New Bounds and Constraint Programming Models for the Weighted Vertex Coloring Problem

Olivier Goudet, Cyril Grelier, David Lesaint

2023

LERIA

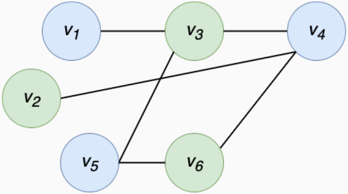
université angers

Sommaire

Problem definition and related work

GCP - Graph Coloring Problem

- Given an undirected graph $G = (V, E)$, a (legal) coloring is a partition $\{V_1, \ldots, V_k\}$ of V into independent sets.
- Find a coloring $s = \{V_1, \ldots, V_k\}$ of G of minimum size k.

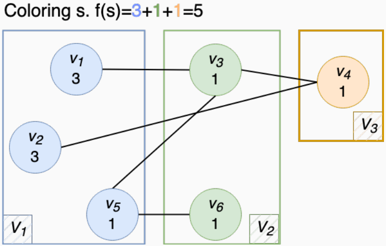


$\chi_G = 2$

- χ_G is the chromatic number of G.

WVCP - Weighted Vertex Coloring Problem

- A WVCP instance is defined by a vertex-weighted graph (G, w) :
  - $G = (V, E)$ is an undirected graph.
  - $w : V \mapsto \mathbb{N}^*$ is the weight function.
- Find a coloring $s = \{V_1, \ldots, V_k\}$ of G with minimum score $f(s) = \sum_{i=1}^{k} \max_{v \in V_i} w(v)$.



- WVCP is NP-hard (GCP is WVCP when w is constant).
- Applications : batch scheduling, traffic assignment for satellite communications, matrix decomposition.

State of the art on WVCP

Exact methods

2-Phase	Generation of independent sets and optimization	[Malaguti et al., 2009]
MWSS	Mixed Integer Linear Programming	[Cornaz et al., 2017]

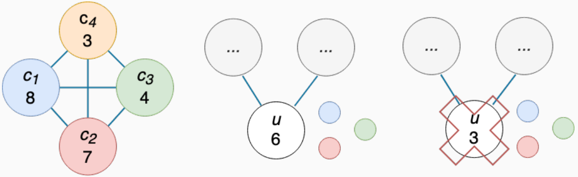
Metaheuristics / Local Search (LS)

R-GRASP	Iterated greedy algorithm with LS	[Prais and Ribeiro, 2000]
AFISA	LS with adaptive management of weights	[Sun et al., 2018]
RedLS	Reduction and LS with weights on edges	[Wang et al., 2020]
ILS-TS	Iterated LS with grenade operator	[Nogueira et al., 2021]
DLMCOL	Memetic algorithm with deep learning	[Goudet et al., 2022]
MCTS	Monte Carlo Tree Search with LS	[Grelier et al., 2022] [Grelier et al., 2023]

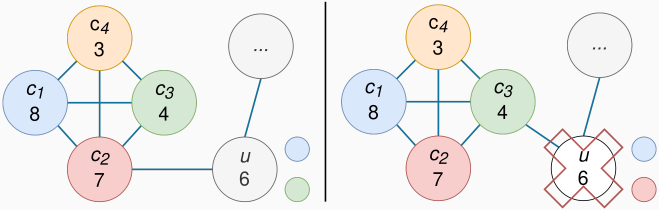
Vertex reduction rules and iterative reduction procedure

First vertex reduction rule (R1)

- Rule R0 of [Wang et al., 2020] : given a clique $C = \{c_1, \ldots, c_n\}$ with $w(c_i) \geq w(c_{i+1})$, removes $u \notin C$ if $w(c_{\Delta(u)+1}) > w(u)$.

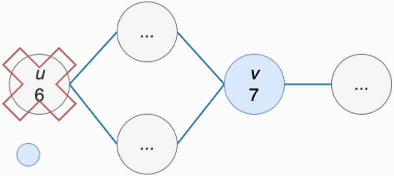


- Our rule R1 takes into account that u may have neighbors in C.

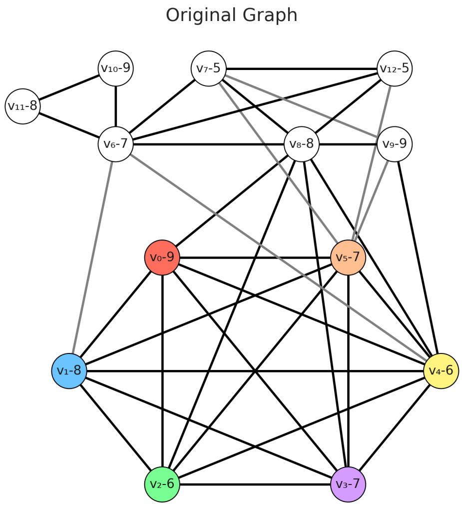


Second vertex reduction rule (R2)

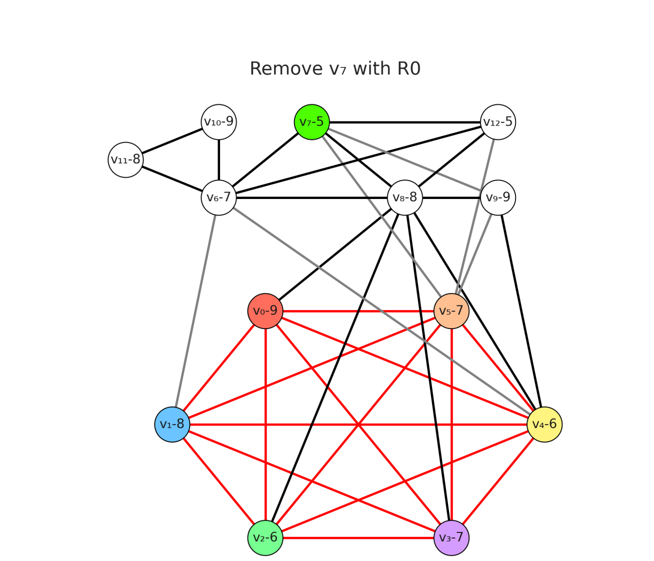
- Our rule R2 adapts an operator of [Cheeseman et al., 1991] for GCP.
  - Removes vertex u if its neighborhood is included in the neighborhood of a non-adjacent and no lighter vertex v ($N(u) \subset N(v)$ and $w(u) \leq w(v)$).



