

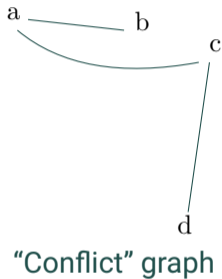
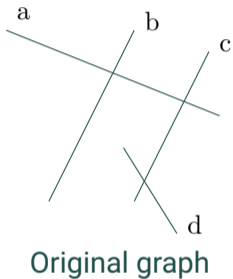
# LOCAL SEARCH WITH WEIGHTING SCHEMES (CG:SHOP 2022 COMPETITION)

---

Florian Fontan, Pascal Lafourcade, Luc Libralesso, and Benjamin Momège  
<libralesso.l@gmail.com>

CP 2021    October 2020

# Vertex coloring problem



Only solving the vertex coloring problem

# Local search for the vertex coloring problem

- ▶ Popular approach
- ▶ In many SoTA approaches

## Outline

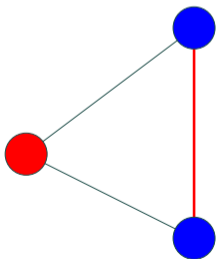
1. Find a starting solution (greedy)
2. Remove a color (become infeasible)
3. Perturbations to restore feasibility
4. *Repeat*

# Local search schemes

## Conflict-based

- ▶ Every vertex is colored
- ▶ Conflicting edges allowed

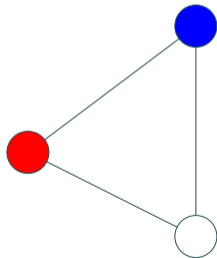
**min. nb of conflicting edges**



## Partial-coloring-based

- ▶ No conflicting edges
- ▶ Some vertices uncolored

**min. nb of uncolored vertices**



# Weighting schemes

Local search can loop over some states. How to avoid this?

## Rationale

- ▶ Penalize often seen conflicts / uncolored vertices.
- ▶ Minimize sum of weights instead of the number.

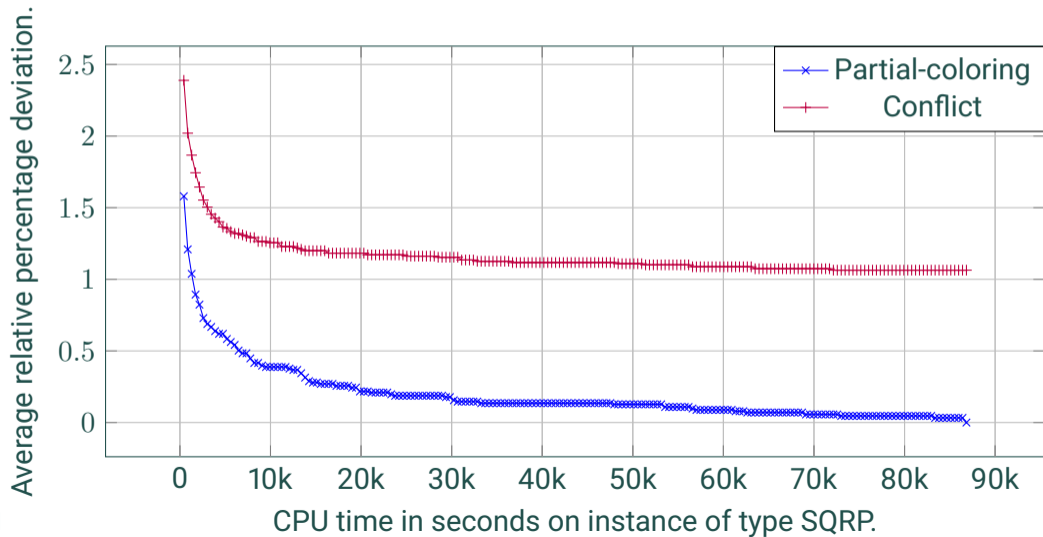
## Conflict-based

- ▶ Weights on edges
- ▶ +1 each time its endpoints conflict

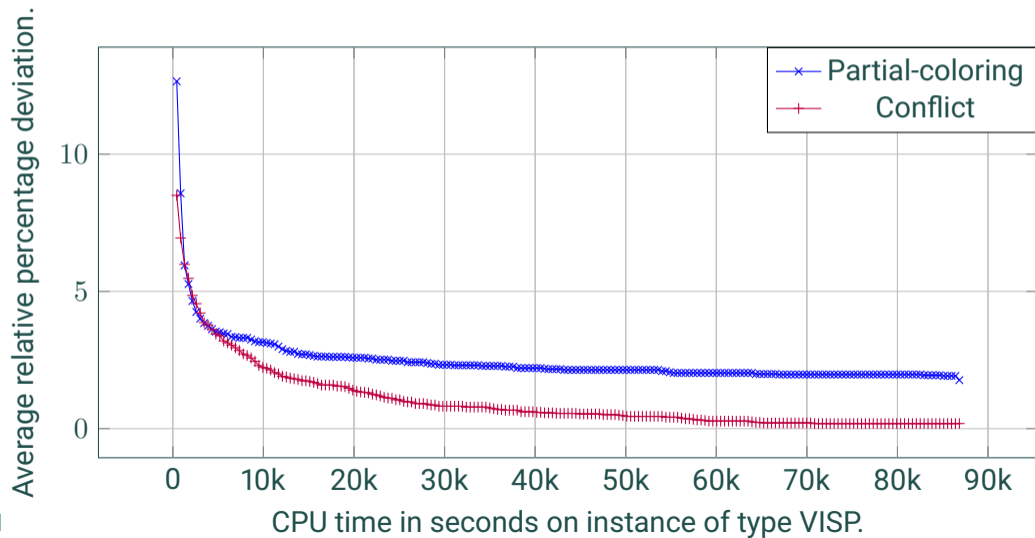
## Partial-coloring-based

- ▶ Weights on vertices
- ▶ +1 each time it is uncolored

# Partial-coloring based wins...



# But not always



source code: <https://github.com/librallu/dogs-color>

# LOCAL SEARCH WITH WEIGHTING SCHEMES (CG:SHOP 2022 COMPETITION)

---

Florian Fontan, Pascal Lafourcade, [Luc Libralesso](mailto:libralesso.l@gmail.com), and Benjamin Momège  
<libralesso.l@gmail.com>

CP 2021    October 2020



# Results (ARPD)

Instance class	conflict	partial-coloring	DSATUR	orientation greedy
reecn	1.04	<b>0.0</b>	24.31	16.41
rsqrp	0.51	0.11	24.03	5.89
rsqrpecn	0.0	0.0	24.86	18.08
rvisp	<b>0.06</b>	0.99	25.06	26.94
rvispecn	0.0	0.14	30.9	49.93
sqrp	1.06	<b>0.0</b>	31.22	4.9
sqrpecn	0.88	<b>0.03</b>	22.74	14.11
visp	<b>0.19</b>	1.77	29.23	27.6
vispecn	0.49	<b>0.08</b>	34.75	54.66

Table 1: Average relative percentage deviation to the best solutions we obtained