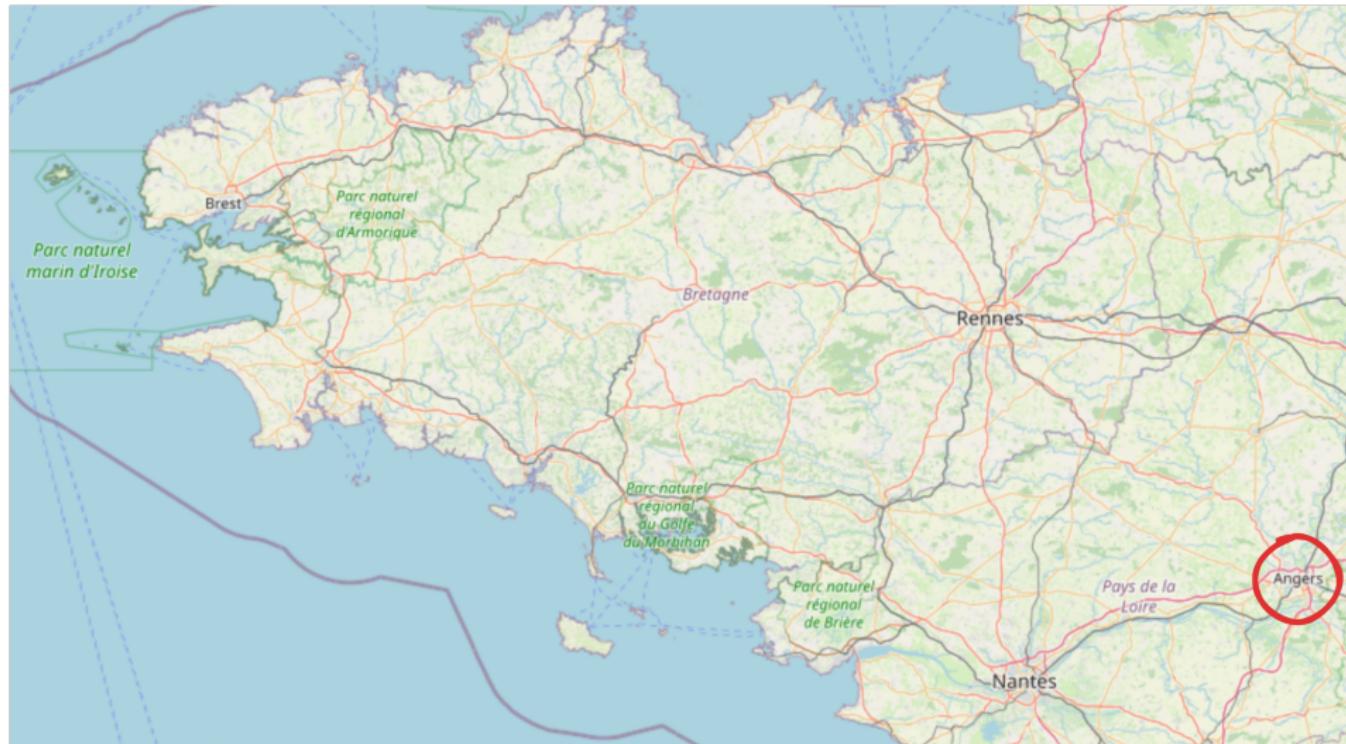


Learning-Guided Metaheuristics for the Graph Coloring Problem

Seminar - IMT Atlantique Brest

Cyril Grelier

Angers



Studies

- 2013 BTEC Higher National Diploma in **Plastics Industry**
BTS - La Baronnerie - Angers
- 2014 Year of scientific upgrading
Université Catholique de l'Ouest - Angers
- 2016 Associate's Degree in **Electrical Engineering** and **Industrial Computing**
DUT - IUT Angers
- 2016 ERASMUS in Sweden
Halmstad University - **English** and **Computer Science** courses
- 2018 Bachelor's Degree in **Computer Science**
Licence - Université d'Angers
- 2020 Master's Degree in **Computer Science - Artificial Intelligence**
Master - Université d'Angers
- 2023 PhD in **Computer Science**
Learning-Guided Metaheuristics for the Graph Coloring Problem
Université d'Angers - LERIA - with Olivier Goudet and Jin-Kao Hao

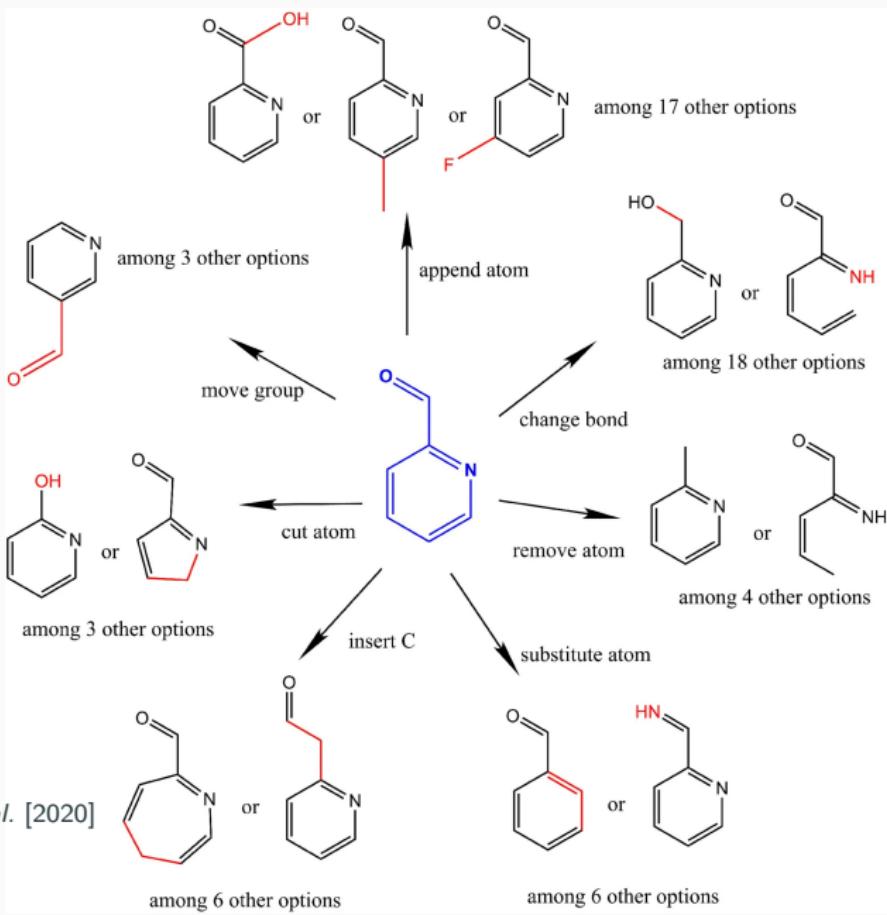
Current Position - Research Engineer at the LERIA

QuChemPedIA: Quantum Chemistry Collaborative EncycloPedIA

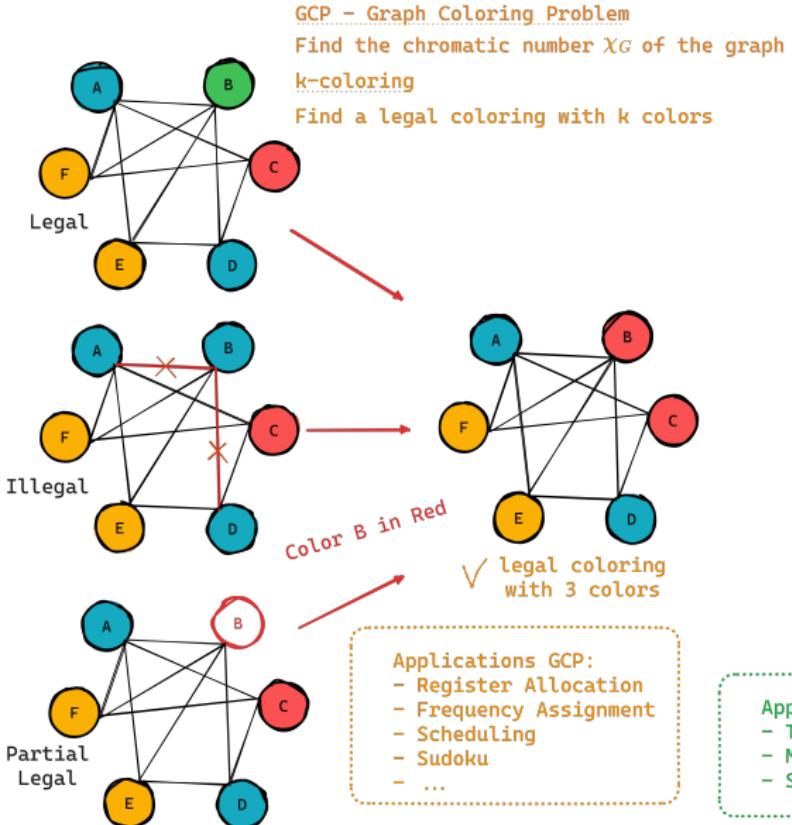
EvoMol (Leguy *et al.* [2020]): an evolutionary algorithm for the generation of molecules with desired properties (light absorption, drug-likeness, synthetic accessibility, etc.)

Objective: more modular code, suitable for commercial use by chemists (still open source), optimized calculations, easier visualization, parallel execution, and integration of reinforcement learning.

EvoMol - Search Space



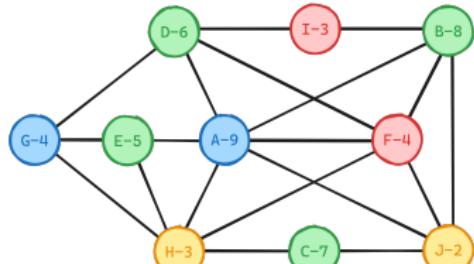
Graph Coloring



WVCP – Weighted Vertex Coloring Problem

Find a legal coloring that minimizes the sum of the weights of the heaviest vertices in each color

$$Score = \sum_{i=1}^k \max_{v \in V_i} w(v)$$



Score =
$$\begin{matrix} 9 & + & 8 & + & 4 & + & 3 & = 24 \\ 4 & & 7 & & 3 & & 2 \end{matrix}$$

Applications WVCP:

- Traffic management in satellite communications
- Matrix decomposition problem
- Scheduling batch job in parallel

Publications

Publications 1/2

Journal Paper

1. A deep learning guided memetic framework for graph coloring problems. OG, CG, & JKH. *Knowledge-Based Systems* 2022 (rank Q1)
Contribution: **DLMCOL** for **GCP** & **WVCP**

Paper in Conference Proceedings

4. A memetic algorithm with adaptive operator selection for graph coloring. CG, OG, & JKH. *EvoCOP 2024* (rank B - acceptance rate $\approx 45\%$)
Contribution: **AHEAD** for **GCP** & **WVCP**
3. New Bounds and Constraint Programming Models for the Weighted Vertex Coloring Problem. OG, CG, & DL. *IJCAI 2023* (rank A* - acceptance rate $< 15\%$)
Contribution: reduction of vertices for **GCP** & **WVCP**, bounds for **WVCP**, 3 **CP** models for **WVCP**
2. Monte Carlo Tree Search with Adaptive Simulation: A Case Study on Weighted Vertex Coloring. CG, OG, & JKH. *EvoCOP 2023*
Contribution: **MCTS + Hyperheuristics** for **WVCP**
1. On Monte Carlo Tree Search for weighted vertex coloring. CG, OG, & JKH. *EvoCOP 2022*
Contribution: **TW(TabuWeights)**, **MCTS + Greedy**, & **MCTS + LS** for **WVCP**

Preprint

1. Combining Monte Carlo Tree Search and Heuristic Search for Weighted Vertex Coloring. CG, OG, & JKH. *Extended version of EvoCOP 2022 for Springer Nature Computer Science*. (rank Q2)
Contribution: original paper + optimized results + discussion about **GCP**

Publications 2/2

Thesis

1. Métaheuristiques Guidées par l'Apprentissage pour la Coloration de Graphe. **CG**. 2023

Presentations in National Conferences

4. Algorithme mémétique avec sélection automatique d'opérateurs pour la coloration de graphe. **CG**, OG, & JKH. *ROADEF 2024*
3. Sélection automatique d'opérateurs dans un arbre de recherche de Monte-Carlo pour la coloration de graphe pondéré. **CG**, OG, & JKH. *ROADEF 2023*
2. Algorithme mémétique guidé par l'apprentissage profond pour des problèmes de coloration de graphes. OG, **CG**, & JKH. *ROADEF 2023*
1. Recherche arborescente Monte-Carlo pour la coloration de graphe pondéré. **CG**, OG, & JKH. *ROADEF 2022*

Authors: Cyril Grelier (**CG**), Olivier Goudet (OG), Jin-Kao Hao (JKH)

Contributions

Vertex Reduction Rules and Iterative Reduction Procedure

-

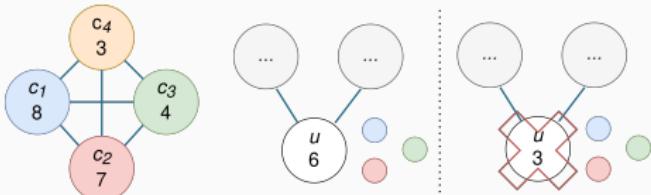
IJCAI 2023

Reduction - Reduction Rules

R0

Wang *et al.* [2020]

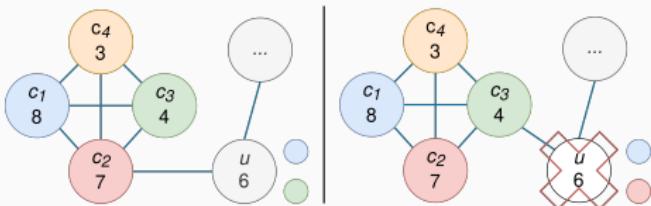
GCP and WVCP



R1

takes into account that u may have neighbors in C

WVCP

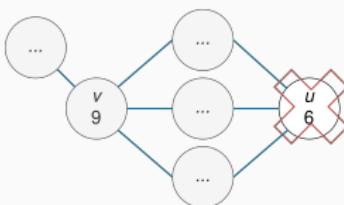


R2

adapted from

Cheeseman *et al.* [1991]

GCP and WVCP



Reduction of vertices - Iterative Procedure and Conclusion

Iterative Reduction Procedure demo

1. Compute a **clique** for each vertex with FastWCIq Cai et Lin [2016]
2. **Sort vertices** by weight then degree
3. **Apply R1 and R2** on each vertex
4. **Repeat** until no vertices are removed
5. Reduced graph is **ready!**

Conclusion

Works often **better** when:

- Graph is **not too dense**
- Existence of **large cliques** (and **heavy** for **WVCP**)
- **Structure** in the graph (geometric/social networks/books/wap/...)