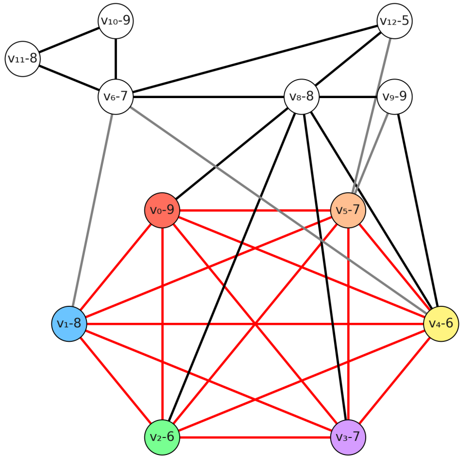
Iterative reduction rule



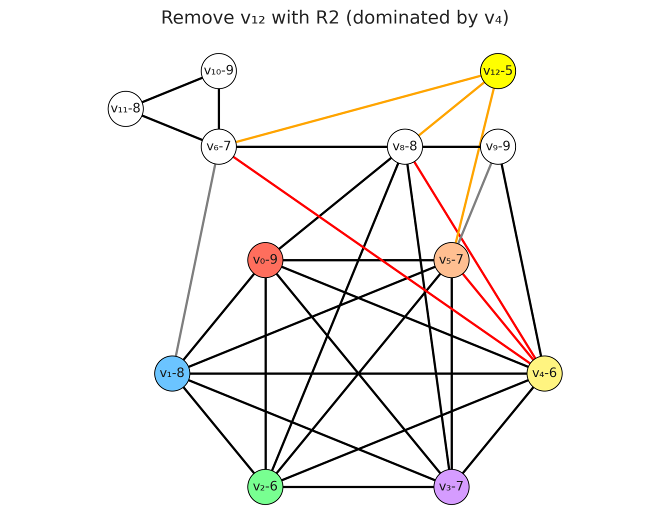
Iterative reduction rule



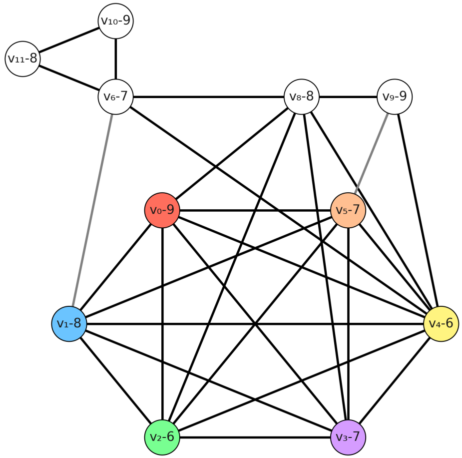
Iterative reduction rule



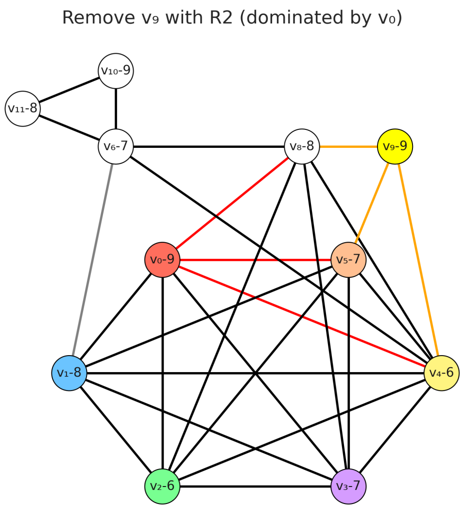
Iterative reduction rule



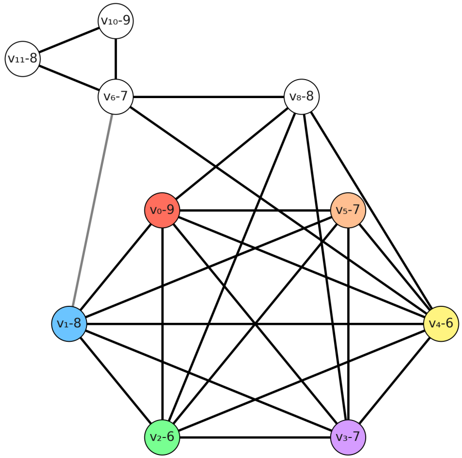
Iterative reduction rule



Iterative reduction rule

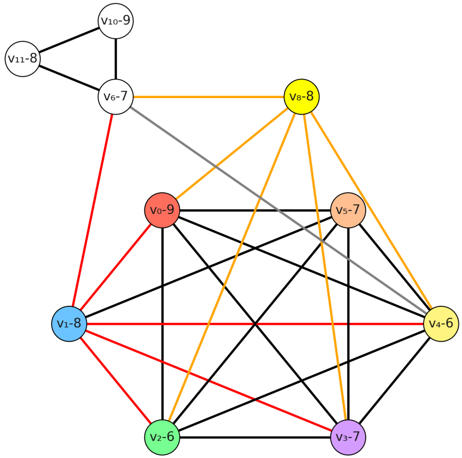


Iterative reduction rule

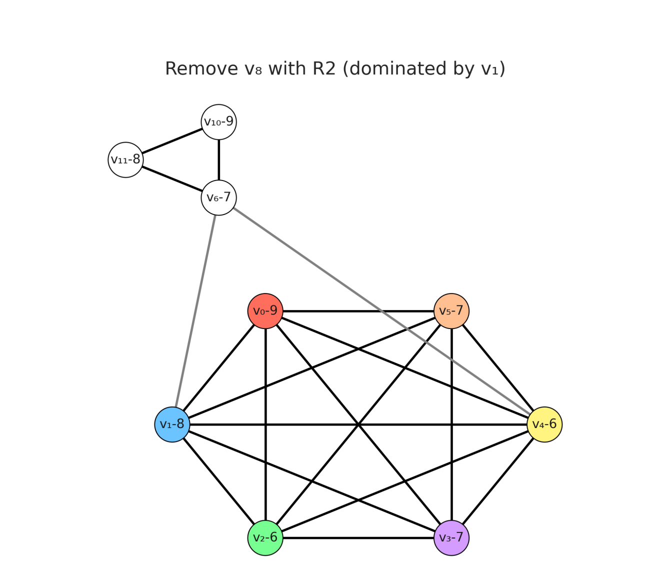


Iterative reduction rule

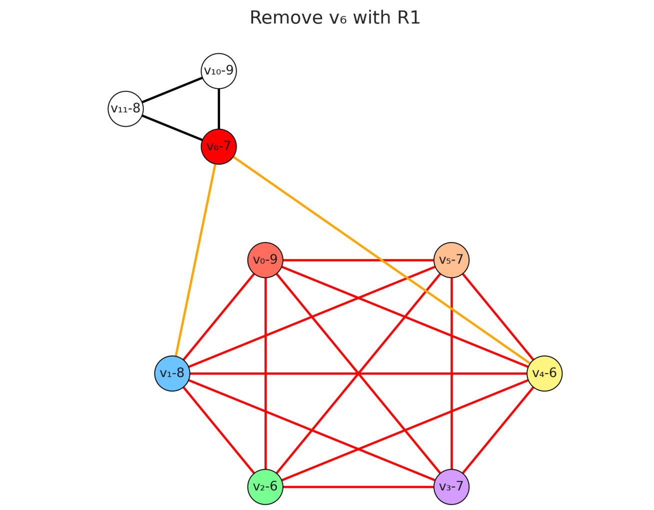
Remove v_8 with R2 (dominated by v_1)



