k y_c$	The objective minimizes the number of used colors.
$y_c \in \{0, 1\}$ indicates if color c is used in the coloring.	$s.t. \min \sum_{c=1}^k z_{i,c} = 1 \quad \forall i \in I$	The 1st set ensures that each vertex is assigned exactly one color.
	$z_{i,c} + z_{j,c} \leq y_c \quad \forall (v_i, v_j) \in E,$ $\forall c \in \llbracket 1; k \rrbracket$	The 2nd set ensures that adjacent vertices do not share the same color.

³Efficient formulations: extended column generation by Furini and Malaguti [1], and reduced formulation to MWSSP by Cornaz et al. [2].

Observations

- Having an upper bound of the chromatic number as the initial value k (or simply $k = |V|$) guarantees the optimality of the solution.
- The size of k strongly affects the performance of ILP solvers.
- Symmetries in the model (e.g., colors are permutable) enlarge the search space for Branch-and-Bound algorithms (the same solution can be represented in multiple ways)

Representative ILP Model⁴, asymmetric and easily to LP-relax

$$\begin{array}{ll}
 x_{i,i'} \in \{0,1\}, & \\
 \forall i, i' \in V & \\
 s.t. i \leq i', & \\
 \text{indicates if} & \\
 \text{vertices } v_i \text{ and} & \\
 v_{i'} \text{ share the} & \\
 \text{same color and} & \\
 i \text{ is the} & \\
 \text{minimum} & \\
 \text{index of its} & \\
 \text{color class.} &
 \end{array}
 \begin{array}{ll}
 \min_z \sum_{i=1}^n x_{i,i} & \\
 s.t. \sum_{i' \leq i} x_{i',i} \geq 1 & \forall i \in I \\
 x_{j,i} + x_{i,j} \leq x_{j,j} & \forall (v_i, v_j) \in E, \\
 & \forall i < j \in I
 \end{array}$$

The objective counts the number of representative vertices (i.e., used colors).

The 1st set ensures either $x_{i,i} = 1$ or v_i shares the color with a vertex $v_{i'}$ with $i' < i$.

The 2nd set expresses the color incompatibility between adjacent vertices and $x_{j,i} = 1$ implies that $x_{j,j} = 1$

⁴A vertex is representative of its color class if it has the minimum index among the vertices sharing the same color.

Standard DSATUR Algorithm

Algorithm 1: Standard DSATUR algorithm

Input: $G = (V, E)$ a non-empty and non-oriented graph

Initialization:

define partial coloring c with $c_i := -1$ for all $i \in I$

define saturation table S with $S_i := \emptyset$ for all $i \in I$

initialize set $U := V$, and color $k := 0$

while $U \neq \emptyset$

find $u \in U$, a maximum of \succsim in U .

if $|S_u| = k$ **then** $k := k + 1$ // a new color is added

compute $c_i := \min S_u$ // assign color to u

 remove u from U

for all $i \in \delta_u \cap U$, $S_i = S_i \cup \{c_i\}$ // update saturation

end while

return color k and (c) a k -coloring of G

- DSATUR is an **adaptive greedy heuristic** proposed by Brélaz [3], which colors vertices iteratively.
- Selection of the uncolored vertex to color is given with order \succsim , maximizing first the saturation degree and secondly the degree.
- Coloring a new vertex updates saturation, the iteration order of vertices is thus adaptive.

DASTUR Matheuristic Variants⁵

⁵N. Dupin, “Matheuristic Variants of DSATUR for the Vertex Coloring Problem,” in *Metaheuristics 2024* [[4](#)]

Initialization

Defining an initial partial coloring and computing the saturation table for the uncolored vertices, **before** starting the main DSATUR iterations.

Variants:

1. maxDeg: color the vertex with the maximum degree — equivalent to standard DSATUR by definition of \succ , it would suffer from many ties;
2. col- n : consider n vertices having the maximum degree and color them solving a representative ILP model for the *induced* subgraph — more depth pre-processing, it tries to prevent erroneous decisions in the initial steps of DSATUR;
3. clq: find a maximum clique⁶ and color it with different colors — an exact pre-processing (not heuristic), it leads to a better initial saturation table S for the uncolored vertices;
4. clq-col- n : combine clq and col- n — best of both worlds.

⁶It is NP-hard, an heuristic can be used.

Local Optimization with Larger Neighborhoods

Now, (c) is a partial k -coloring.

Thank You!

Bibliography

- [1] F. Furini and E. Malaguti, “Exact weighted vertex coloring via branch-and-price,” *Discrete Optimization*, vol. 9, no. 2, pp. 130–136, 2012.
- [2] D. Cornaz, F. Furini, and E. Malaguti, “Solving vertex coloring problems as maximum weight stable set problems,” *Discrete Applied Mathematics*, vol. 217, pp. 151–162, 2017.
- [3] D. Brélaz, “New methods to color the vertices of a graph,” *Commun. ACM*, vol. 22, no. 4, pp. 251–256, Apr. 1979.
- [4] N. Dupin, “Matheuristic Variants of DSATUR for the Vertex Coloring Problem,” in *Metaheuristics*, M. Sevaux, A.-L. Olteanu, E. G. Pardo, A. Sifaleras, and S. Makboul, Eds., Cham: Springer Nature Switzerland, 2024, pp. 96–111.