Other points:

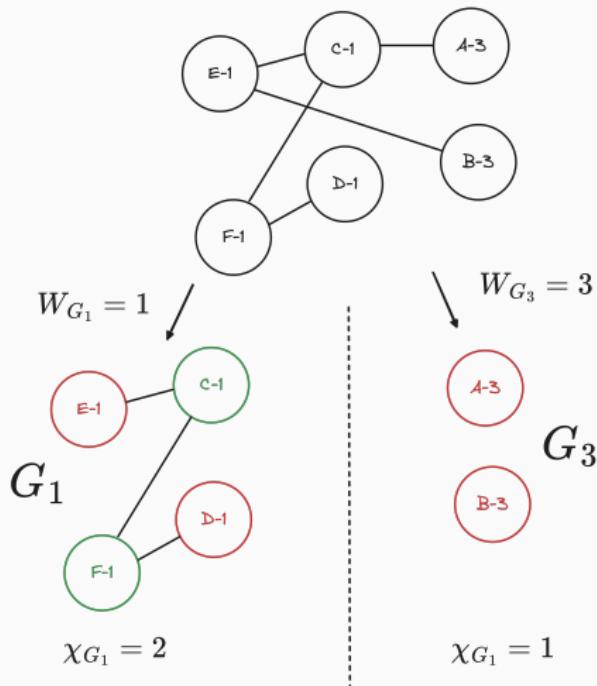
- 67/244 instances out of the **GCP** keep **only one clique**
- **Color original instance from reduced solution** with greedy algorithm
- **Helps exact methods** and **metaheuristics**
- Helps for the calculation of **bounds**

Upper Bound on the Score and Number of Colors for the WVCP

-

IJCAI 2023

Bounds WVCP - Color and Score



Upper Bound on the number of colors :

$$k \leq \sum_{w \in W} \chi_{G_w}$$

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

$$k \leq 2 + 1$$

$$k \leq 3$$

Upper Bound on the score :

$$f(S^*) \leq \sum_{w \in W} w \times \chi_{G_w}$$

$$f(S^*) \leq W_{G_1} \times \chi_{G_1} + W_{G_3} \times \chi_{G_3}$$

$$f(S^*) \leq 1 \times 2 + 3 \times 1$$

$$f(S^*) \leq 5$$

Bounds WVCP - Conclusion

- 88/188 instances with a **better upper bound** on the number of colors compared to $\Delta(G) + 1$ (Demange *et al.* [2007])
 → On average 13 fewer colors
- In practice:
 - Low impact on an MCTS
 - **Better results with CP models**
 → 95 proven optimal instances versus 72 without the bounds

Three Constraint Programming Models for WVCP

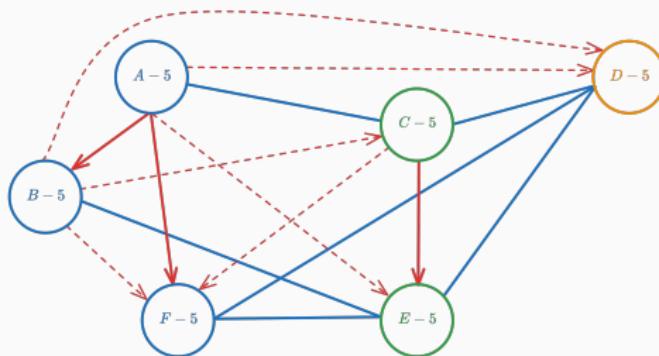
-

IJCAI 2023

CP models - Primal, Dual and Joint

Primal: Compact solution with new global constraint:

MAX_LEFT_SHIFT(reduce the color value for each vertex)



Primal

$$\text{Score}_p = w(A) + w(C) + w(D) = 12$$

Dual

$$\text{Score}_D = w(B) + w(E) + w(F) = 7$$

$$\text{Score}_p + \text{Score}_D = \sum_{i=1}^n w(v_i) = 19$$

Dual: from Cornaz *et al.* [2017] (MIP adapted to CP)

Joint: Primal + Dual + Channeling Constraints

Conclusion:

10 new optimality proofs

Primal-76, Dual-68, Joint-100 optimality proofs/188 instances

Monte Carlo Tree Search

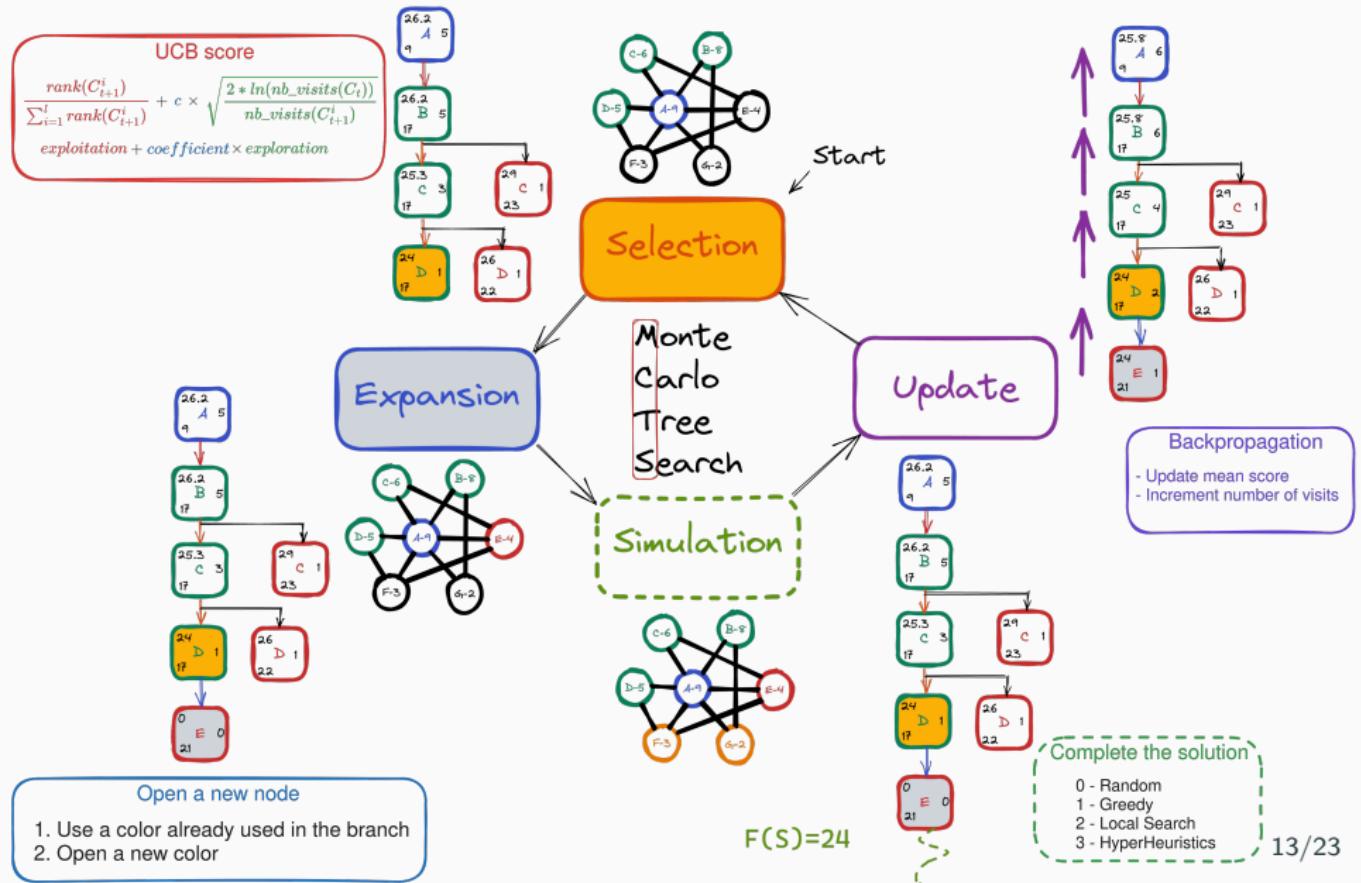
-

EvoCOP 2022

EvoCOP 2023

Preprint SN Computer Science

MCTS - Monte Carlo Tree Search



MCTS + Greedy

Why using a greedy algorithm as simulation?

- Random simulation not efficient enough

Greedy Algorithms

- **R** Random - Random choice in existing colors + one new color
- **C** Constraint - Random choice in existing colors
- **D** Welsh et Powell [1967] Deterministic - First legal color
- **DSatur** Brélaz [1979] Choose the most saturated vertex
- **RLF** Leighton [1979] Construct large stables

To summarize

- **MCTS proves optimality** for 110/244 **GCP** instances and 49/188 **WVCP** instances
- MCTS is very **good on small instances** and some medium ones
- **Less efficient on larger instances**

What if, after the greedy, we launched a local search?

- **Explore the neighborhood** by improving the greedy solution
- **Improve results** on medium/large instances
- Look for a **good starting point** for the local search

To summarize - GCP

- **Local search alone more efficient**
- Different need for **diversification**

To summarize - WVCP

- **Improves results** for 3/4 local searches
- **Results depends** on the **instance** and the **local search**

MCTS + Hyperheuristics - Introduction

Why?

- **No best local search:**
None dominates the others
- **Adaptation:**
Choose the right local search operator **without prior knowledge**

Which local search operators?

- **AFISA** Sun *et al.* [2018]
- **RedLS** Wang *et al.* [2020]
- **ILS-TS** Nogueira *et al.* [2021]
- **TW** Grelier *et al.* [2022]

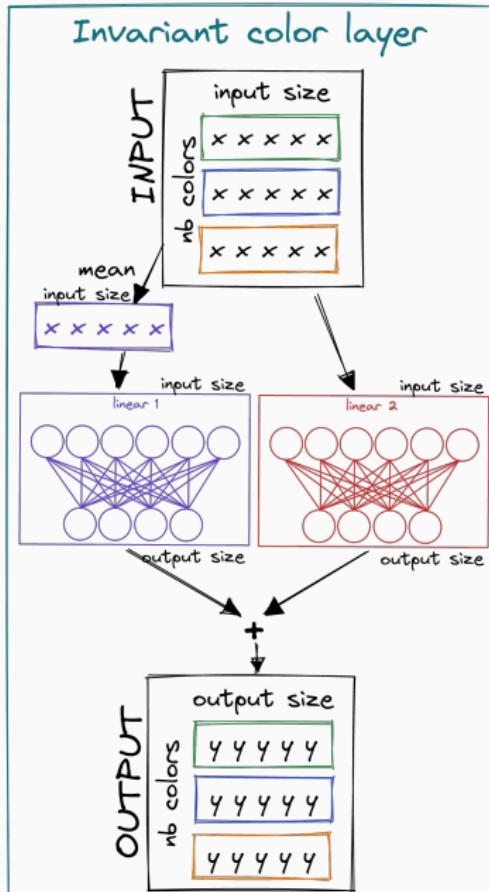
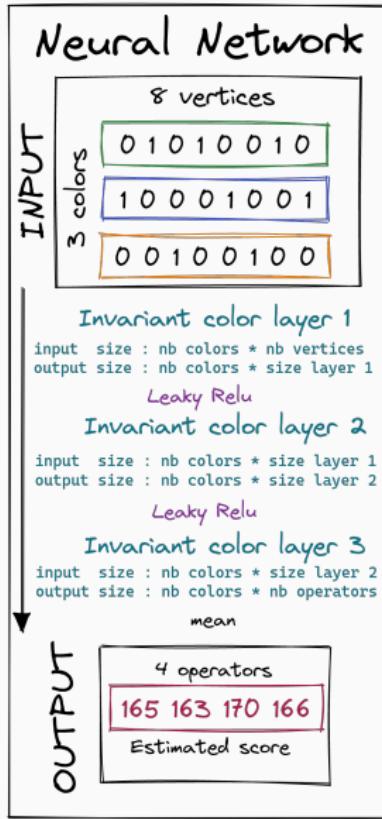
How? During the simulation :

1. Complete the solution
2. **Criteria** selects an operator
3. Perform local search
4. **Update** the criteria (**reward**)

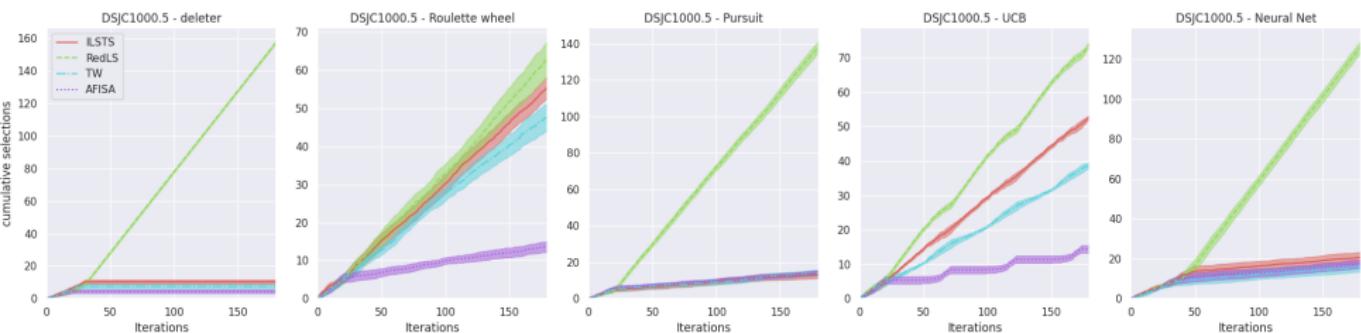
Selection criteria

- **Random** Uniform random choice
- **Deleter** Keeps only best operator
- **Roulette** Random biased by rewards
- **Pursuit** Random in favor of the best
- **UCB** Exploitation and Exploration
- **NN** Neural Network recommendation

Neural Network - NN - Deep sets



MCTS + Hyperheuristics - Conclusion



Selection of Local Search Operators

- Generally **one** or two **good operators per instance**
- Importance of having **complementary operators**
- No change** in the choice during search

To summarize

- MCTS catches up with ILS-TS** on about 15/188 instances
- RedLS and ILS-TS remain better** on about 12 instances where MCTS does not intensify enough

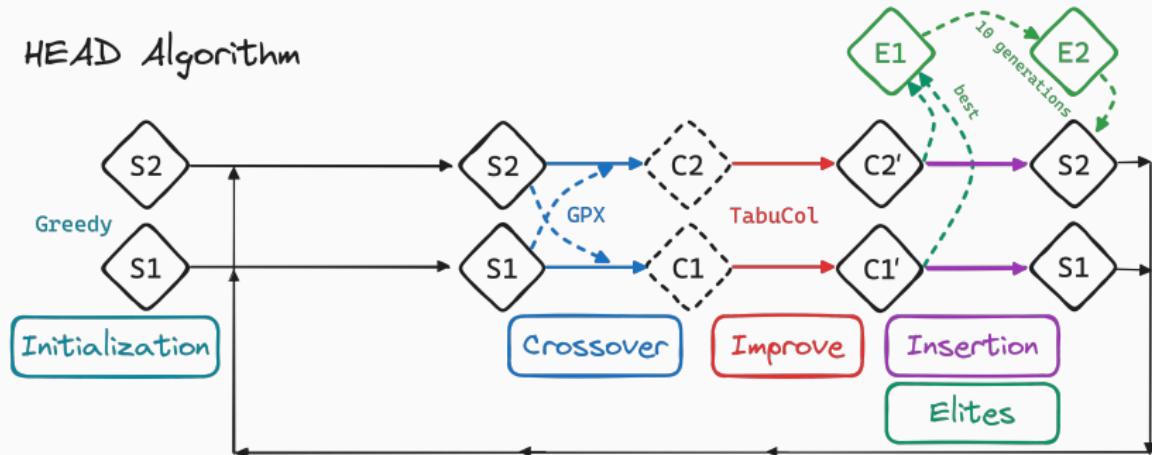
Memetic Algorithm - AHEAD

-

EvoCOP 2024

HEAD - Hybrid Evolutionary Algorithm in Duet

HEAD Algorithm



Moalic et Gondran [2018] – Variations on memetic algorithms for graph coloring problems

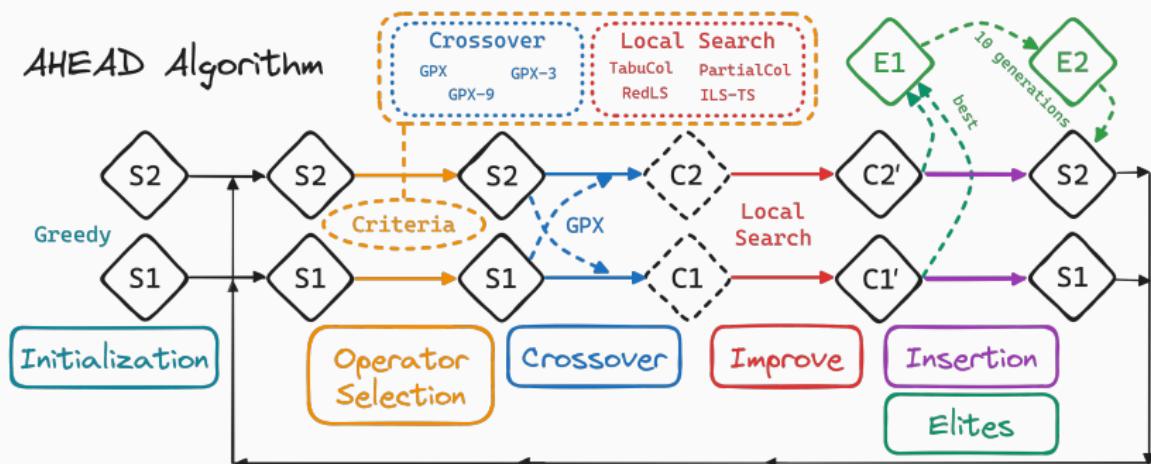
Why HEAD?

- One of the best algorithm for GCP
- Simple and efficient

How?