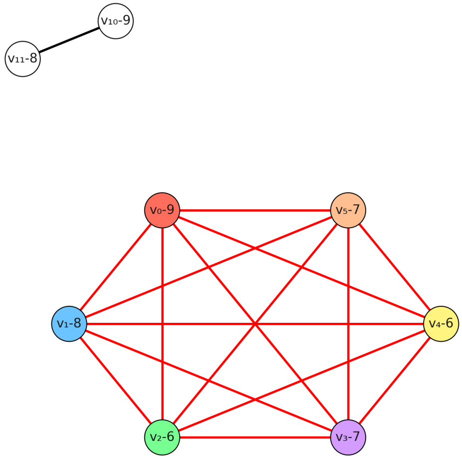
Iterative reduction rule



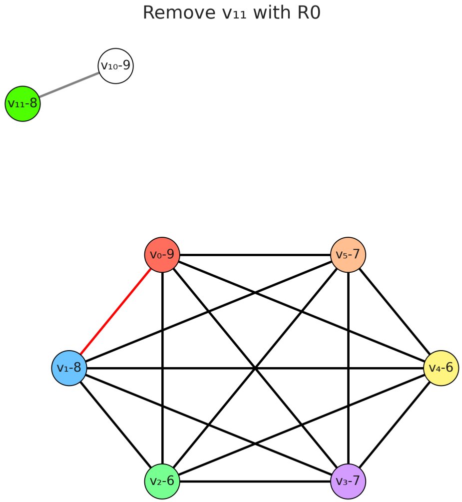
Iterative reduction rule



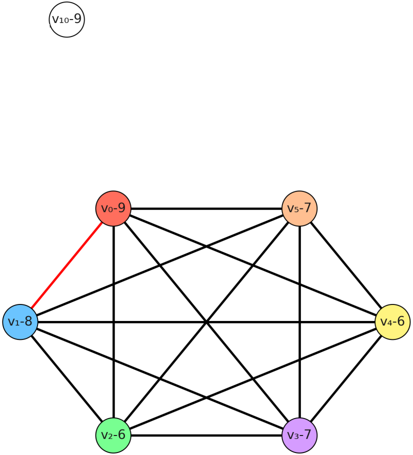
Iterative reduction rule



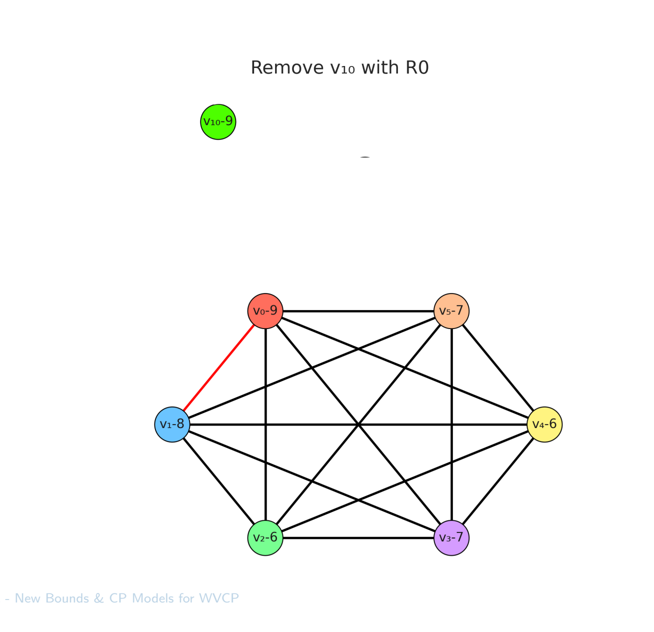
Iterative reduction rule



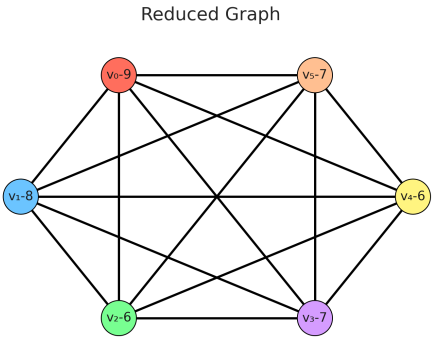
Iterative reduction rule



Iterative reduction rule



Iterative reduction rule



Impact of reduction rules on benchmark instances

Experiments on 188 instances

- $\#I$: number of reduced instances
- $\%V$: % of removed vertices
- $t(s)$: average run time in seconds

	# I	$\%V$ avg	$\%V$ max	t(s)
R0	82	13.4	65	2.6
R1	84	14.7	66.4	3.8
R1+R2	85	15.4	69	4.1
Iterative	85	23.3	80.9	9.8

instance	$\|V\|$	density	R0	R1	R1+R2	Iterated	time(s)
DSJC125.5g	125	0.5	0	0	0	0	0.26
DSJR500.1	500	0.03	78	80	80	**256**	1.32
GEOM110	110	0.11	6	9	9	**23**	0.09
inithx.i.1	864	0.05	469	574	596	**683**	19.45
le450_25b	450	0.08	90	90	90	**105**	2.26
mulsol.i.5	186	0.23	28	53	75	**82**	1.16
queen10_10	100	0.59	0	0	0	0	0.08
p42	138	0.12	1	1	1	**3**	0.1
r30	301	0.09	0	0	0	0	0.48

Upper bound on the number of colors

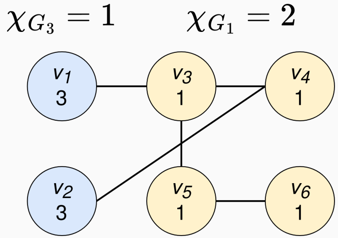
Upper bound on the number of colors for WVCP

- Default upper bound : $k \leq |V|$.
- [Demange et al., 2007] : $k \leq \Delta(G) + 1$.
- Interesting for reducing search space.

New upper bound on number of colors for WVCP

$W = \{w(v) \mid v \in V\}$ is the set of weights used in $P = (G, w)$.
$G_w = (V_w, E_w)$ is the subgraph of G induced by weight $w \in W$.

Theorem. Let $s^* = \{V_1, \ldots, V_k\}$ be an optimal solution to P. Then $k \leq \sum_{w \in W} \chi_{G_w}$.