- 2 individuals
- GPX (Galinier et Hao [1999])
- Improved TabuCol (Hertz et Werra [1987])

AHEAD - Adaptive Hybrid Evolutionary Algorithm in Duet



GCP - Conclusion

- AHEAD improve HEAD
- New best score on C2000.9
(Success: 404 Found)

WVCP - Conclusion

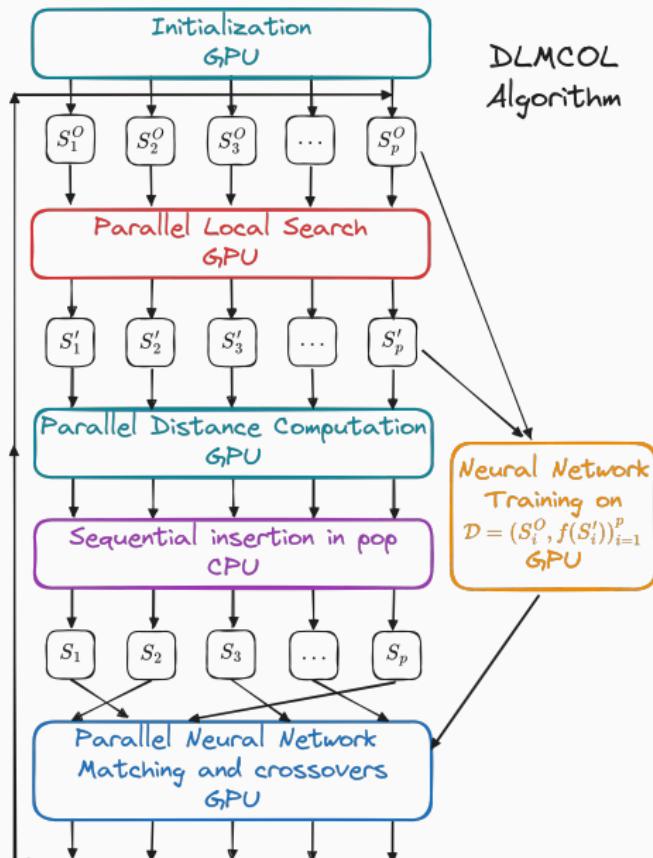
- Better results than LS and MCTS
- New best scores on le450_15a/b
(211/215) and queen14_14 (214)

Memetic Algorithm - DLMCOL

-

Knowledge-Based Systems 2022

DLMCOL - Deep Learning Guided Memetic framework



- DLMCOL Algorithm
- **20 480 individuals**
 - Random initialization
 - Local Search:
 - TabuCol for **GCP**
 - AFISA for **WVCP**
 - **Approximate Distance**
Porumbel *et al.* [2011]
 - **Elitist insertion with distance**
Goudet *et al.* [2021]
 - **NN selects best crossovers**
among 16 or 32

Conclusion

- Requires a **lot of resources**
- **New best scores** for **WVCP**

Conclusion & Perspectives

Conclusion & ★ Perspectives

- **Theoretical contributions**
 - Vertex reduction - IJCAI 2023
 - ★ New rules of reduction
 - Bounds on the number of colors and the score - IJCAI 2023
- **Implemented methods**
 - Constraint programming models - IJCAI 2023
 - MCTS + Greedy - EvoCOP 2022
 - ★ Adapting the exploration vs. exploitation coefficient
 - ★ DSatur when building the tree
 - ★ BeamSearch
 - TabuWeight - EvoCOP 2022
 - MCTS + Local Search - EvoCOP 2022
 - MCTS + Hyperheuristics - EvoCOP 2023
 - ★ Non systematic local search
 - AHEAD - EvoCOP 2024
 - DLMCOL - Knowledge-Based Systems 2022
 - EvoMol
 - ★ TabuEdges
 - ★ EvoWeight
 - ★ LNS

Integration Project - Research

Integration Project - Research

Research interest: **hybridation of metaheuristics, exact methods and machine learning.**

- Collaborate on the project **MAMUT** (Machine learning And Matheuristics algorithms for Urban Transportation)
 - Lots of **features** for VRP problems (Arnold et Sørensen [2019])
 - Features have an impact on resolution (Herdianto *et al.* [2023])
 - Could use **Hyperheuristics** (Dokeroglu *et al.* [2023]) to
 - adapt to the features of the instance before search
 - adapt to the features of the solution during search
 - adapt to the raw solution during search
 - **Adaptive Large Neighborhood Search** (Mara *et al.* [2022])
- Propose an **ANR project**
 - **Automatic selection of operators with reinforcement learning** (mutation, crossover, perturbation, exact model, local search, ...)
 - In a metaheuristics framework (**intensification/diversification**)
 - Ability to learn to **destruct** the solution **to reach** a better one

Thank you for your attention!

Questions?

Publications, source code, results:



<https://cyril-grelier.github.io/>

References

- Florian Arnold and Kenneth Sørensen. What makes a VRP solution good? The generation of problem-specific knowledge for heuristics. *Computers & Operations Research*, 106:280–288, 2019.
- Christian Bessiere, Emmanuel Hebrard, Brahim Hnich, Zeynep Kiziltan, and Toby Walsh. Filtering algorithms for the nvalueconstraint. *Constraints*, 11(4):271–293, 2006.
- Ivo Blöchliger and Nicolas Zufferey. A graph coloring heuristic using partial solutions and a reactive tabu scheme. *Computers & Operations Research*, 35(3):960–975, 2008.
- Daniel Brélaz. New methods to color the vertices of a graph. *Communications of the ACM*, 22(4):251–256, 1979.
- Rowland Leonard Brooks. On colouring the nodes of a network. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 37, pages 194–197. Cambridge University Press, 1941.
- Shaowei Cai and Jinkun Lin. Fast solving maximum weight clique problem in massive graphs. In *IJCAI*, pages 568–574, 2016.

Bibliography ii

- Tristan Cazenave, Benjamin Negrevergne, and Florian Sikora. Monte carlo graph coloring. In *Monte Carlo Search: First Workshop, MCS 2020, Held in Conjunction with IJCAI 2020, Virtual Event, January 7, 2021, Proceedings 1*, pages 100–115. Springer, 2021.
- Peter C Cheeseman, Bob Kanefsky, William M Taylor, et al. Where the really hard problems are. In *IJCAI*, volume 91, pages 331–337, 1991.
- Denis Cornaz, Fabio Furini, and Enrico Malaguti. Solving vertex coloring problems as maximum weight stable set problems. *Discrete Applied Mathematics*, 217:151–162, 2017.
- Denis Cornaz and Vincent Jost. A one-to-one correspondence between colorings and stable sets. *Operations Research Letters*, 36(6):673–676, 2008.
- Marc Demange, D de Werra, Jérôme Monnot, and V Th Paschos. Time slot scheduling of compatible jobs. *Journal of Scheduling*, 10(2):111–127, 2007.
- Tansel Dokeroglu, Tayfun Kucukyilmaz, and El-Ghazali Talbi. Hyper-heuristics: A survey and taxonomy. *Computers & Industrial Engineering*, page 109815, 2023.
- Philippe Galinier and Jin-Kao Hao. Hybrid evolutionary algorithms for graph coloring. *Journal of Combinatorial Optimization*, 3:379–397, 1999.
- Adrien Goëffon, Frédéric Lardeux, and Frédéric Saubion. Simulating non-stationary operators in search algorithms. *Applied Soft Computing*, 38:257–268, 2016.
- Olivier Goudet, Béatrice Duval, and Jin-Kao Hao. Population-based gradient descent weight learning for graph coloring problems. *Knowledge-Based Systems*, 212:106581, 2021.

Bibliography iii

- Olivier Goudet, Cyril Grelier, and Jin-Kao Hao. A deep learning guided memetic framework for graph coloring problems. *Knowledge-Based Systems*, 258:109986, 2022.
- Olivier Goudet, Cyril Grelier, and David Lesaint. New bounds and constraint programming models for the weighted vertex coloring problem. *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*, pages 1927–1934, 2023.
- Cyril Grelier, Olivier Goudet, and Jin-Kao Hao. On monte carlo tree search for weighted vertex coloring. In *Evolutionary Computation in Combinatorial Optimization: 22nd European Conference, EvoCOP 2022, Held as Part of EvoStar 2022, Madrid, Spain, April 20–22, 2022, Proceedings*, pages 1–16. Springer, 2022.
- Cyril Grelier, Olivier Goudet, and Jin-Kao Hao. 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, Held as Part of EvoStar 2023, Brno, Czech Republic, April 12–14, 2023, Proceedings*, pages 98–113. Springer, 2023.
- Cyril Grelier, Olivier Goudet, and Jin-Kao Hao. A memetic algorithm with adaptive operator selection for graph coloring. In *European Conference on Evolutionary Computation in Combinatorial Optimization (Part of EvoStar)*, pages 65–80. Springer, 2024.
- Bachtiar Herdianto, Romain Billot, Flavien Lucas, and Marc Sevaux. Integration of Knowledge Extraction Processes into Metaheuristic Algorithm. In *21th EU/ME meeting x Quantum School "Emerging optimization methods: from metaheuristics to quantum approaches"*, Troyes, France, April 2023.

Bibliography iv

- Alain Hertz and D de Werra. Using tabu search techniques for graph coloring. *Computing*, 39(4):345–351, 1987.
- Jorik Jooken, Pieter Leyman, Tony Wauters, and Patrick De Causmaecker. Exploring search space trees using an adapted version of monte carlo tree search for combinatorial optimization problems. *Computers & Operations Research*, 150:106070, 2023.
- Jules Leguy, Thomas Cauchy, Marta Glavatskikh, Béatrice Duval, and Benoit Da Mota. Evomol: a flexible and interpretable evolutionary algorithm for unbiased de novo molecular generation. *Journal of cheminformatics*, 12:1–19, 2020.
- Frank Thomson Leighton. A graph coloring algorithm for large scheduling problems. *Journal of research of the national bureau of standards*, 84(6):489, 1979.
- Zhipeng Lü and Jin-Kao Hao. A memetic algorithm for graph coloring. *European Journal of Operational Research*, 203(1):241–250, 2010.
- Enrico Malaguti, Michele Monaci, and Paolo Toth. Models and heuristic algorithms for a weighted vertex coloring problem. *Journal of Heuristics*, 15:503–526, 2009.
- Setyo Tri Windras Mara, Rachmadi Norcahyo, Panca Jodiawan, Luluk Lusiantoro, and Achmad Pratama Rifai. A survey of adaptive large neighborhood search algorithms and applications. *Computers & Operations Research*, 146:105903, 2022.
- Laurent Moalic and Alexandre Gondran. Variations on memetic algorithms for graph coloring problems. *Journal of Heuristics*, 24:1–24, 2018.

Bibliography v

- Bruno Nogueira, Eduardo Tavares, and Paulo Maciel. Iterated local search with tabu search for the weighted vertex coloring problem. *Computers & Operations Research*, 125:105087, 2021.
- Laurent Perron and Vincent Furnon. Or-tools v9.4.
<https://developers.google.com/optimization/>, 2022. Accessed: 2023-05-25.
- Daniel Cosmin Porumbel, Jin-Kao Hao, and Pascale Kuntz. A search space “cartography” for guiding graph coloring heuristics. *Computers & Operations Research*, 37(4):769–778, 2010.
- Daniel Cosmin Porumbel, Jin Kao Hao, and Pascale Kuntz. An efficient algorithm for computing the distance between close partitions. *Discrete Applied Mathematics*, 159(1):53–59, 2011.
- Wen Sun, Jin-Kao Hao, Xiangjing Lai, and Qinghua Wu. Adaptive feasible and infeasible tabu search for weighted vertex coloring. *Information Sciences*, 466:203–219, 2018.
- Yiyuan Wang, Shaowei Cai, Shiwei Pan, Ximing Li, and Monghao Yin. Reduction and local search for weighted graph coloring problem. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2433–2441, 2020.
- Dominic JA Welsh and Martin B Powell. An upper bound for the chromatic number of a graph and its application to timetabling problems. *The Computer Journal*, 10(1):85–86, 1967.
- Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.
- Yangming Zhou, Béatrice Duval, and Jin-Kao Hao. Improving probability learning based local search for graph coloring. *Applied Soft Computing*, 65:542–553, 2018.

EvoMol

Graph Coloring

State of the Art

Vertex Reduction Rules and Iterative Reduction Procedure

Upper Bound on the Score and Number of Colors for the WVCP

Three Constraint Programming Models for WVCP

MCTS

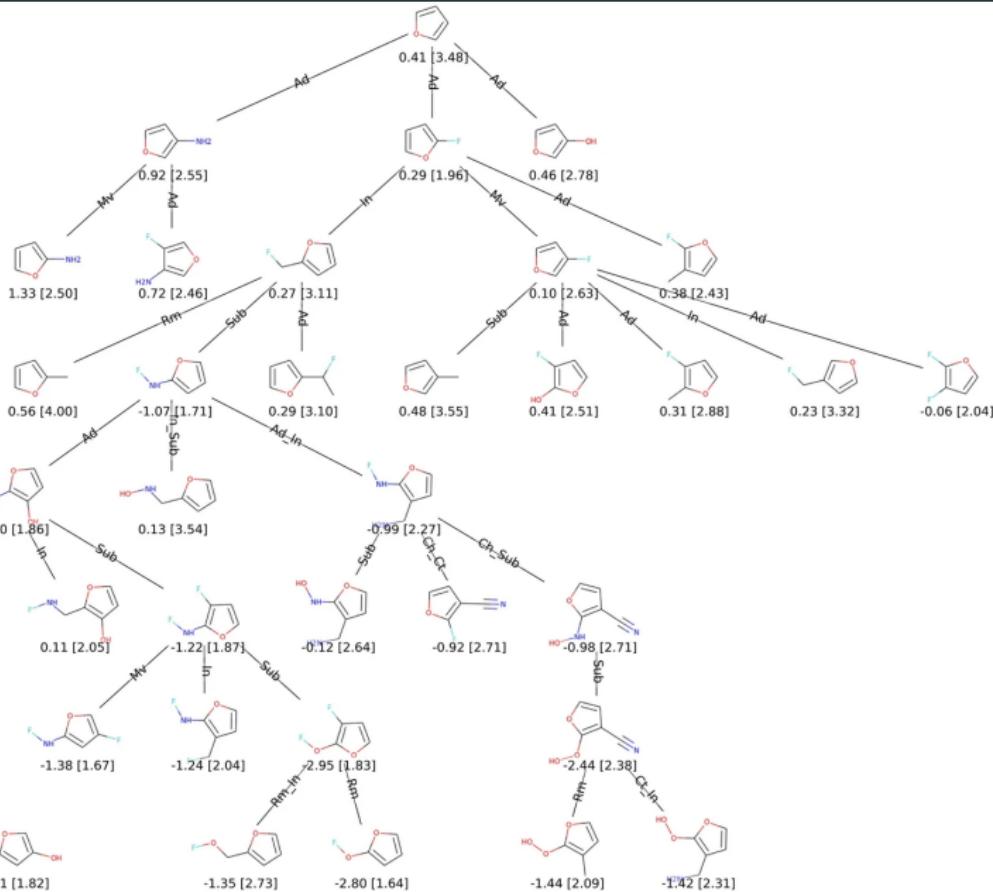
AHEAD

DLMCOL

EvoMol

EvoMol - Tree Exploration (min LUMO energies)

Source: Leguy *et al.* [2020]



EvoMol - Tree Exploration (QED optimisation)



Source: Leguy *et al.* [2020]

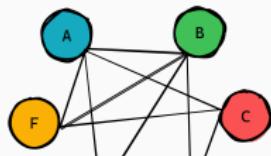
Graph Coloring

GCP - Graph Coloring Problem

Example of k -coloring with $k=3$

Search Space

Legal

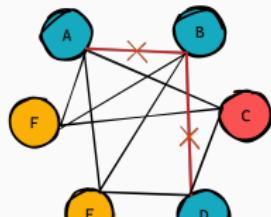


Unsatisfied constraints

Score

GCP : Find the chromatic number of the graph

Illegal



4 colors used

Score = 4

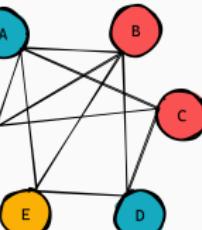
2 conflicts

Score = 2

k -coloring : Find a legal coloring with k colors

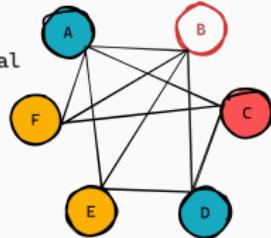
NP-Hard Problems

Color B in Red



legal coloring with 3 colors

Partial-Legal



vertex B not colored

Score = 1

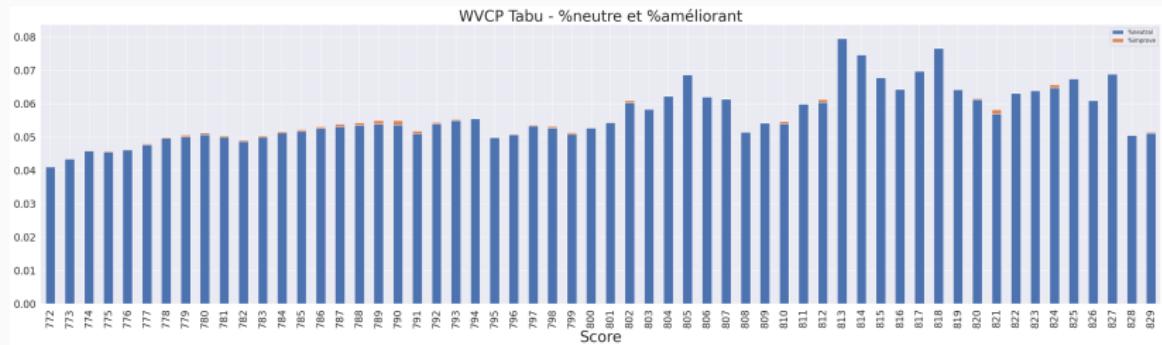
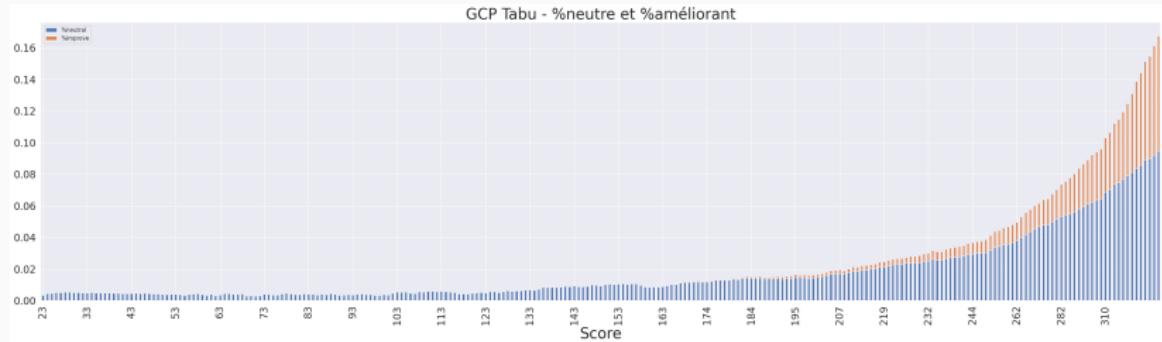
Applications:

- Register Allocation
- Frequency Assignment
- Scheduling
- Sudoku
- ...

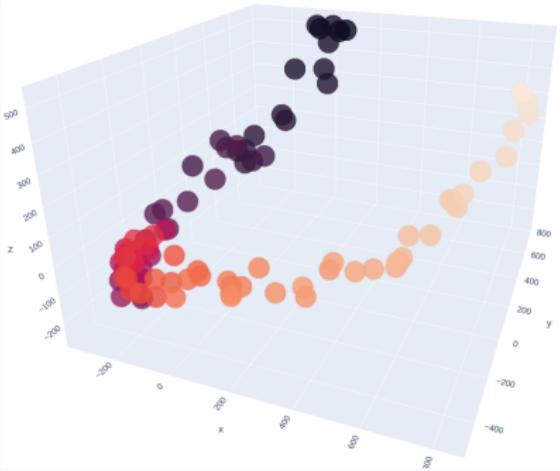
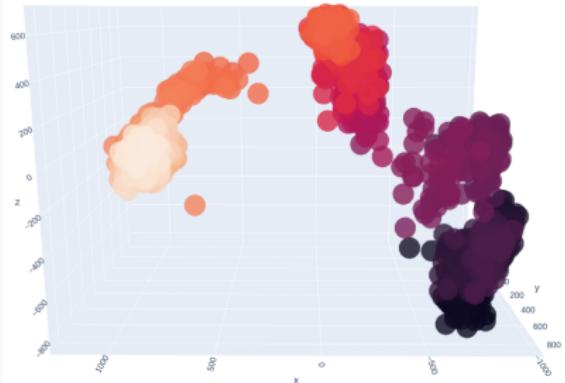
WVCP - Scheduling Parallel Batch Jobs

<p>8 Jobs</p> <p>J1 - 9s J2 - 8s J3 - 8s J4 - 6s J5 - 5s J6 - 5s J7 - 4s J8 - 2s</p> <p>3 Resources</p> <p>1 - Prepare the jobs in a bipartite graph (jobs - resources)</p>	<p>2 - Projection of the bipartite graph onto the resources to obtain a common needs graph</p>	<p>3 - Use the time of each task as a weight for each vertex</p>															
<p>optimal score = $9 + 8 + 6 + 2 = 25$</p> <p>4 - Solve the problem by minimizing the sum of the maximum weights of each color</p>	<p>4 Batches</p> <table border="1"> <tr> <td>B1 - 9s</td> <td>B2 - 8s</td> <td>B3 - 6s</td> <td>B4 - 2s</td> <td>Total : 25s</td> </tr> <tr> <td>J1 - 9s J3 - 8s J5 - 5s</td> <td>J2 - 8s</td> <td>J4 - 6s J6 - 5s J7 - 4s</td> <td>J8 - 2s</td> <td></td> </tr> <tr> <td>R1 R2 R3</td> <td>R1 R2 R3</td> <td>R1 R2 R3</td> <td>R1</td> <td></td> </tr> </table> <p>8 Jobs</p> <p>3 Resources</p>	B1 - 9s	B2 - 8s	B3 - 6s	B4 - 2s	Total : 25s	J1 - 9s J3 - 8s J5 - 5s	J2 - 8s	J4 - 6s J6 - 5s J7 - 4s	J8 - 2s		R1 R2 R3	R1 R2 R3	R1 R2 R3	R1		<p>5 - Prepare the batches according to the color of each job</p>
B1 - 9s	B2 - 8s	B3 - 6s	B4 - 2s	Total : 25s													
J1 - 9s J3 - 8s J5 - 5s	J2 - 8s	J4 - 6s J6 - 5s J7 - 4s	J8 - 2s														
R1 R2 R3	R1 R2 R3	R1 R2 R3	R1														

WVCP - GCP - Exploration



WVCP - GCP - Exploration



https://cyril-grelier.github.io/assets/html/search_path_wvcp.html

https://cyril-grelier.github.io/assets/html/search_path_gcp.html

State of the Art

GCP - State of the Art and Contributions

- Local Search:
 - **TabuCol** Hertz et Werra [1987] : illegal, one-move
 - **PartialCol** Blöchliger et Zufferey [2008] : partial legal, grenade
 - **TabuEdges** [work in progress] guided local search
- Memetic Algorithm :
 - **HEA** Galinier et Hao [1999] : GPX, TabuCol
 - **Evo-Div** Porumbel *et al.* [2010] : multi-parents crossover, distances
 - **MACOL** Lü et Hao [2010] : multi-parents crossover, distances
 - **HEAD** Moalic et Gondran [2018] : HEA in Duet, 2 individuals
 - **DLMCOL** Goudet *et al.* [2022] : MA, +20 000, NN select crossover
 - **AHEAD** Grelier *et al.* [2024] Adaptive HEAD
 - **EvoWeight** [work in progress] guided memetic algorithm
- Learning :
 - **PLSCOL** Zhou *et al.* [2018] : local search, reinforcement learning
 - **TensCol** Goudet *et al.* [2021] : tensor, gradient descent
 - **NRPA** Cazenave *et al.* [2021] : MCTS, sequence, gradient descent
 - **MCTS + Greedy** [Thesis - Chap 5] MCTS, greedy as simulation

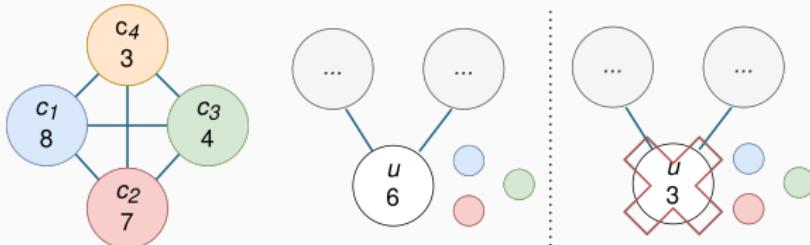
WVCP - State of the Art and Contributions

- Learning :
 - **MCTS + Local Search** Grelier *et al.* [2022] LS as simulation
 - **MCTS + Hyperheuristics** Grelier *et al.* [2023] : select LS
- Memetic Algorithms :
 - **DLMCOL** Goudet *et al.* [2022] : MA, +20000, NN select crossover
 - **AHEAD** Grelier *et al.* [2024] : Adaptive HEAD
- Local Search :
 - **AFISA** Sun *et al.* [2018] : illegal, one-move, adaptive coefficient
 - **RedLS** Wang *et al.* [2020] : illegal, weighted edges, perturbations
 - **ILS-TS** Nogueira *et al.* [2021] : p-legal, 6 neighbors, perturbations
 - **TW (TabuWeight)** Grelier *et al.* [2022] : legal, one-move
- Exact Methods :
 - **2-Phase** Malaguti *et al.* [2009] : column generation, ILP
 - **MWSS** Cornaz *et al.* [2017] : MIP, max weight stable set problem
 - **CP** Goudet *et al.* [2023] : 3 CP models, reduction, bounds

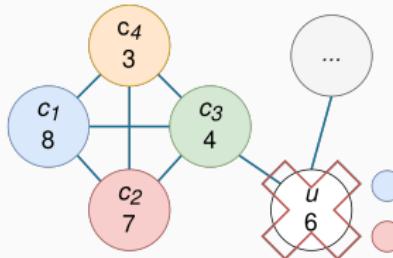
Vertex Reduction Rules and Iterative Reduction Procedure

Reduction Rules R0 and R1

- R0, Wang *et al.* [2020] (GCP and WVCP):

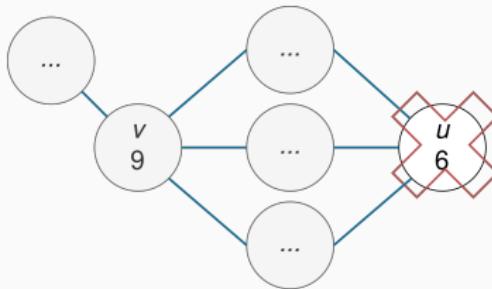


- R1, takes into account that u may have neighbors in C (WVCP):



Reduction Rule R2 and Iterative Reduction Procedure

- R2 adapted from Cheeseman *et al.* [1991] (GCP and WVCP):



- Iterative Reduction Procedure : demo¹
 1. Compute a clique for each vertex with FastWCIq Cai et Lin [2016]
 2. Sort vertices by weight then degree
 3. Apply R1 and R2 on each vertex
 4. Repeat until no vertices are removed
 5. Reduced graph is ready!

¹<https://tinyurl.com/gc-reduction>

Reduction - Results

$\# RI$: number of reduced instances
 $\# RV$: number of removed vertices

$\%RV$: percentage of removed vertices
 $t(s)$: average time in seconds

GCP - /244	$\# RI$	$\# RV$ avg	$\# RV$ max	$\%RV$ avg	$\%RV$ max	t(s) avg
R0	132	80.2	1199	26.3	87.7	16.3
R1	132	80.2	1199	26.3	87.7	16.3
R1+R2	146	115.4	1960	29.6	87.7	78.1
Iterative	146	182.5	4033	58.5	97.7	96.7

WVCP - /188	$\# RI$	$\# RV$ avg	$\# RV$ max	$\%RV$ avg	$\%RV$ max	t(s) avg
R0	82	34.2	469	13.4	65	2.6
R1	84	39.5	574	14.7	66.4	3.8
R1+R2	85	41.7	596	15.4	69	4.1
Iterative	85	54.3	683	23.3	80.9	9.8

Reduction - Results WVCP

instance	V	density	R0	R1	R1+R2	Iterated	time(s)
DSJC125.1g	125	0.1	0	0	0	0	0
DSJC125.5g	125	0.5	0	0	0	0	0
DSJC125.9g	125	0.9	0	0	0	0	3
DSJR500.1	500	0.0	78	80	80	256	1
GEOM110	110	0.1	6	9	9	23	0
inithx.i.1	864	0.1	469	574	596	683	19
le450_15a	450	0.1	28	28	28	30	1
le450_25b	450	0.1	90	90	90	105	2
mulsol.i.5	186	0.2	28	53	75	82	1
queen10_10	100	0.6	0	0	0	0	0
p42	138	0.1	1	1	1	3	0
r30	301	0.1	0	0	0	0	0
wap02a	2464	0.0	161	165	165	249	168
wap04a	5231	0.0	244	244	244	321	527

Reduction - Results GCP

instance	V	density	R0	R1	R1+R2	Iterated	time(s)
DSJC125.1	125	0.1	0	0	0	0	0
DSJC125.5	125	0.5	0	0	0	0	0
DSJC125.9	125	0.9	0	0	0	0	1
DSJR500.1	500	0.1	150	150	151	488	0
GEOM110	110	0.1	17	17	17	101	0
inithx.i.1	864	0.1	705	705	709	769	4
le450_15a	450	0.1	41	41	41	43	0
le450_25b	450	0.1	131	131	131	156	0
mulsol.i.5	186	0.2	106	106	108	114	0
queen10_10	100	0.3	0	0	0	0	0
p42	138	0.1	10	10	10	124	0
r30	301	0.1	0	0	0	0	0
r1000.1	1000	0.1	99	99	99	954	0
wap04a	5231	0.0	1199	1199	1199	1199	26

Reduction - Conclusion

Works often better when:

- Graph is not too dense
- Existence of large cliques (and heavy for WVCP)
- Structure in the graph (geometric graphs, social networks/books/wap graphs, . . .)

Other points:

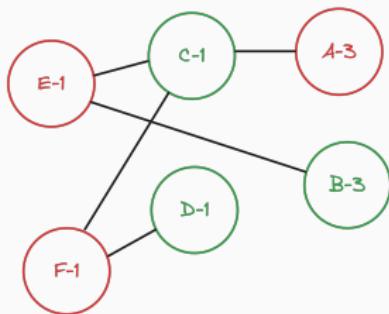
- 67 instances out of 244 of the GCP keep only one clique.
- Help for exact and approximate resolution methods
- Help for the calculation of bounds

Upper Bound on the Score and Number of Colors for the WVCP

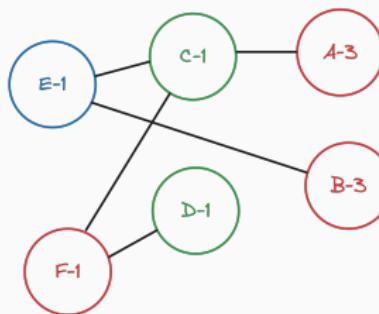
Bounds WVCP - Introduction

Why ?

- No limit on the number of colors
- Necessary for exact methods
- Reduce the search space



$$\begin{aligned} \text{Score} &= 3 + 3 = 6 \\ &\begin{matrix} 1 & 1 \\ 1 & 1 \end{matrix} \end{aligned}$$



$$\begin{aligned} \text{Score} &= 3 + 1 + 1 = 5 \\ &\begin{matrix} 3 & 1 \\ 1 & \end{matrix} \end{aligned}$$

Bounds WVCP - State of the Art

Number of Colors

- Lower bound : size of the largest clique
- Upper bound : $k \leq \Delta(G) + 1$
Brooks [1941] GCP, Demange *et al.* [2007] WVCP

Score

- Lower bound : weight of the heaviest clique
- Upper bound : sum of the weights of the vertices, BKS ²

²Best Known Score in the literature

Bounds WVCP - Contributions

New bounds - see Theorem in Goudet *et al.* [2023]

Given an instance of the WVCP $G = (V, E, w)$ and an optimal solution S^* with k color groups. Then the upper bounds are:

- on the number of colors :

$$k \leq \sum_{w \in W} \chi_{G_w}$$

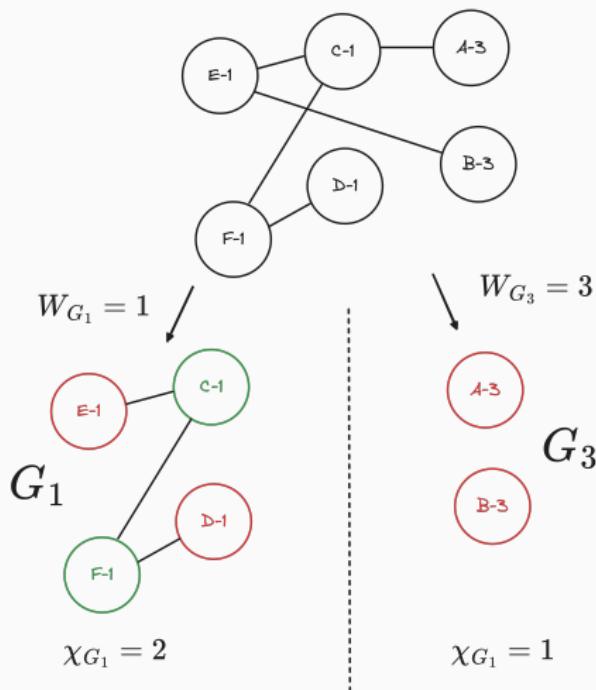
- on the score :

$$f(S^*) \leq \sum_{w \in W} w \times \chi_{G_w}$$

with :

- $W = \{w(v) \mid v \in V\}$ the set of weight values of G .
- $G_w = (V_w, E_w)$ the subgraph of G induced by the weight w .
- χ_{G_w} the chromatic number of G_w .