$k \leq \chi_{G_1} + \chi_{G_3} = 3$

- Requires computing each chromatic number χ_{G_w}.
- χ_{G_w} may be approximated using GCP heuristics such as TabuCol [Hertz and de Werra, 1987].

Lower and upper bounds on score and colors

Color bounds

lb : max. clique size using FastWClq

ub : $min(\sum_{\mathbf{w} \in \mathbf{W}} \chi_{\mathbf{G_w}}, \Delta + 1)$

Score bounds

lb : method of [Wang et al., 2020]

ub : $\sum_{\mathbf{w} \in \mathbf{W}} \mathbf{w} \times \chi_{\mathbf{G_w}}$

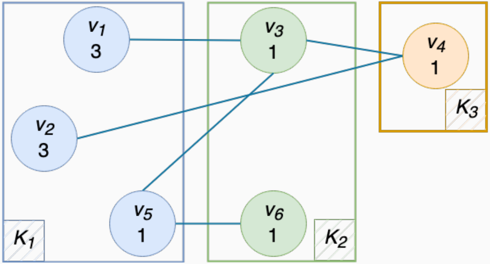
Instance	$\|V\|$	density	$\|W\|/\|V\|$	$\Delta + 1$	lb	ub	lb	ub
DSJC125.1g	125	0.1	0.04	24	4	**14**	19	42
DSJC125.5g	125	0.5	0.04	76	10	**34**	42	105
DSJC125.9g	125	0.9	0.04	121	32	**72**	124	220
DSJR500.1	244	0.03	0.08	26	12	26	166	477
GEOM110	87	0.11	0.11	20	9	20	65	151
inithx.i.1	181	0.05	0.1	169	54	**78**	569	800
le450_15a	420	0.08	0.05	99	15	**61**	206	628
le450_25b	345	0.08	0.06	108	25	**73**	307	735
mulsol.i.5	104	0.23	0.18	88	31	**58**	367	574
queen10_10	100	0.59	0.19	36	10	36	153	420
p42	135	0.12	0.46	25	14	25	2466	8108
r30	301	0.09	0.76	35	19	35	9816	104285

Constraint Programming models for WVCP

Primal model

- Extends classic CP model for GCP to WVCP.
  - Vertices and color dominants as integer variables.
  - Colors as set variables.
- Sorts colors to break symmetries by descending order of weights.

An optimal d-solution of score 5=3+1+1 (virtual vertices omitted).



Primal model for P_k

minimize x^o *subject to*

$$x^o \in \{\max_{v_i \in V}(w(v_i)), \ldots, \sum_{v_i \in V} w(v_i)\} \quad \text{(P1)}$$

$$\forall v_i \in U : x_i^U \in K \quad \text{(P2)}$$

$$\forall k \in K : x_k^K \in 2^U \quad \text{(P3)}$$

$$\forall k \in K : x_k^D \in U \quad \text{(P4)}$$

$$\text{INT\_SET\_CHANNEL}([x_k^K | k \in K], [x_i^U | v_i \in U]) \quad \text{(P5)}$$

$$\forall k \in K : x_{|V|+k}^U = k \quad \text{(P6)}$$

$$\forall \{v_i, v_j\} \in E : x_i^U \neq x_j^U \quad \text{(P7)}$$

$$\forall k \in K : x_k^D = \min(x_k^K) \quad \text{(P8)}$$

$$x^o = \sum_{k \in K} w[x_k^D] \quad \text{(P9)}$$

$$\text{STRICTLY\_INCREASING}(x^D) \quad \text{(P10)}$$

Results on primal model

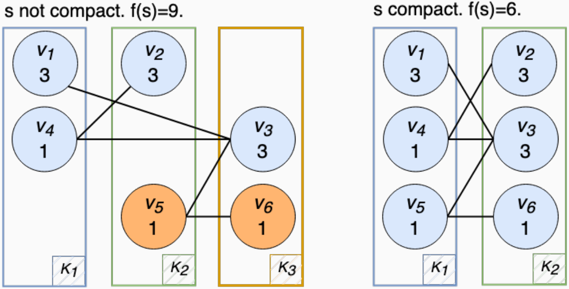
- Using Minizinc with OR-Tools.
- First-fail on vertex variables with domain bisection.
- 1 hour max/run on a single CPU.

	no bounds	color ub	all bounds
nb BKS	101/188	105/188	107/188
nb optim	72/188	75/188	95/188

		primal		primal ub color		primal all bounds	
instance	BKS	score	time(s)	score	time(s)	score	time(s)
DSJC125.1g	23	**23***	862	**23***	435	**23***	451
DSJC125.5gb	240	270	tl	270	tl	270	tl
DSJC125.5g	71	78	tl	78	tl	78	tl
DSJC125.9g	169*	176	tl	176	tl	176	tl
DSJR500.1	169	187	tl	177	tl	169	tl
GEOM110	68*	69	tl	**68***	1893	**68***	1729
inithx.i.1	569*	569	tl	569	tl	**569***	54
le450_15a	212	245	tl	234	tl	234	tl
le450_25b	307	307	tl	307	tl	**307***	322
mulsol.i.5	367*	367	tl	367	tl	**367***	31
queen10_10	162	170	tl	169	tl	169	tl
p42	2466*	2480	tl	2466	tl	**2466***	2908
r30	9816*	9831	tl	9831	tl	9831	tl

Compact d-solutions

The primal model restricts the search to *compact d-solutions*, d-solutions wherein no vertex can be moved to a lower-ranked color.



Theorem. There exists a computable idempotent function that maps any d-solution to a compact d-solution with no score increase.

Global constraint MAX_LEFT_SHIFT for compactness

- Let $y, x_1, \ldots, x_n$ be integer domain variables s.t. $x_i > 0$ ($i = 1..n$). $\text{MAX\_\_LEFT\_\_SHIFT}(y, [x_1, \ldots, x_n]) \leftrightarrow y = \min_{k=1..n+1}(\{k | \bigwedge_{i=1..n} x_i \neq k\})$.
- Ensures solution compactness if applied to each neighborhood.