Bounds WVCP - Example



Upper Bound on the number of colors :

$$k \leq \sum_{w \in W} \chi_{G_w}$$

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

$$k \leq 2 + 1$$

$$k \leq 3$$

Upper Bound on the score :

$$f(S^*) \leq \sum_{w \in W} w \times \chi_{G_w}$$

$$f(S^*) \leq W_{G_1} \times \chi_{G_1} + W_{G_3} \times \chi_{G_3}$$

$$f(S^*) \leq 1 \times 2 + 3 \times 1$$

$$f(S^*) \leq 5$$

Bounds WVCP - Results

Instance	V'	density	h_W	$\Delta + 1$	bounds colors		bounds score	
					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

Bounds WVCP - Impact on Primal Model

instance	BKS	primal		primal		primal all bounds	
		score	time(s)	ub	color		
DSJC125.1g	23	<u>23*</u>	862	<u>23*</u>	435	<u>23*</u>	451
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
#BKS		101/188		105/188		107/188	
#Optimal		72/188		75/188		95/188	

Bounds WVCP - Conclusion

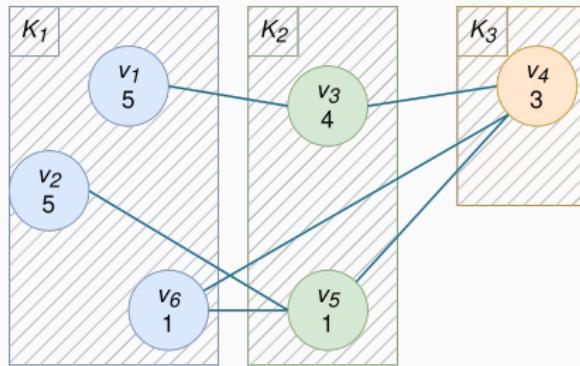
- 88/188 instances with a better upper bound on the number of colors compared to $\Delta(G) + 1$
↪ On average 13 fewer colors
- In practice :
 - Better results with a CP model Goudet *et al.* [2023]
↪ 95 proven optimal instances versus 72 without the bounds
 - Low impact on an MCTS

Three Constraint Programming Models for WVCP

Definitions

- P_κ denotes the problem of determining the existence of a solution to P that uses a number of colors smaller than or equal to κ .
- Solutions to P_κ are modeled as maps $s : [V] \mapsto K$ where $K = \{1, \dots, \kappa\}$.
- A total ordering \geq_w over V is defined which is consistent with the descending order of weights ($u \geq_w v \rightarrow w(u) \geq w(v)$ for $u, v \in V$).
- A solution is d-sorted if non-empty colors start from rank 1 and are sorted consistently with the ordering \geq_w of their dominant vertices.
- \mathcal{S}_{P_κ} : set of d-solutions using a number of colors smaller than κ .

Example of d-sorted solution



Primal model for P_κ

minimize x^o s.t.

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

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

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

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

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

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

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

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

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

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

Experimental Settings

- Intel Xeon ES 2630, 2.66 GHz.
- OR-Tools Perron et Furnon [2022] solver.
- Heuristics *first-fail* combined with domain bisection.
- Time limit of 1 hour for each run on a single CPU.

Primal model results and impact of pre-computed bounds

instance	BKS	primal		primal ub color		primal all bounds	
		score	time(s)	score	time(s)	score	time(s)
DSJC125.1g	23	<u>23*</u>	862	<u>23*</u>	435	<u>23*</u>	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	<u>68*</u>	1893	<u>68*</u>	1729
inithx.i.1	569*	569	tl	569	tl	<u>569*</u>	54
le450_15a	212	245	tl	234	tl	234	tl
le450_25b	307	307	tl	307	tl	<u>307*</u>	322
mulsol.i.5	367*	367	tl	367	tl	<u>367*</u>	31
queen10_10	162	170	tl	169	tl	169	tl
p42	2466*	2480	tl	2466	tl	<u>2466*</u>	2908
r30	9816*	9831	tl	9831	tl	9831	tl
nb bks reached		101/188		105/188		107/188	
nb optim		72/188		75/188		95/188	

Table 1: Primal model results and impact of pre-computed bounds.

Compact solutions

- A solution is compact, if the color value of each vertex cannot be reduced.
- Proposal of an algorithm, g_{P_κ} , compacting any d-sorted solution without deteriorating its score.

Theorem

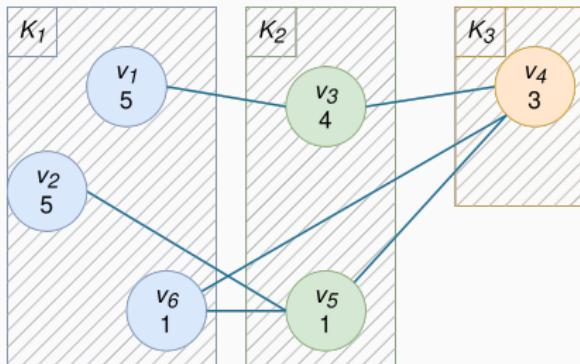
Let P_κ be a satisfiable WVCP instance. There exists $g_{P_\kappa} : \mathcal{S}_{P_\kappa} \mapsto \mathcal{S}_{P_\kappa}$ such that, for all $s \in \mathcal{S}_{P_\kappa}$, $g_{P_\kappa}(s)$ is compact, $f(g_{P_\kappa}(s)) \leq f(s)$ and $g_{P_\kappa}(g_{P_\kappa}(s)) = g_{P_\kappa}(s)$.

Corollaries

1. Reduction of the domain of each variable v to $\{1, \dots, \min(\kappa, \Delta(v) + 1)\}$.
2. New global constraint to achieve compactness

Example of compact solution

- This d-sorted solution is compact since neither v_3 , v_4 nor v_5 may be left-shifted.
- If (v_3, v_4) and (v_5, v_6) were not part of the graph, then v_5 and v_4 could be left-shifted to colors K_1 and K_2 respectively to compact the solution.



Enforcing solution compactness

- **Definition:** Let y be an integer domain variable and $[x_1, \dots, x_n]$ be a vector of positive integer domain variables ($n \geq 0$).

$\text{MAX_LEFT_SHIFT}(y, [x_1, \dots, x_n])$ holds iff

$$y = \min_{k=1..n+1}(\{k \mid \forall i = 1..n : x_i \neq k\}).$$

- **New global constraint for the primal model:**

$$\forall v_i \in V : \text{MAX_LEFT_SHIFT}(x_i^U, [x_j^U \mid v_j \in N(v_i)]) \quad (\text{P11})$$

Implementation of `max_left_shift`

Decomposition of `MAX_LEFT_SHIFT` using constraints (M1-M3) with global constraint `NVALUE` Bessiere *et al.* [2006]:

$$\text{MAX_LEFT_SHIFT}(y, [x_1, \dots, x_n]) \equiv \forall i \in \{1, \dots, n\} : z_i \in \{0, \dots, n+1\} \quad (\text{M1})$$

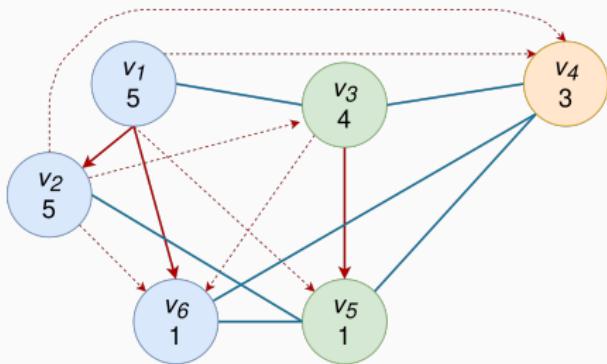
$$\forall i \in \{1, \dots, n\} : z_i = (y > x_i) \times x_i \quad (\text{M2})$$

$$\text{NVALUE}(y, [0, z_1, \dots, z_n]) \quad (\text{M3})$$

Impact of this symmetry breaking rule on the results

instance	BKS	primal		primal + P11	
		score	time(s)	score	time(s)
DSJC125.1g	23	<u>23*</u>	862	<u>23*</u>	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 reached		101/188		102/188	
nb optim		72/188		76/188	

Dual graph from Cornaz et Jost [2008]



Primal

$$\text{Score}_P = w(v_1) + w(v_3) + w(v_4) = 12$$

Dual

$$\text{Score}_D = w(v_1) + w(v_3) + w(v_4) = 7$$

$$\text{Score}_P + \text{Score}_D = \sum_{i=1}^n w(v_i) = 19$$

- **Set of arcs of the dual graph:**
 $\overrightarrow{E^c} = \{ij \mid v_i, v_j \in V \wedge \{v_i, v_j\} \notin E \wedge v_i \geq_w v_j\}.$
- **Solution in the dual model:** a set of simplicial stars that span disjoint subsets of nodes.

Dual model for P_κ - adaptation from Cornaz *et al.* [2017]

maximize y^o s.t.

$$\forall ij \in \vec{E^c} : y_{ij}^A \in \{0, 1\} \quad (D1)$$

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

$$y^o = \sum_{ij \in \vec{E^c}} w(v_j) \times y_{ij}^A \quad (D3)$$

$$\forall ij, ik \in \vec{E^c} \text{ s.t. } \{jk, kj\} \cap \vec{E^c} = \emptyset :$$

$$y_{ij}^A + y_{ik}^A \leq 1 \quad (D4)$$

$$\forall ij, jk \in \vec{E^c} : y_{ij}^A + y_{jk}^A \leq 1 \quad (D5)$$

$$\forall hj, ij \in \vec{E^c} : y_{hj}^A + y_{ij}^A \leq 1 \quad (D6)$$

$$\forall v_i \in V : z_i^V \in \{0, 1\} \quad (D7)$$

$$\forall v_i \in T : z_i^V = 1 - \max_{(h,i) \in \vec{E^c}} (y_{hi}^A) \quad (D8)$$

$$\forall v_i \in V \setminus T : z_i^V = 1 \quad (D9)$$

$$\sum_i z_i^V \leq \kappa \quad (D10)$$

Joint model

Joint Model = Primal + Dual + J1-J4 channeling constraints.

minimize x^o s.t.

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

$$\text{GCC}([x_k^D \mid k \in K], V, [z_i^V \mid v_i \in V]) \quad (\text{J2})$$

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

$\forall v_i \in V, v_j \in \overline{N(v_i)}$ s.t. $v_j \geq_w v_i$:

$$\left(\bigwedge_{v_h \in N(v_i) \cap \overline{N(v_j)}} x_h^U \neq x_j^U \right) \Rightarrow x_i^U \leq x_j^U \quad (\text{J4})$$

Results of the different CP models

instance	BKS	primal		primal + P11		dual		joint + J4	
		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 reached		101/188		102/188		79/188		112/188	
nb optim		72/188		76/188		68/188		100/188	

Table 3: Results of the different CP models.

New optimality proofs

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

instance	$ V $	BKS	score	time(s)	instance	$ V $	BKS	score	time(s)
DSJC125.1gb	125	90	<u>90*</u>	25	myciel7gb	191	109	<u>109*</u>	69
DSJC125.1g	125	23	<u>23*</u>	11	myciel7g	191	29	<u>29*</u>	241
DSJR500.1	500	169	<u>169*</u>	66	queen9_9g	81	41	<u>41*</u>	509
myciel6gb	95	94	<u>94*</u>	17	queen10_10g	100	43	<u>43*</u>	820
myciel6g	95	26	<u>26*</u>	17	le450_25b	450	307	<u>307*</u>	322

Table 4: New optimality proofs for difficult benchmark instances.

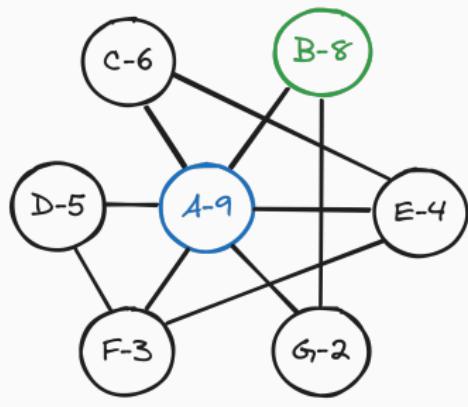
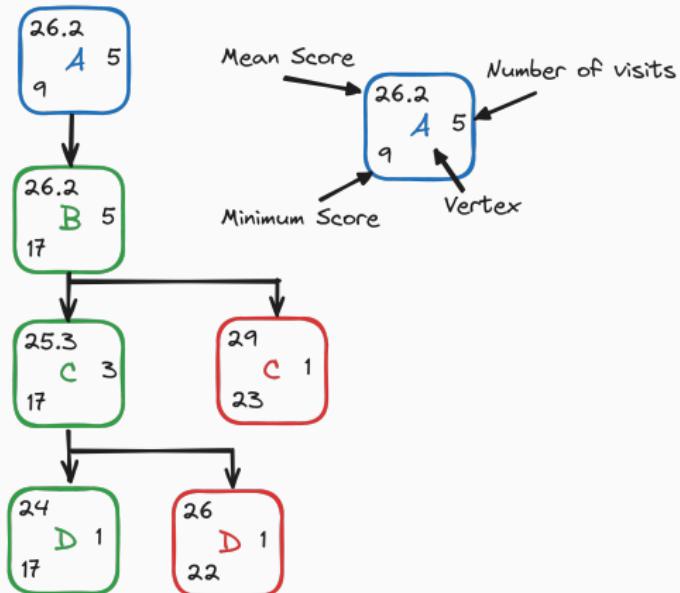
Conclusion

- Iterative reduction procedure and new upper bounds on the score and the number of colors.
 - Reduce the search space.
- Three competitive and complementary CP models.
- 10 new optimality proofs for difficult benchmark instances.
- Future work: investigate possible hybridizations of the CP models with metaheuristics.

MCTS

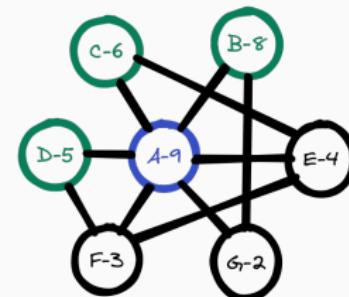
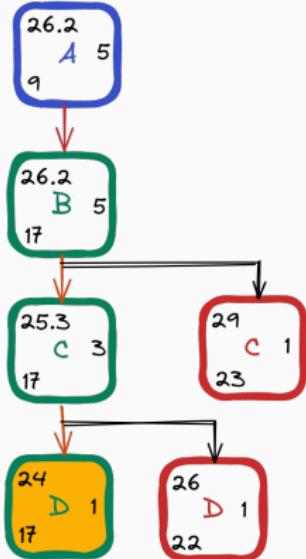
MCTS - A Search Tree and a Graph

Search Tree



Graph

MCTS - Phase 1 : Selection



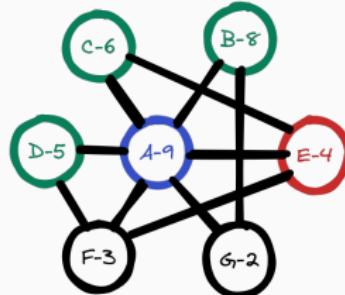
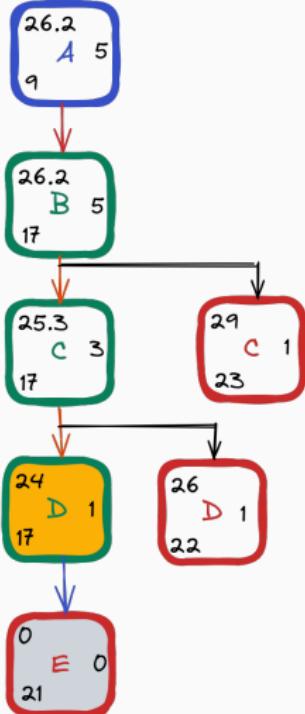
Selection

Use UCB score (Jooken et al. [2023])

Exploitation + coefficient * Exploration

$$\frac{\text{rank}(C_{t+1}^i)}{\sum_{i=1}^I \text{rank}(C_{t+1}^i)} + c \times \sqrt{\frac{2 * \ln(\text{nb_visits}(C_t))}{\text{nb_visits}(C_{t+1}^i)}}$$

MCTS - Phase 2 : Expansion

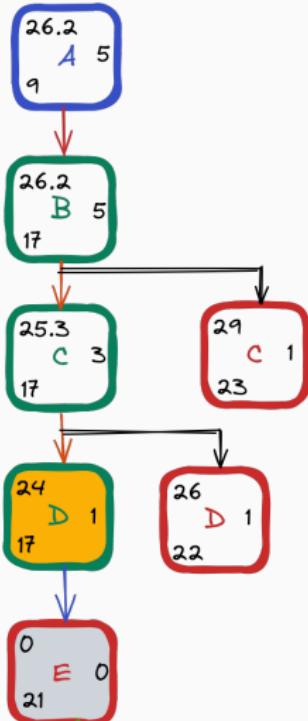


Expansion

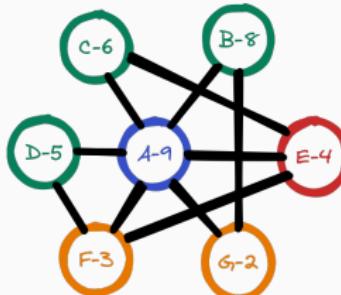
Open a new node:

- 1 - Use a color already in use in the branch
- 2 - Open a new color (limit with bounds and score)

MCTS - Phase 3 : Simulation



↓
F(S)=24

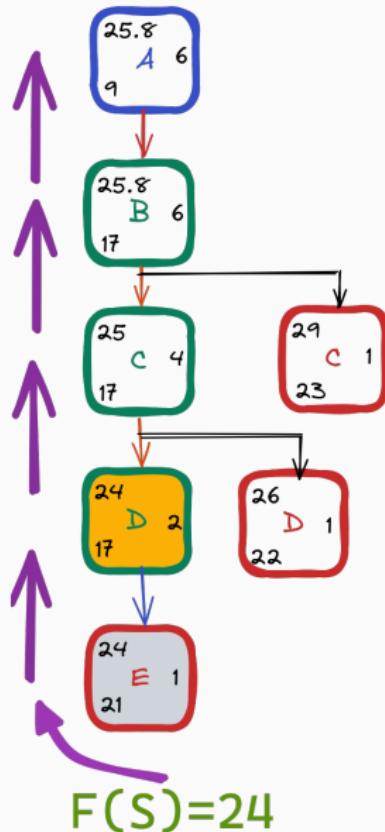


Simulation

Complete the coloring

- 0 - Purely random choice
- 1 - Greedy Algorithm
- 2 - Local Search (Grelier *et al.* [2022])
- 3 - Hyperheuristics (Grelier *et al.* [2023])

MCTS - Phase 4 : Update

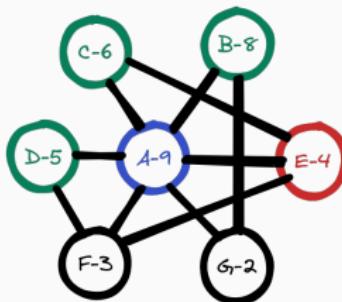
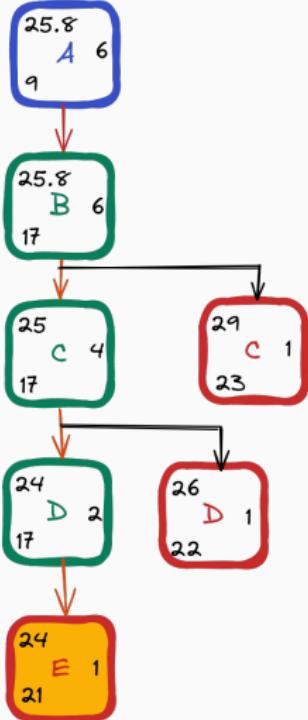


Update

Backpropagation of the score to update :

- Mean score
- Number of visits

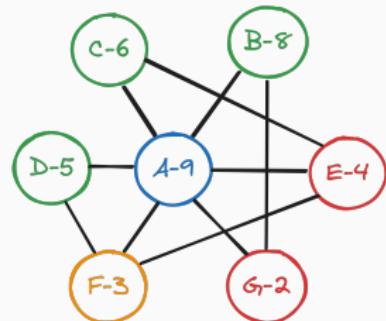
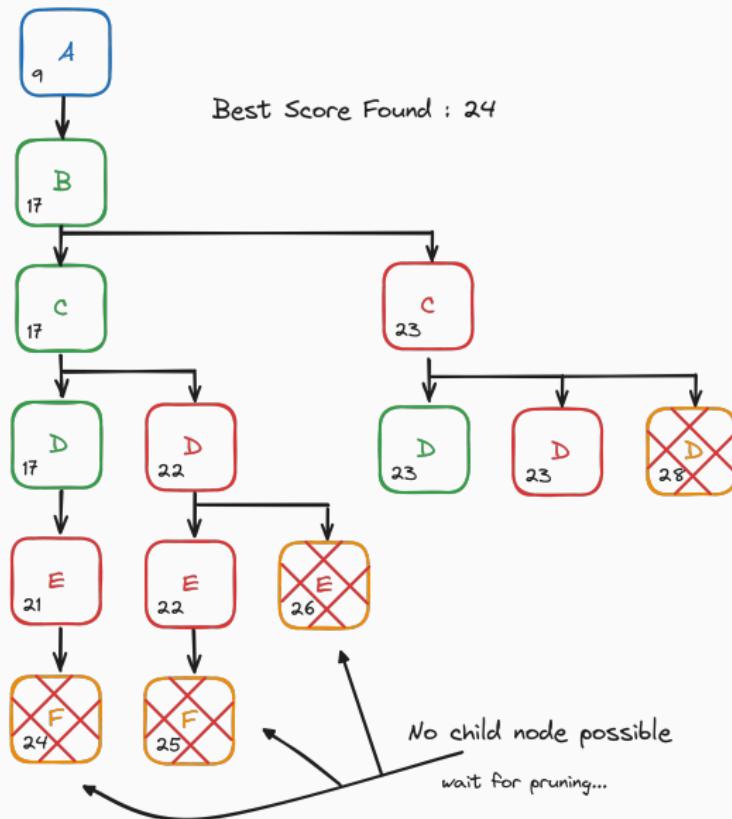
MCTS - Phase 1 : Selection...



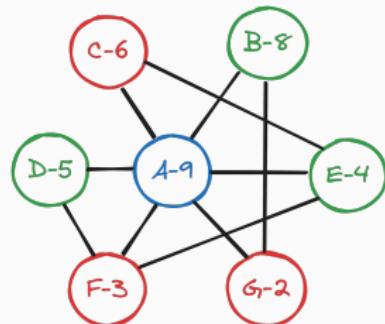
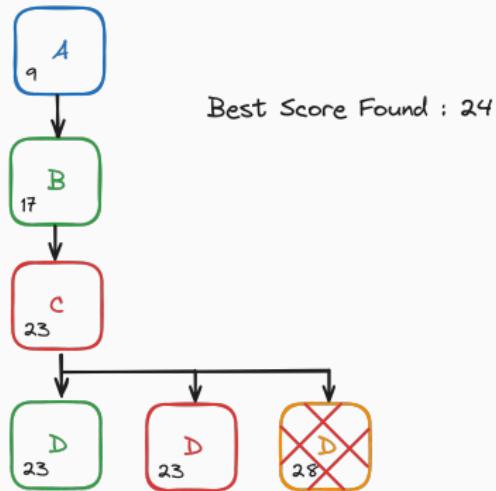
Selection

New Iteration...

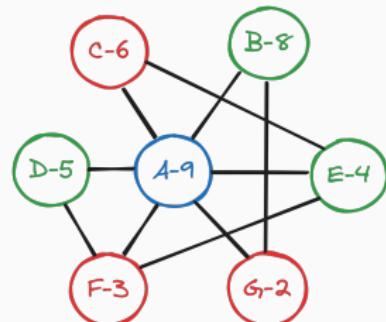
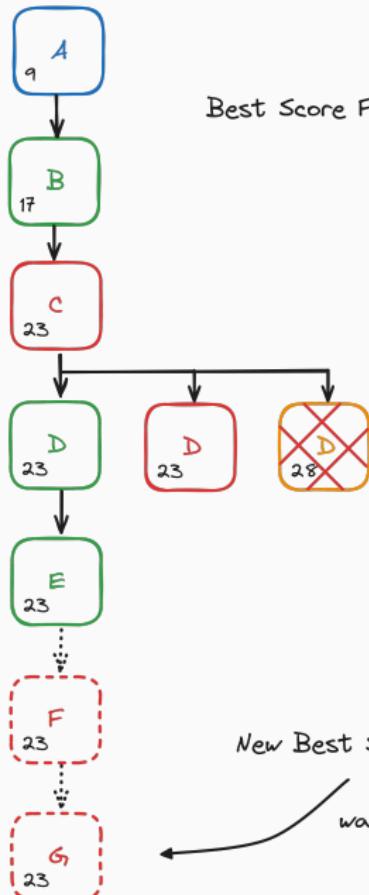
MCTS - Pruning 1



MCTS - Pruning 2



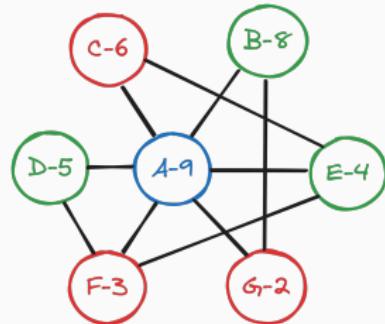
MCTS - Pruning 3



MCTS - Pruning 4

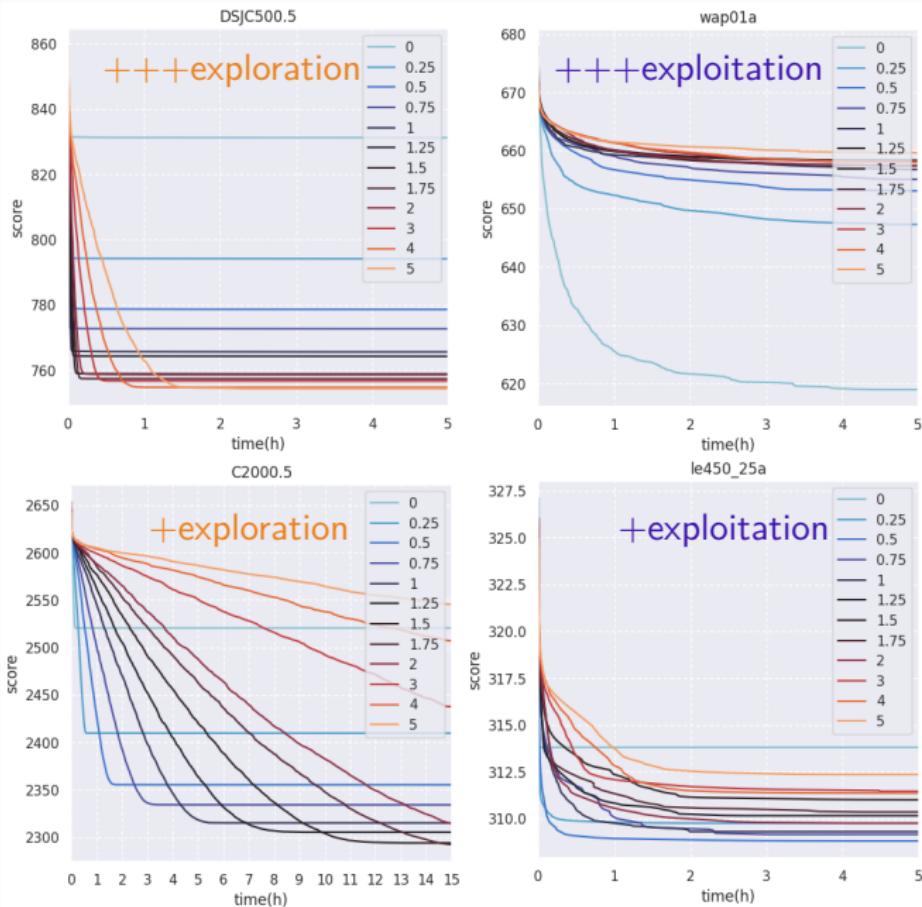


Best Score Found : 23



Tree fully explored -> 23 is the optimal score!

MCTS + Greedy - Exploration vs. Exploitation Coefficient



MCTS + Greedy - Results GCP

1 point/instance if mean significantly better for row method than column method (Wilcoxon signed-rank test, p-value < 0.001)

/244 instances	MCTS+R	MCTS+C	MCTS+D	MCTS+DSatur	MCTS+RLF	NRPA	TabuCol	#BKs	#Best	#Best Avg	#Optimal
MCTS+R	-	0	2	0	1	43	9	172	172	151	109
MCTS+C	80	-	20	10	2	45	10	180	181	178	109
MCTS+D	78	24	-	10	0	48	9	178	178	179	110
MCTS+DSatur	81	46	46	-	10	55	12	189	189	189	109
MCTS+RLF	80	55	60	34	-	63	10	190	191	192	110
NRPA	78	53	59	40	28	-	11	202	206	159	0
TabuCol	78	60	64	50	52	46	-	213	235	229	0

NRPA : Cazenave *et al.* [2021] - TabuCol : Hertz et Werra [1987]

MCTS + Greedy - Results GCP

instance	BKS	DSatur		MCTS+DSatur			RLF		MCTS+RLF			NRPA			TabuCol		
		best	time	best	mean	time	best	time	best	time	best	mean	time	best	mean	time	
C2000.5	145	214	0	210	210.7	3569	195	163	192	2482	211	213.1	2596	162	163.2	1802	
C2000.9	408	581	1	565		265	511	343	511	322	587	628.5	3185	412	413	2758	
C4000.5	259	392	3	383	383.1	3342	356	1314	355	2621	397	401.4	2545	304	305.3	2303	
DSJC500.1	12	16	0	14		3084	15	0	14	3	13	13.6	1945	12		77	
DSJC500.5	47	70	0	64		412	61	2	58	22	58	58.9	2372	49		476	
DSJC500.9	126	175	0	159		2917	151	4	148	356	148	149.8	1197	126	126.5	2328	
DSJC1000.1	20	28	0	26		1	24	3	23	1161	24		569	21		0	
DSJC1000.5	82	123	0	116		200	108	19	105	2062	110	112	1414	88	88.1	1321	
DSJC1000.9	222	316	0	303		1579	280	39	273	2188	291	294.9	2821	224	225.1	3181	
flat1000_60_0	60*	123	0	114		2938	106	19	103	1578	110	111.3	2178	60		237	
flat1000_76_0	76*	119	0	115		1847	105	19	104	124	110	111.5	1271	86	87.4	3320	
GEOM120a	16*	17	0	16		0	16	0	16	0	16		0	16		0	
GEOM120b	16*	17	0	16		0	17	0	16	0	16		3	16		0	
GEOM120	11*	11	0	11*		0	11	0	11*	0	11		211	11		0	
latin_square_10	97	151	0	126		96	122	19	122	17	122	124.9	1758	100	101.2	1976	
le450_25a	25*	25	0	25		0	25	0	25	0	25		0	25		0	
le450_25b	25*	25	0	25		0	25	0	25	0	25		0	25		0	
le450_25c	25*	30	0	27		597	28	0	27	7	26		636	26		0	
le450_25d	25*	30	0	27		201	28	0	27	4	26		958	26		0	
queen10_10	11*	14	0	11		1524	13	0	12	0	11	11.2	1019	11		0	
queen11_11	11*	16	0	13		0	14	0	13	0	12	12.9	1848	12		0	
queen12_12	12*	16	0	14		30	15	0	14	0	14		2	13		0	
r1000.5	234	278	0	246		3080	251	16	247	1674	239	240.4	2276	244	246	3404	
wap01a	41*	54	0	44		3230	47	6	44	567	44	45	2839	42	43.2	1306	
wap02a	40*	48	0	44		6	44	5	43	40	44	44.2	1869	41	41.2	362	
wap03a	43	55	0	51		611	51	41	49	194	54	54.5	1577	44	46.2	1125	
wap04a	41	49	0	46		119	47	40	45	187	48	48.8	1608	42	43.3	2997	
#BKS		148		189		171		190			202				213		
#Best		148		189		171		191			206				235		
#Best Avg		148		189		171		192			159				229		
#Optimal		0		109		0		110			0				0		

MCTS + Greedy - Results WVCP

1 point/instance if mean significantly better for row method than column method (Wilcoxon signed-rank test, p-value < 0.001)

/188 instances	MCTS+R	MCTS+C	MCTS+D	MCTS+DSatur	MCTS+RLF	AFISA	RedLS	ILS-TS	#BKS	#Best	#Best Avg	#Optimal
MCTS+R	-	0	1	1	32	22	46	0	75	75	56	48
MCTS+C	122	-	23	32	92	77	85	0	114	114	75	49
MCTS+D	121	44	-	30	97	98	103	1	92	92	91	48
MCTS+DSatur	121	48	35	-	99	93	107	0	93	93	93	47
MCTS+RLF	87	14	14	14	-	58	79	0	70	70	70	45
AFISA	96	33	46	50	89	-	53	0	114	114	45	0
RedLS	79	51	60	57	88	41	-	14	112	131	47	0
ILS-TS	126	101	94	95	118	104	96	-	159	170	171	0

AFISA : Sun *et al.* [2018] - RedLS : Wang *et al.* [2020] - ILS-TS : Nogueira *et al.* [2021]