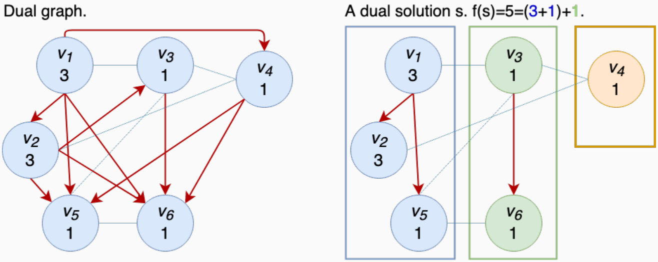
$$\forall v_i \in V : \text{MAX\_\_LEFT\_\_SHIFT}(x_i^U, [x_j^U | v_j \in N(v_i)]) \qquad \text{(P11)}$$

Impact of compactness constraints (MAX_LEFT_SHIFT)

instance	BKS	primal score	primal time(s)	primal + P11 score	primal + P11 time(s)
DSJC125.1g	23	**23***	862	**23***	628
DSJC125.5g	71	78	tl	78	tl
DSJC125.9g	169*	176	tl	176	tl
DSJR500.1	169	187	tl	173	tl
GEOM110	68*	69	tl	**68***	53
inithx.i.1	569*	569	tl	569	tl
le450_15a	212	245	tl	235	tl
le450_25b	307	307	tl	310	tl
mulsol.i.5	367*	367	tl	367	tl
queen10_10	162	170	tl	170	tl
p42	2466*	2480	tl	2480	tl
r30	9816*	9831	tl	9831	tl
nb BKS		101/188		102/188	
nb optim		72/188		76/188	

Dual model for P_k

- Based on reduction of WVCP to Maximum Weighted Stable Set Problem [Cornaz and Jost, 2008].
  - Digraph $(V, \{ij \mid \{v_i, v_j\} \notin E \wedge v_i \geq_w v_j\})$ built from $\overline{E}$ and $\geq_w$.
  - A solution is a forest of simplicial stars spanning disjoint sets of nodes
  - Scored by summing the weights of the target nodes in the stars.



- Adaptation of MIP model of [Cornaz et al., 2017].

Dual model for P_k

maximize y^o *subject to*

$$\forall ij \in \overrightarrow{E^c} : y^A_{ij} \in \{0,1\} \tag{D1}$$

$$y^o \in \{0, \dots, \sum_{v_i \in V} (w(v_i))\} \tag{D2}$$

$$y^o = \sum_{ij \in \overrightarrow{E^c}} w(v_j) \times y^A_{ij} \tag{D3}$$

$$\forall ij, ik \in \overrightarrow{E^c} \ s.t. \ \{jk, kj\} \cap \overrightarrow{E^c} = \emptyset : y^A_{ij} + y^A_{ik} \leq 1 \tag{D4}$$

$$\forall ij, jk \in \overrightarrow{E^c} : y^A_{ij} + y^A_{jk} \leq 1 \tag{D5}$$

$$\forall hj, ij \in \overrightarrow{E^c} : y^A_{hj} + y^A_{ij} \leq 1 \tag{D6}$$

$$\forall v_i \in V : z^V_i \in \{0,1\} \tag{D7}$$

$$\forall v_i \in T : z^V_i = 1 - \max_{(h,i) \in \overrightarrow{E^c}} (y^A_{hi}) \tag{D8}$$

$$\forall v_i \in V \setminus T : z^V_i = 1 \tag{D9}$$

$$\sum_{v_i \in V} z^V_i \leq k \tag{D10}$$

Joint model for P_k

Combines and matches primal and dual solution models.

- (J1) Identical colorings and arc inclusion.
- (J2) Color dominants and star centers.
- (J3) Primal and dual scores.

minimize x^o *s.t.*

$$\forall ij \in \overrightarrow{E^c} : y_{ij}^A \leq (x_i^U = x_j^U) \tag{J1}$$

$$\textsc{gcc}([x_k^D \mid k \in K],\ V,\ [z_i^V \mid v_i \in V]) \tag{J2}$$

$$x^o + y^o = \sum_{v_i \in V} w(v_i) \tag{J3}$$

Comparison of the CP models

		primal		primal + P11		dual		joint	
instance	BKS	score	time(s)	score	time(s)	score	time(s)	score	time(s)
DSJC125.1g	23	**23***	862	**23***	628	26	tl	24	tl
DSJC125.5g	71	78	tl	78	tl	84	tl	78	tl
DSJC125.9g	169*	176	tl	176	tl	**169***	56	**169***	380
DSJR500.1	169	187	tl	173	tl	187	tl	186	tl
GEOM110	68*	69	tl	**68***	53	73	tl	**68***	741
inithx.i.1	569*	569	tl	569	tl	569	tl	**569***	1923
le450_15a	212	245	tl	235	tl	250	tl	-	tl
le450_25b	307	307	tl	310	tl	314	tl	-	tl
mulsol.i.5	367*	367	tl	367	tl	367	tl	**367***	203
queen10_10	162	170	tl	170	tl	177	tl	172	tl
p42	2466*	2480	tl	2480	tl	2517	tl	**2466***	673
r30	9816*	9831	tl	9831	tl	9831	tl	9831	tl
nb BKS		101/188		102/188		79/188		112/188	
nb optim		72/188		76/188		68/188		100/188	

Optimality proofs

Running the CP models with pre-computed bounds during 1h in parallel on 10 threads.

	primal + P11	dual	joint
nb BKS	101/137	102/122	112/132
nb optim	76/130	68/111	100/128

10 new optimality proofs (4 also found with primal without parallelism) :

instance	$\|V\|$	BKS	score	time(s)	instance	$\|V\|$	BKS	score	time(s)
DSJC125.1gb	125	90	**90***	25	myciel7gb	191	109	**109***	69
DSJC125.1g	125	23	**23***	11	myciel7g	191	29	**29***	241
DSJR500.1	500	169	**169***	66	queen9_9g	81	41	**41***	509
myciel6gb	95	94	**94***	17	queen10_10g	100	43	**43***	820
myciel6g	95	26	**26***	17	le450 25b	450	307	**307***	322

Conclusion

- Effective vertex reduction procedure for preprocessing.
- New upper bounds on score/colors from weight-induced subgraphs.
- A new global constraint MAX_LEFT_SHIFT to compact solutions.
- Three competitive and complementary CP models.
- 10 new optimality proofs for difficult benchmark instances.
- Future work : hybridization of CP models with metaheuristics.

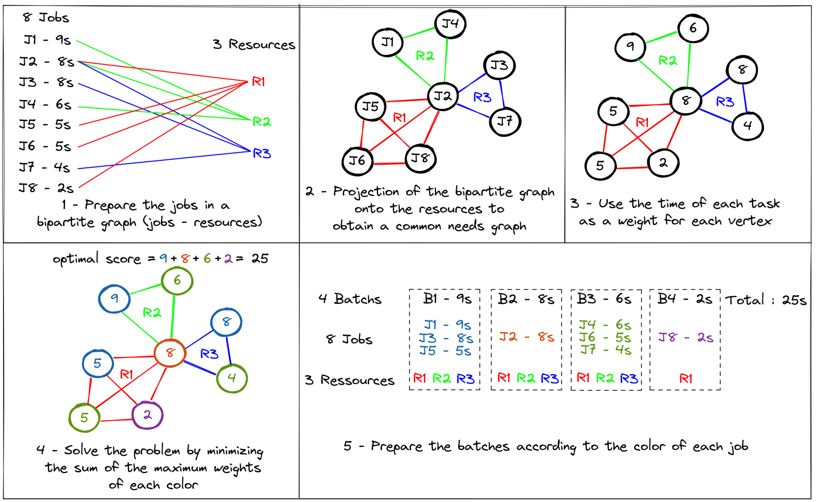
Thank you for your attention !

Technical appendix and source code :

https://github.com/Cyril-Grelier/gc_wvcp_cp

A WVCP application

Scheduling on a Batch Machine with Job Compatibilities



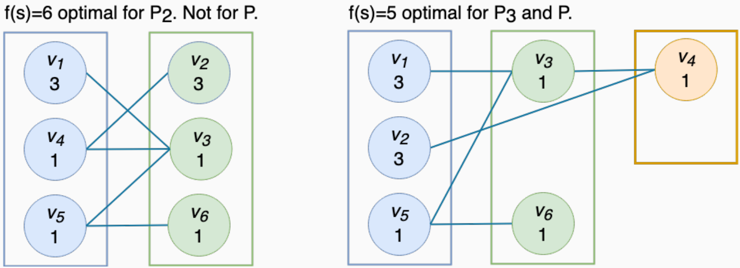
Iterative reduction procedure

1. Extract one clique of maximum weight per vertex using the FastWClq algorithm [Cai and Lin, 2016].
2. Apply R1 (resp. R2) on each vertex considering each clique (resp. each non-adjacent no-lighter vertex).
3. Iterate until no vertex can be removed - updating cliques and graph as needed and maintaining a list L of deleted vertices.

When a solution is found for the reduced graph, a solution of the same score may be obtained for the original instance by coloring each vertex of L with a greedy algorithm in the reverse order of arrival in L.

Upper bounds on the number of colors for WVCP

- Let P_k be the problem of determining a solution to WVCP P using a number of colors $\leq k$.
  - An optimal solution to P_k is not necessarily optimal for P_{k+1}.



- χ_G is not a valid upper bound in the general case.
  - Default upper bound : $k \leq |V|$.
  - [Demange et al., 2007] : $k \leq \Delta(G) + 1$.

Global constraint MAX_LEFT_SHIFT for compactness

Let $y, x_1, \ldots, x_n$ be integer domain variables s.t. $x_i > 0$ $(i = 1..n)$.
$\text{MAX\_LEFT\_SHIFT}(y, [x_1, \ldots, x_n]) \leftrightarrow y = \min_{k=1..n+1}(\{k | \bigwedge_{i=1..n} x_i \neq k\})$.

- Ensures solution compactness if applied to each neighborhood.

$$\forall v_i \in V : \text{MAX\_LEFT\_SHIFT}(x_i^U, [x_j^U | v_j \in N(v_i)]) \tag{P11}$$

- Decomposed using NVALUE - counts the different values taken by $[x_1, \ldots, x_n]$ [Bessiere et al., 2006] :

$$\text{MAX\_LEFT\_SHIFT}(y, [x_1, \ldots, x_n]) \equiv$$

$$\forall i \in \{1, \ldots, n\} : y \neq x_i \tag{M1}$$

$$\forall i \in \{1, \ldots, n\} : z_i \in \{0, \ldots, n+1\} \tag{M2}$$

$$\forall i \in \{1, \ldots, n\} : z_i = (y > x_i) \times x_i \tag{M3}$$

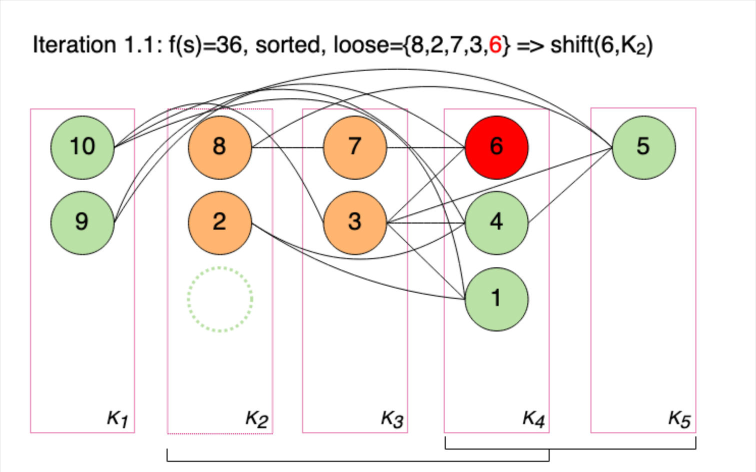
$$\text{NVALUE}(y, [0, z_1, \ldots, z_n]) \tag{M4}$$

3 CP models : primal, dual, joint

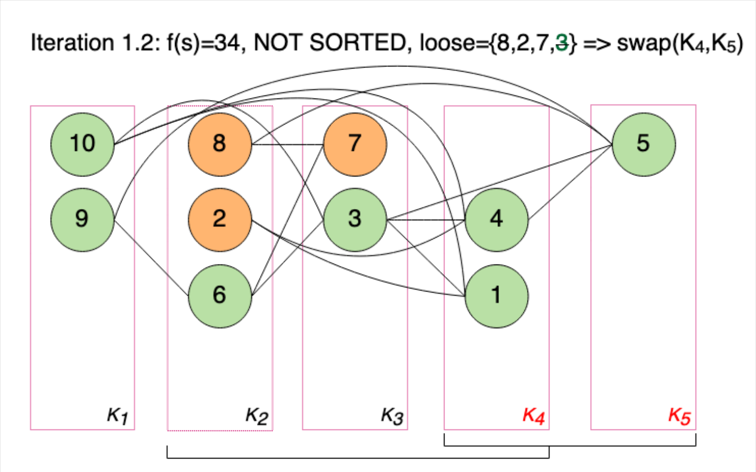
- Address P_k (solving P using a number of colors $\leq k$).
- Must be instantiated with any valid upper bound k.
  - $k = |V|$
  - $k = \Delta(G) + 1$
  - $k = \sum_{w \in W} \chi_{G_w}$
  - ...
- Support additional lower/upper bounds on colors/score.

- Require a total order $\geq_w$ sorting vertices by descending order of weights.
  - *u dominates v* ($u \geq_w v$) implies $w(u) \geq w(v)$ for $u, v \in V$.
  - Each color in a solution has a unique dominant vertex.