MCTS + Greedy - Results WVCP

instance	BKS	MCTS+C			MCTS+D			MCTS+DSatur			AFISA			RedLS			ILS-TS		
		best	mean	time	best	mean	time	best	mean	time	best	mean	time	best	mean	time	best	mean	time
C2000.5	2144	2505	2537.7	3422	2385	1685	2397	2398.8	3390	2384	2403.4	3601	2167	2193.8	2403	2237	2266.4	3498	
C2000.9	5477	6233	6272.9	3575	6125	6147.8	3238	6275	163	6582	6650.1	0	5502	5528.1	3303	5910	5969.9	3587	
DSJC250.1	127	134	141.4	13	141	4	139		1151	129	133.6	3294	130	131.6	1	127	127.8	1608	
DSJC250.5	392	422	429.4	14	427	92	421		705	411	424.2	541	398	401.2	103	393	397.6	2567	
DSJC250.9	934*	973	988.7	2793	986	16	984		559	949	976.1	232	934	935.6	718	936	942.1	3053	
DSJC500.1	184	203	208.3	148	203	638	203		2943	198	201.5	1817	187	201.9	537	188	188.8	1744	
DSJC500.5	685	754	765.6	136	755	635	775	780.9	3542	762	778.1	1307	706	716.1	2840	724	735.5	1744	
DSJC500.9	1662	1771	1787.4	113	1794	171	1795	1797.1	3453	1744	1775.5	652	1670	1675.1	945	1720	1742	3039	
DSJC1000.1	300	334	337.4	1618	333	2823	340		2474	319	325	2182	305	307.2	1235	305	307.4	1574	
DSJC1000.5	1185	1271	1293.6	1668	1318	1437	1338		203	1308	1330	3531	1198	1213.7	2381	1245	1269.2	231	
DSJC1000.9	2836	3040	3070.4	978	3078	2863	3172		3338	3066	3107	3601	2840	2858.5	2953	3026	3066.8	3580	
DSJR500.1	169*	169	46	177	0	176	21	169		54	171	184.5	0	169		0			
flat1000_50_0	924	1236	1255.8	1452	1251	1251.1	3037	1303	1843	1267	1293.3	2736	1155	1173.8	3238	1222	1235	1712	
flat1000_60_0	1162	1275	1295.8	1478	1260	1260.7	3424	1343	311	1309	1323.2	3584	1191	1205.7	1080	1250	1270	125	
flat1000_76_0	1165	1252	1269.7	1477	1244	1248.2	3557	1313	1798	1288	1304.4	3278	1176	1194	1107	1232	1247.5	1198	
GEOM120	72*	72	73	0	72	0	72		36	72	73	0	72	75.2	0	72		0	
latin_square_10	1480	1721	1757.1	966	1726	1726.8	3418	1805	2360	1607	1652.4	2257	1505	1523	2369	1559	1581.2	1368	
le450_25a	306	307	310	3364	312	5	310		1919	311	316.3	2699	306	307.4	503	306		174	
le450_25b	307*	309	309.1	993	309	0	309		224	308	312.6	1891	307	313.4	56	307		10	
le450_25c	342	365	372.7	2844	369	950	376		1572	355	364.4	3436	351	354.8	43	348	351.8	3207	
le450_25d	330	354	358.8	2802	364	1657	369		1411	351	357.8	1630	332	338.9	154	339	342.5	1999	
myciel7gb	109*	117	118.3	476	109	42	109		38	109	111.8	1129	109	116.4	0	109		4	
myciel7g	29*	29	29.8	578	29	0	29		0	29	30.7	2542	29	29.8	244	29		0	
queen10_10	162	165	169.1	986	171	0	169		8	165	166.9	524	162	166.2	504	162		13	
queen11_11	172	178	180.4	941	180	80	179		50	179	181.1	722	174	177.1	761	172	172.6	1820	
queen12_12	185	189	196.8	2	193	1102	197		1823	193	196.7	2026	188	190.2	7	185	186.1	1261	
wap01a	545	657	659.8	2965	599	3272	595		2904	653	664.5	3413	563	688.4	252	548	552	3263	
wap02a	538	645	648.7	1301	588	3265	590		262	658	666.2	3416	552	586.5	589	540	542.7	2658	
wap03a	562	746	749	803	653	1096	647		1805	767	788	0	569	573.1	2893	576	578.6	3102	
wap04a	563	755	757.8	3085	649	603	643		828	773	792.9	0	564	572.5	3280	569	574.6	3529	
wap05a	541	583	591.1	1867	566	172	565		2882	563	572.5	3378	543	544.5	970	541	543.4	3306	
wap06a	516	591	595.4	2280	566	9	561		664	559	567.2	2859	518	590.4	1005	519	522.4	1123	
wap07a	555	693	697.2	3186	635	165	639		986	670	683.2	3584	729	745.7	0	564	569.5	2680	
wap08a	529	664	672	3539	612	2401	604		2583	654	665.3	3402	536	613.9	3035	543	549.5	3585	
p42	2466*	2466	2466.8	33	2466	0	2466	4	2466	2474.8	1180	2466	2522.5	0	2466		4		
r30	9816*	9816	9819.6	11	9818	0	9818	0	9826	9960	2378	9836	9988.2	1	9816		5		
#BKS		114			92			93		114			112			159			
#Best		114			92			93		114			131			170			
#Best Avg		75			91			93		45			47			171			
#Optimal		49			48			47		0			0			0			

MCTS + Greedy - Results

GCP

- vs NRPA (Cazenave *et al.* [2021]):
 - NRPA reaches more best scores
 - NRPA better on geometric or patterned instances
 - MCTS gets better averages
 - MCTS better on random, denser/larger instances
- vs TabuCol (Hertz et Werra [1987]³):
 - TabuCol remains better in general

WVCP

- vs AFISA (Sun *et al.* [2018]) and RedLS (Wang *et al.* [2020]) :
 - MCTS better on small/medium instances
- vs ILS-TS (Nogueira *et al.* [2021]) :
 - ILS-TS remains better in general

³optimized by Moalic et Gondran [2018]

MCTS + Hyperheuristics - Criteria

Selection criteria

- **Random** Uniform random choice
- **Deleter** Delete the least performing operators (o)
- **Roulette** Goëffon *et al.* [2016] Random selection weighted by rewards (r)

$$proba[o] = p_{min} + (1 - |O| * p_{min}) * \frac{r[o]}{\sum r}$$

- **Pursuit** Goëffon *et al.* [2016] Selection in favor of the best operator (b)

$$\begin{cases} proba[b] = proba[b] + \beta(p_{max} - proba[b]) \\ proba[o] = proba[o] + \beta(p_{min} - proba[o]) \end{cases}$$

- **UCB** Focusing on the best while encouraging exploration

$$score[o] = r[o] + c * \sqrt{2 * \frac{\log(\sum visits)}{visits[o]}}$$

- **NN** Recommendation of a neural network on a raw solution with Deep Sets (Zaheer *et al.* [2017])

MCTS+Hyperheuristics - Results WVCP

instance	BKS	RedLS			MCTS+RedLS			ILS-TS			Random			Deleter			Roulette			UCB			Pursuit			NN			
		best	avg	time	best	avg	time	best	avg	time	best	avg	time	best	avg	time	best	avg	time	best	avg	time	best	avg	time	best	avg	time	
C2000.5	2144	2167	2193.8	2403	2354	2369	2091	2237	2266.4	3498	2336	2347.1	2870	2331	2342.2	2952	2330	2347.3	1845	2335	2349.8	1958	2332	2340.8	1271	2337	2346.2	6924	
C2000.9	5477	5502	5521.3	3303	6060	6093.6	1107	5910	5969.9	3857	6035	6110.6	1968	6067	6105.4	492	6073	6109.4	2542	6077	6111.4	2952	6069	6103.4	205	6109	6128.5	7363	
DSJC250.1	127	130	131.6	1	127	127.5	1785	127	127.8	1609	127	128.1	1604	127	127.7	2271	127	127.8	2593	127	127.6	2125	127	127.7	1712	127	127.6	2094	
DSJC250.5	392	398	401.2	103	396	397.9	3045	393	397.6	2567	397	399.1	1242	393	397.5	3558	397	398.6	1971	394	396.4	1746	396	397.8	1143	396	398.2	2701	
DSJC250.9	934*	934	935.6	719	935	935.9	2403	936	942.1	3053	936	936.8	1933	935	936.3	2463	935	936.2	1802	935	936.1	1856	935	936.2	3054	934	935.6	2451	
DSJC500.1	184	187	201.9	537	187	187.8	1765	188	188.8	1744	186	188.4	2079	186	188.2	3773	187	188.4	2503	187	188.4	3861	187	188.5	2079	187	188.2	3043	
DSJC500.5	682	700	716.1	2840	716	719.2	3234	724	735.5	1744	715	720.2	964	715	719	3234	715	720.6	1045	712	720.6	880	715	719.8	715	713	718.9	631	
DSJC500.9	1662	1670	1675.1	946	1667	1694.7	726	1720	1742	3039	1694	1697.5	2229	1685	1692.9	847	1689	1694.5	2622	1691	1695.4	1881	1690	1694.3	1227	1688	1693.5	3134	
DSJC1000.1	303	305	307.2	1235	307	307.9	1799	305	307.4	1575	304	306.2	3570	303	305.8	1190	302	305.6	588	304	305.9	2835	305	305.4	2151	303	305.1	4617	
DSJC1000.5	1185	1198	1213.7	2381	1244	1248.5	1197	1245	1269.2	231	1245	1251.4	756	1238	1246.6	4128	1241	1251.4	2142	1242	1250	1239	1247.8	1554	1240	1247.6	3159		
DSJC1000.9	2836	2840	2858.5	2953	2974	2992.6	651	3026	3066.8	3580	2982	2995.9	1890	2983	2992.5	3486	2984	2995.2	2982	2983	2995.1	504	2983	2995.2	2058	2976	2993.2	4499	
flu1000_50_0	924	1155	1173.8	3238	1199	1202.6	334	1222	1235	1713	1203	1210.7	3045	1191	1202.6	630	1201	1208.1	945	1205	1210.6	756	1198	1207.8	3654	1196	1204.6	4370	
flu1000_60_0	1162	1191	1205.7	1080	1238	1245.4	1401	1250	1270	125	1235	1249.5	3045	1242	1247.5	525	1241	1248.2	1095	1239	1247.8	882	1239	1245.3	1701	1238	1245.1	1393	
flu1000_76_0	1165	1176	1194	1107	1214	1222	1176	1232	1247.5	1198	1210	1224.9	2751	1217	1222.1	1470	1211	1223.8	168	1218	1226.2	504	1214	1223.1	2121	1211	1222.5	4602	
GEOM120	105*	105	109.2	9	105	105	9	105	105	0	105	105	9	105	105	6	105	105	7	105	105	8	105	105	7	105	105	8	
GEOM1200	35*	35	35	0	35	35	1	35	35	0	35	35	4	35	35	5	35	35	4	35	35	5	35	35	3	35	35	3	
GEOM1200	72*	72	75.2	0	72	72	14	72	72	0	72	72	3	72	72	8	72	72	5	72	72	4	72	72	7	72	72	2	
latin_square_10	1480	1505	1523	2369	1533	1544.8	1938	1559	1581.2	1369	1546	1552.9	1406	1533	1550.4	1606	1540	1550.6	3268	1522	1548.6	456	1529	1547.2	798	1538	1546.2	4558	
le450_25a	306	306	307.4	504	306	306	313	306	306	305	306	306	453	306	306	305	306	306	349	306	306	406	306	306	306	306	306	406	
le450_25b	307*	307	313.4	504	307	307	87	307	307	10	307	307	53	307	307	50	307	307	50	307	307	50	307	307	50	307	307	47	
le450_25c	342	351	354.8	44	348	349.4	383	348	351.8	3207	349	350.2	945	347	349.4	2147	350	350.3	1543	348	349.8	2052	347	349.5	2889	347	349.4	3388	
le450_25d	330	332	338.9	154	334	334.9	1439	330	334.5	1999	336	336.8	1403	334	336.1	1393	335	336.7	1532	333	336.9	2394	335	335.9	1386	335	336	2054	
queen11_11	172	174	177.1	762	172	173.2	1935	172	172.6	1821	172	172.9	2011	172	172.8	1952	172	172.8	1960	172	172.8	1923	172	172.8	1770	172	172.7	2632	
queen12_12	185	188	190.2	8	185	186.6	432	185	186.1	1261	186	186.7	1379	186	186.3	1506	186	186.5	1723	186	186.6	2051	186	186.4	1502	186	186.4	1369	
queen13_13	194	195	198.7	1528	194	194.6	1605	194	195.7	3476	194	194.9	2997	194	194.5	1668	194	194.8	2707	194	194.5	2734	194	194.6	1352	194	194.3	2140	
queen14_14	215	217	224.8	2	216	216.9	918	216	217.3	218	217	213.8	408	216	216.9	2373	216	217	218	3268	216	217	3268	216	217	3268	216	217	3268
queen15_15	223	227	229.7	1110	225	226.2	718	227	228.4	2167	226	226.9	2274	226	226.2	1292	226	226.6	2088	225	226.4	2408	224	226.1	2500	225	226.2	2647	
queen16_16	234	237	239.9	91	237	237.7	1813	238	240.2	2493	237	238.3	1998	236	237.4	2627	237	237.7	2104	236	237.8	3720	236	237.6	3726	236	237.7	1544	
wap01a	545	563	688.4	252	561	566.6	2860	546	552	3263	560	593.0	2996	587	591.6	2462	567	598.5	362	570	591.0	3080	586	591.5	1672	567	598.9	440	
wap02a	538	552	586.5	589	551	556.6	90	540	542.7	2658	552	560.2	1215	554	558.9	945	551	558.8	945	549	558.5	2520	548	556.5	3105	551	559.1	5815	
wap03a	562	569	573.1	2993	587	588.8	2085	576	578.6	3102	587	589.5	1656	587	589.5	3404	587	589.8	2024	587	589.3	3268	586	588.9	2300	586	589.9	8953	
wap04a	563	564	572.5	3280	581	583.9	1287	560	574.6	3529	582	585.2	1906	583	584.8	2138	582	584.2	2525	582	584.1	2624	581	584.1	2277	582	585.1	1685	
wap05a	541	543	544.5	970	543	544.8	1320	541	543.4	3306	544	545.8	1139	544	545.7	1397	545	545.6	1929	545	545.6	1649	543	545.4	578	543	545.3	1571	
wap06a	516	518	590.4	1009	521	523.5	1332	519	522.4	1123	521	526.1	1188	522	529.7	270	523	527.1	756	522	526.6	972	524	528.3	2199	521	527.4	252	
wap07a	555	579	729	745.7	0	566	632.6	770	564	569.5	2680	567	595.9	210	600	610.6	945	569	608.5	140	604	610.2	3290	567	606.4	70	568	606	2904
wap08a	529	536	613.9	3035	543	545.8	144	543	549.5	3585	544	546.9	2844	544	551	288	545	546.9	1830	544	547.4	2736	544	547.1	828	545	547.9	1215	
p40	4984*	4987	5055.7	0	4984	4984	557	4984	4984	0	4984	4984	8	4984	4984	7	4984	4984	6	4984	4984	8	4984	4984	8	4984	4984	4	
p41	2688*	2718	2787.3	0	2688	2688	530	2688	2688	0	2688	2688	9	2688	2688	7	2688	2688	10	2688	2688	7	2688	2688	8	2688	2688	8	
p42	2466*	2466	2522.5	109	2466	2466	4	2466	2466	18	2466	2466	19	2466	2466	23	2466	2466	11	2466	2466	16	2466	2466	18				
r28	9407*	9410	9563	45	9407	9412	1573	9407	9407	9	9407	9407	30	9407	9407	26	9407	9407	37	9407	9407	26	9407	9407	25				
r29	8603*	8606	8817.6	0	8603	8609	539	8603	8603	1	8603	8603	23	8603	8603	17	8603	8603	30	8603	8603	17	8603	8603	21				
r30	9616*	9636	9988.2	1	9616	9824	2570	9816	9816	6	9816	9816	47	9816	9816	30	9816	9816	35	9816	9816	27	9816	9816	20				
#BKS / 188		112		152		159		156		156		156		156		156		156		156		156		156		157			
#Best Score / 188		127		153		164	</																						

MCTS + Hyperheuristics - Results WVCP

1 point/instance if mean significantly better for row method than column method (Wilcoxon signed-rank test, p-value < 0.001)