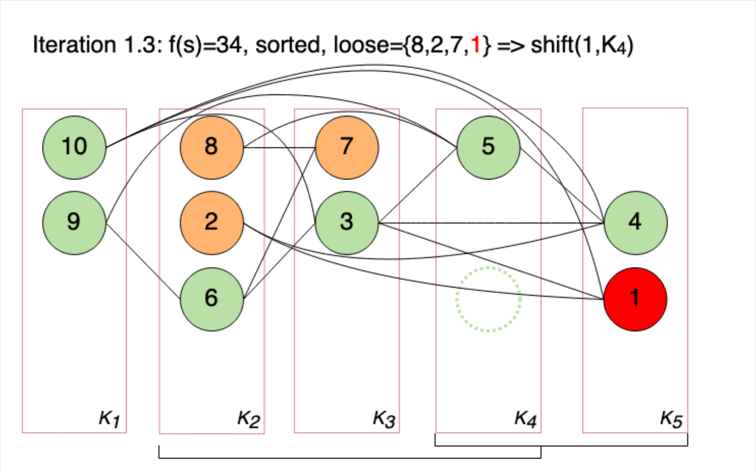
A recursive algorithm to compact d-solutions



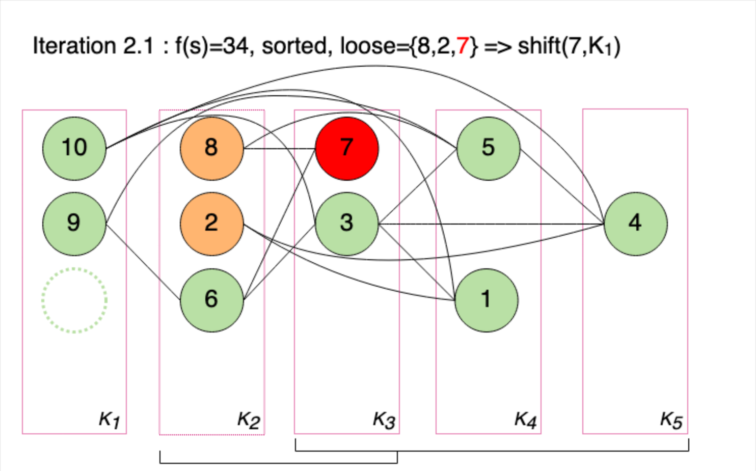
A recursive algorithm to compact d-solutions



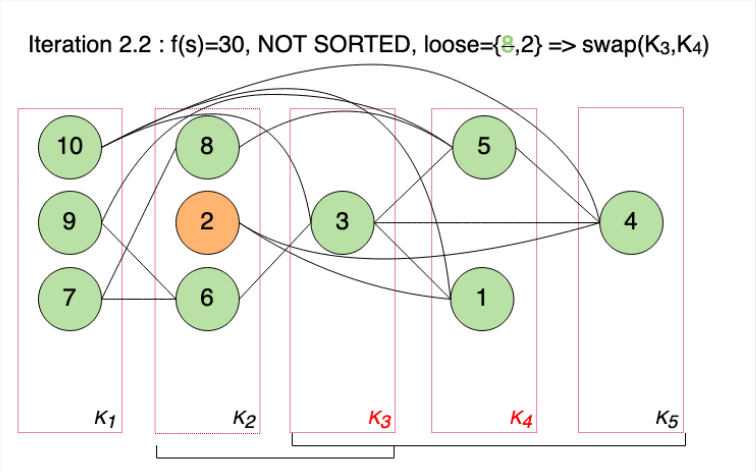
A recursive algorithm to compact d-solutions



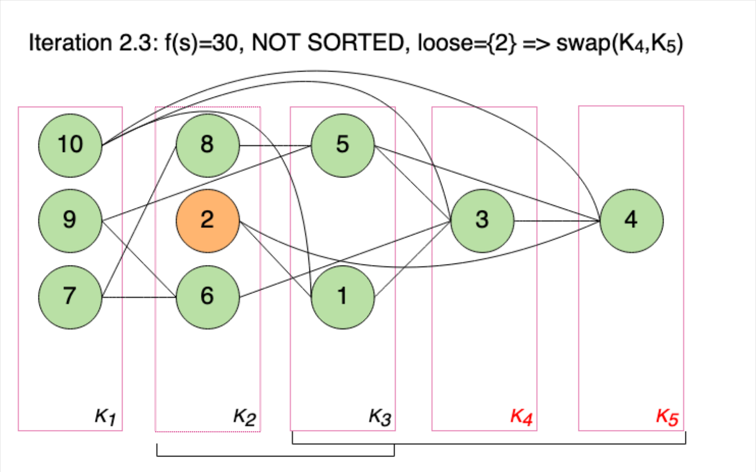
A recursive algorithm to compact d-solutions



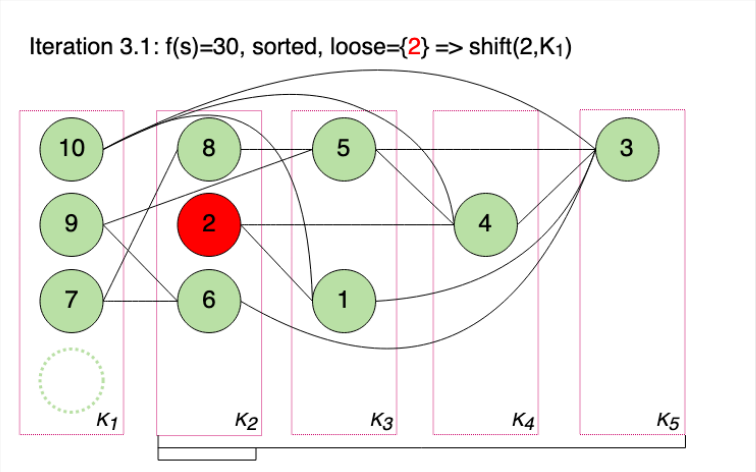
A recursive algorithm to compact d-solutions



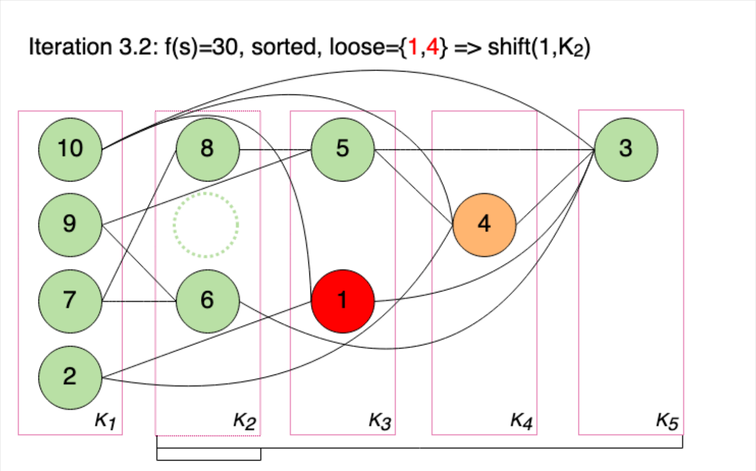
A recursive algorithm to compact d-solutions



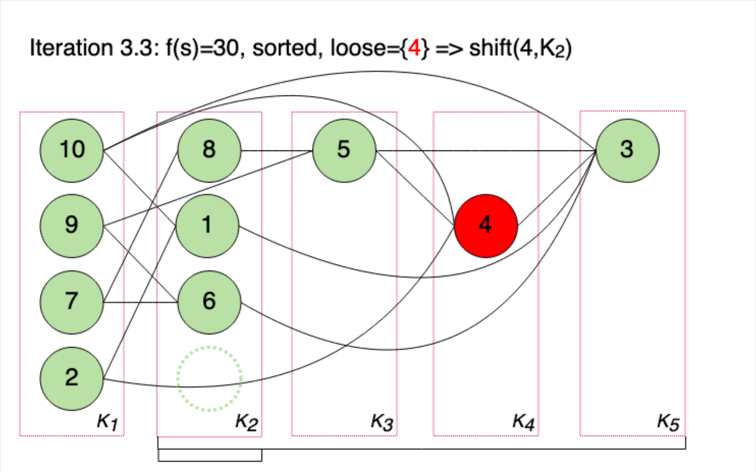
A recursive algorithm to compact d-solutions



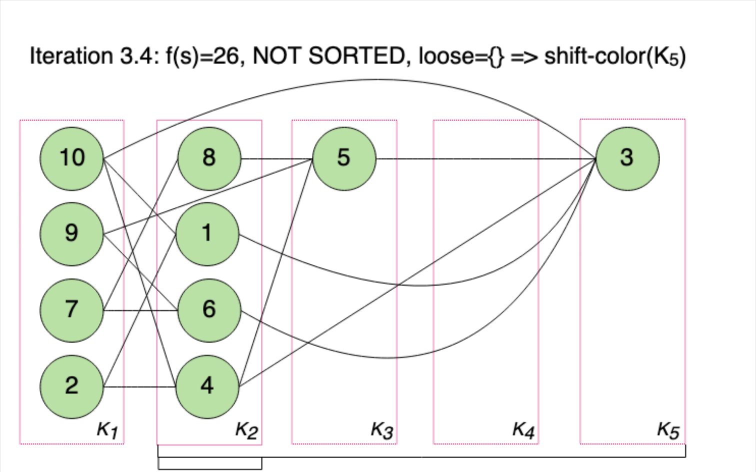
A recursive algorithm to compact d-solutions



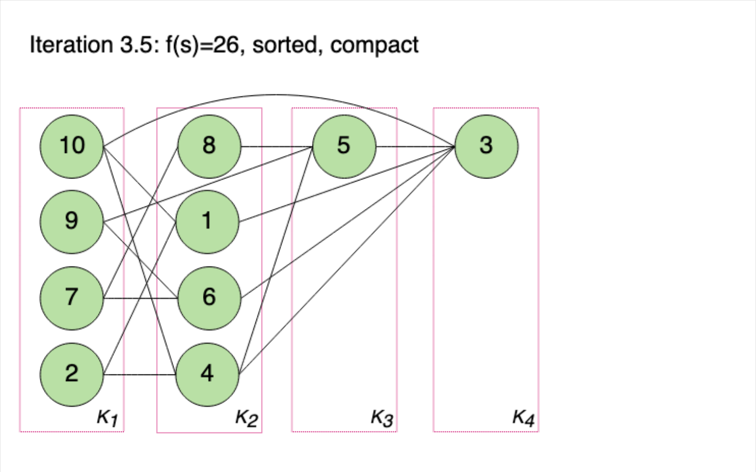
A recursive algorithm to compact d-solutions



A recursive algorithm to compact d-solutions



A recursive algorithm to compact d-solutions



Bibliographie i

Bessiere, C., Hebrard, E., Hnich, B., Kiziltan, Z., and Walsh, T. (2006).
Filtering algorithms for the nvalueconstraint.
Constraints, 11(4) :271–293.

Cai, S. and Lin, J. (2016).
Fast solving maximum weight clique problem in massive graphs.
In *In Proceedings of the 25th International Joint Conference on Artificial intelligence*, pages 568–574.

Cheeseman, P., Kanefsky, B., and Taylor, W. M. (1991).
Where the really hard problems are.
In *Proceedings of the 12th International Joint Conference on Artificial intelligence - Volume 1*, page 331–337.

Bibliographie ii

Cornaz, D., Furini, F., and Malaguti, E. (2017).
Solving vertex coloring problems as maximum weight stable set problems.
Discrete Applied Mathematics, 217 :151–162.

Cornaz, D. and Jost, V. (2008).
A one-to-one correspondence between colorings and stable sets.
Operations Research Letters, 36(6) :673–676.

Demange, M., Werra, D. d., Monnot, J., and Paschos, V. T. (2007).
Time slot scheduling of compatible jobs.
Journal of Scheduling, 10(2) :111–127.

Bibliographie iii

Goudet, O., Grelier, C., and Hao, J.-K. (2022).
A deep learning guided memetic framework for graph coloring problems.
Knowledge-Based Systems, 258 :109986.

Grelier, C., Goudet, O., and Hao, J.-K. (2022).
On monte carlo tree search for weighted vertex coloring.
In Pérez Cáceres, L. and Verel, S., editors, *Evolutionary Computation in Combinatorial Optimization*, volume 13222 of *Lecture Notes in Computer Science*, pages 1–16.

Grelier, C., Goudet, O., and Hao, J.-K. (2023).
Monte carlo tree search with adaptive simulation : A case study on weighted vertex coloring.
In *Evolutionary Computation in Combinatorial Optimization : 23rd European Conference, EvoCOP 2023*, pages 98–113. Springer.

Hertz, A. and de Werra, D. (1987).
Using tabu search techniques for graph coloring.
Computing, 39 :345–351.

Malaguti, E., Monaci, M., and Toth, P. (2009).
Models and heuristic algorithms for a weighted vertex coloring problem.
Journal of Heuristics, 15(5) :503–526.

Nogueira, B., Tavares, E., and Maciel, P. (2021).
Iterated local search with tabu search for the weighted vertex coloring problem.
Computers & Operations Research, 125 :105087.

Bibliographie v

Prais, M. and Ribeiro, C. C. (2000).
Reactive grasp : An application to a matrix decomposition problem in tdma traffic assignment.
INFORMS Journal on Computing, 12(3) :164–176.

Sun, W., Hao, J.-K., Lai, X., and Wu, Q. (2018).
Adaptive feasible and infeasible tabu search for weighted vertex coloring.
Information Sciences, 466 :203–219.

Wang, Yiyuanand Cai, S., Pan, S., Li, X., and Yin, M. (2020).
Reduction and local search for weighted graph coloring problem.
In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 2433–2441.