/188	AFISA	MCTS+AFISA	TW	MCTS+TW	RedLS	MCTS+RedLS	ILS-TS	MCTS+ILS-TS	Random	Deleter	Roulette	UCB	Pursuit	NN	#BKs	#Best	#Best Avg
AFISA	-	35	49	17	53	0	0	0	0	0	0	0	0	0	114	114	45
MCTS+AFISA	40	-	73	10	86	0	1	0	0	0	0	0	0	0	115	115	97
TW	40	40	-	25	48	1	0	3	2	2	2	2	1	2	99	99	53
MCTS+TW	74	73	78	-	101	5	1	0	0	0	0	0	0	0	132	132	98
RedLS	41	35	47	29	-	11	14	15	12	11	12	13	11	11	112	127	44
MCTS+RedLS	103	82	107	72	102	-	20	28	5	5	4	3	2	4	152	153	142
ILS-TS	104	82	106	75	96	19	-	18	10	11	10	9	11	11	159	164	159
MCTS+ILS-TS	102	82	103	73	92	15	2	-	0	1	0	1	1	1	155	155	152
Random	104	82	108	75	98	15	15	25	-	2	0	0	1	1	156	158	151
Deleter	104	82	108	75	99	14	17	23	5	-	0	0	0	1	156	161	156
Roulette	104	82	107	75	101	15	15	25	0	1	-	0	1	2	156	158	151
UCB	104	82	107	75	101	15	17	25	0	1	0	-	2	2	156	158	152
Pursuit	104	82	107	75	102	15	17	26	0	1	1	1	-	1	156	161	153
NN	104	82	107	75	101	15	18	26	4	1	0	1	0	-	157	160	153

MCTS + Hyperheuristics - Results WVCP

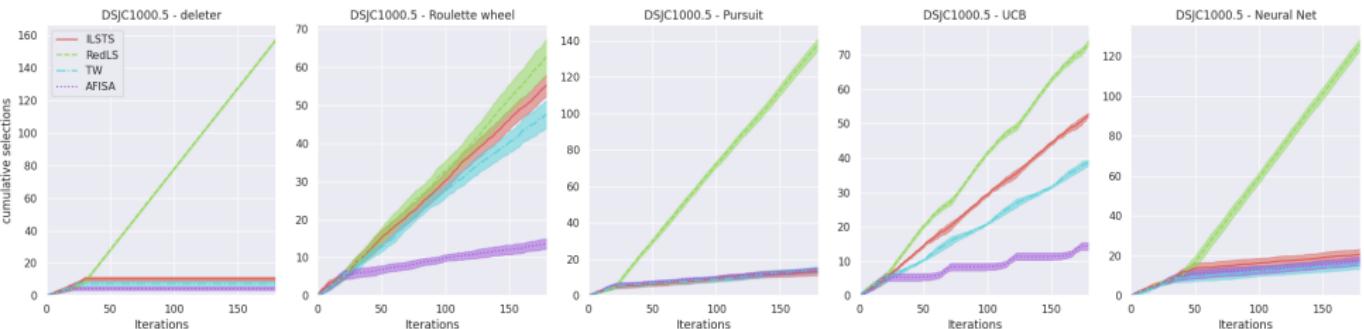
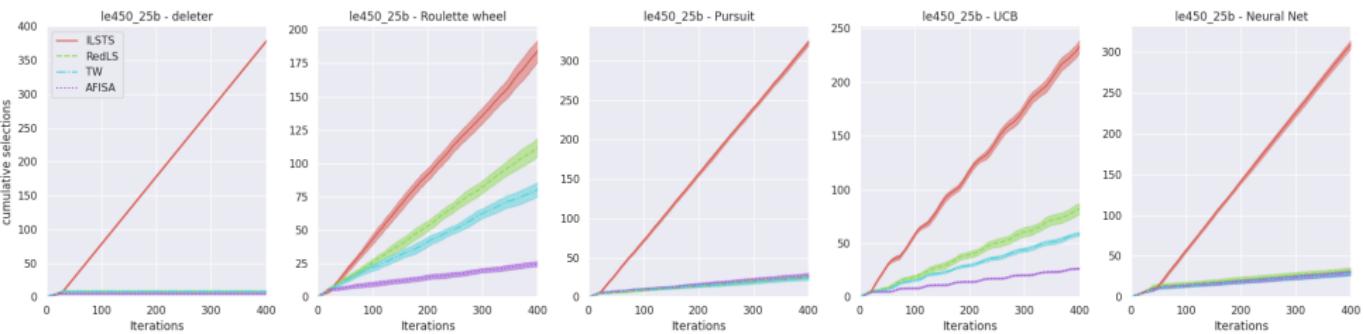
MCTS + LS (Local Search)

- MCTS+LS improves the number of significant differences to the average and the number of best scores for AFISA, TabuWeight and RedLS but not for ILS-TS
- LS alone better on about 20 instances (/188)

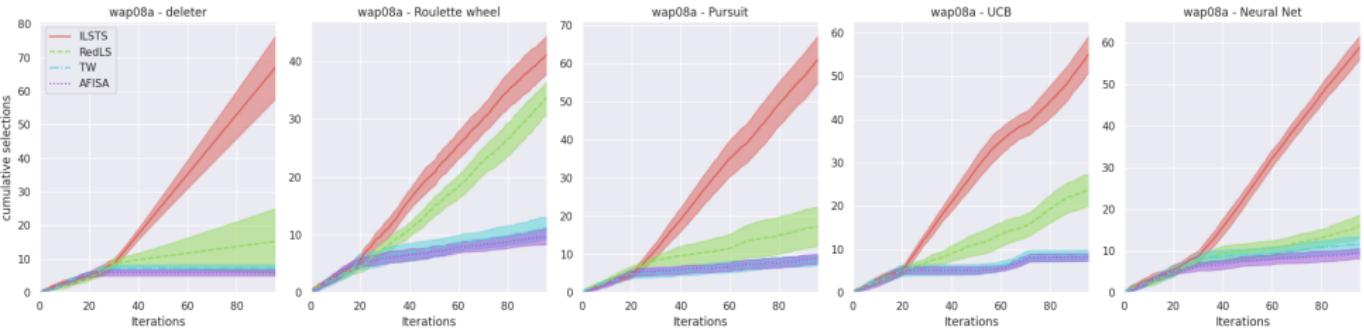
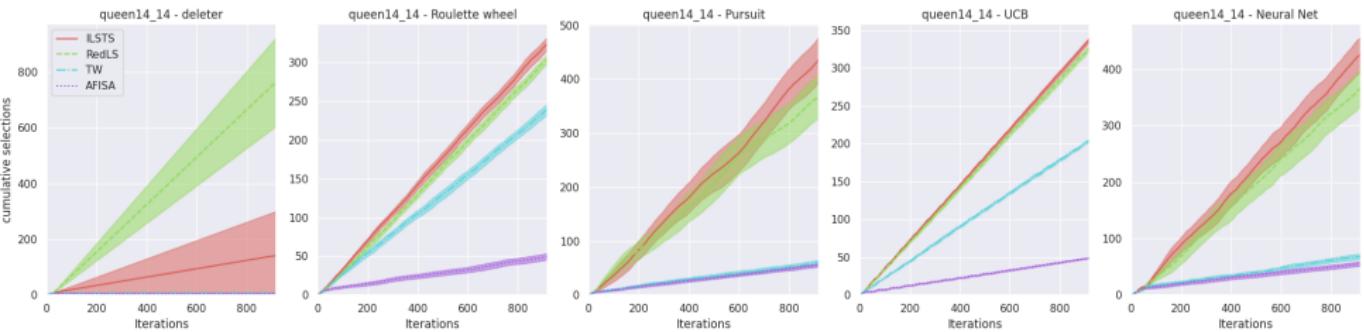
MCTS + HH (Hyperheuristics, with selection criteria)

- MCTS+HH more often better than LS and MCTS+LS
- LS alone better on about 12 instances (/188)
- ILS-TS keeps the advantage on the number of best scores but the difference is smaller
- No very large differences between the criteria
- Deleter, Pursuit and NN a little more often better than Random

MCTS + Hyperheuristics - Selection

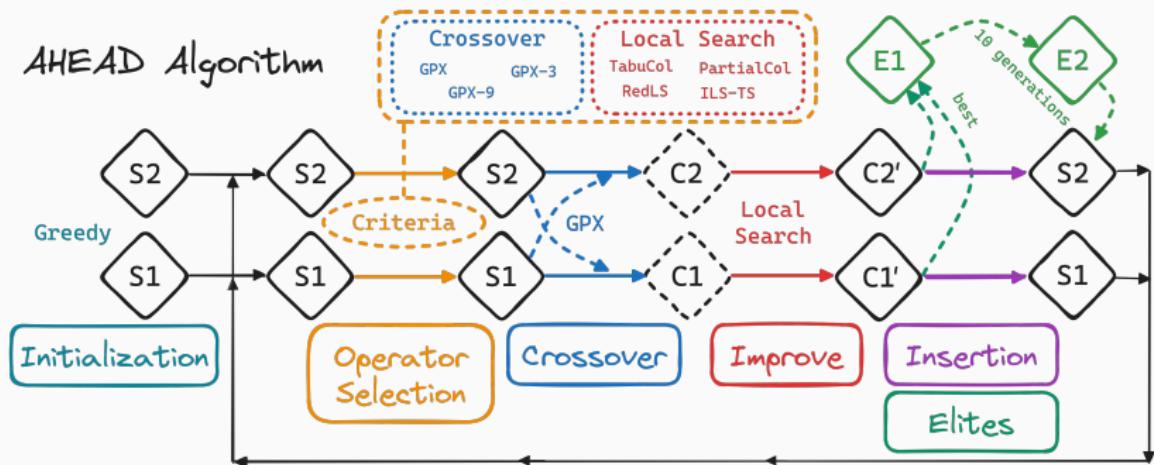


MCTS + Hyperheuristics - Selection 2/2



AHEAD

AHEAD - Adaptive Hybrid Evolutionary Algorithm in Duet



Why AHEAD?

- Attempt to **improve HEAD**
 - **Adapt** to the instance

How?

- Hyperheuristics
 - Multiple crossovers (\pm conservative)
 - Multiple local searches
 - **GCP**: TabuCol, PartialCol
 - **WVCP**: RedLS, ILS-TS

AHEAD - Conclusion

GCP - Conclusion

- HEAD + LS improve LS results
- **AHEAD better than HEAD** but HEAD+TabuCol often very good
- **New best score** on C2000.9 (Success: 404 Found)

WVCP - Conclusion

- HEAD + LS improve RedLS but not ILS-TS
- **Better results** with AHEAD
- RedLS stay better on 9 instances and ILS-TS on less than 6 (/48)
- **New best scores** on le450_15a/b (211/215) and queen14_14 (214)

To summarize

- **Conservative crossover** preferred
- **Local Search choice more important**
- Deleter and Pursuit get the best results

WVCP - Comparison between methods

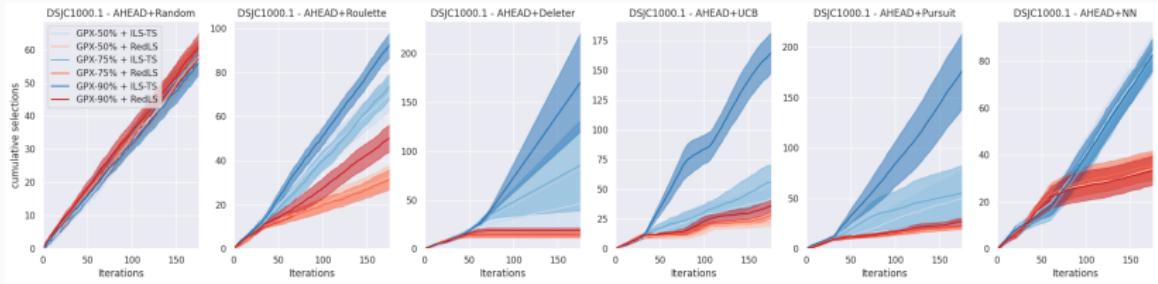
1 point/instance if the average is significantly better for the method on the line compared to the one in the column (Wilcoxon signed-rank, p-value < 0.001)

/48	MCTS+UCB	RedLS	ILS-TS	HEAD+RedLS	HEAD+ILS-TS	Random	Roulette	Deleter	UCB	Pursuit	NN	# BKS	# Best	# Best Avg
MCTS+UCB	-	25	15	3	20	0	0	0	0	0	0	20	19	15
RedLS	11	-	10	9	14	9	9	9	9	9	9	15	24	11
ILS-TS	10	27	-	8	19	3	6	5	4	1	3	23	25	21
HEAD+RedLS	16	26	15	-	25	1	1	0	0	1	0	19	19	11
HEAD+ILS-TS	5	20	6	5	-	0	0	0	0	0	0	18	19	13
Random	19	27	20	10	25	-	0	0	0	0	0	21	22	19
Roulette	17	26	20	9	26	0	-	0	0	0	0	22	22	17
Deleter	20	26	19	9	26	3	0	-	0	0	0	24	28	19
UCB	20	26	20	9	26	1	1	0	-	0	0	23	23	19
Pursuit	19	26	23	11	26	1	0	0	0	-	0	24	26	22
NN	20	27	21	10	27	0	1	0	0	0	-	21	23	19

WVCP - Results

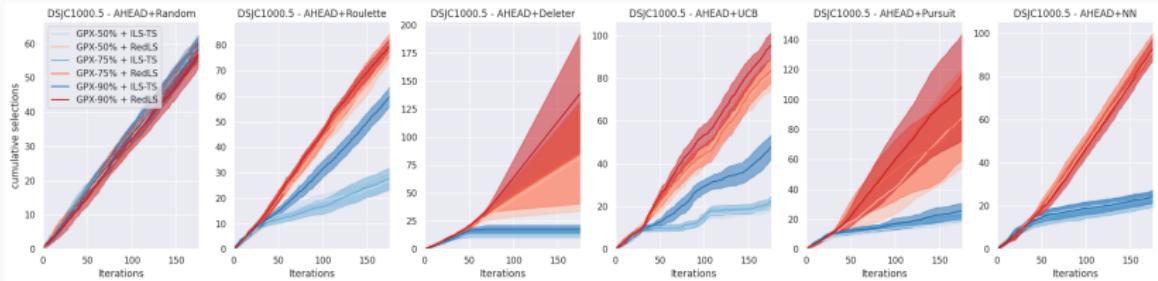
instance	BKS	RedLS			ILS-TS			HEAD+RedLS			AHEAD+Random			AHEAD+Deleter		
		best	mean	time	best	mean	time	best	mean	time	best	mean	time	best	mean	time
C2000.5	2144	2131	2155.7	18367	2244	2264.4	6423	2244	2257.9	7453	2220	2236.8	12962	2218	2236.3	1782
C2000.9	5477	5439	5455.1	23137	5847	5910.1	23014	5732	5748.2	12980	5732	5783.9	12491	5717	5758.8	12327
DSJC1000.1	300	303	306.9	5839	305	306.2	5819	304	305.6	7380	302	303.8	9348	300	302.2	12874
DSJC1000.5	1185	1190	1206.9	12204	1241	1267.7	21935	1225	1229.7	7011	1222	1228.2	5371	1224	1230.5	1476
DSJC1000.9	2836	2828	2841.8	22796	3004	3035.9	25345	2909	2926.5	820	2911	2928.7	12633	2907	2926.8	2379
DSJC500.1	184	187	194	702	185	187.3	7107	186	186.9	6594	185	186.5	10290	184	185.9	8022
DSJC500.5	685	707	712.5	27147	711	721.2	9150	709	712.6	2534	706	711.5	12516	709	713.5	5838
DSJC500.9	1662	1667	1671	9925	1709	1725.3	24351	1680	1683.5	4053	1678	1684.2	12644	1676	1682.8	8149
DSJC250.1	127	129	131.4	56	127	127.1	11901	127	4516	127	3729	127	127.2	3235		
DSJC250.5	392	399	400.8	2602	392	393.9	10722	395	396.2	8349	393	395.2	9592	392	396.6	6028
DSJC250.9	934*	934	935	9679	934	935.1	14740	934	935.1	6741	934	934.2	8097	934	935	5011
flat1000_50_0	924	1152	1165.7	6259	1213	1230.5	570	1181	1187.7	7544	1179	1186.3	4428	1180	1186.8	2952
flat1000_60_0	1162	1196	1204.8	1877	1247	1263.8	25765	1216	1227.2	10824	1213	1223.7	11726	1217	1224.5	9840
flat1000_76_0	1165	1163	1183.2	28084	1228	1242.2	16513	1192	1204	2214	1187	1203	10742	1196	1204	8938
latin_square_10	1480	1505	1515.3	14189	1555	1575	18924	1523	1532.5	11286	1510	1526.2	13987	1517	1527.8	8732
le450_15a	212	213	215.4	54	211	213.6	11684	212	212.8	6777	212	212.8	8819	211	212.4	10557
le450_15b	216	218	219.9	41	217	217.1	10346	216	217	3204	216	217.1	2736	215	216.5	11124
le450_15c	275	282	285.4	82	279	281.7	16288	277	279.4	8360	277	278.8	7220	278	279.4	4788
le450_15d	272	277	280.6	325	275	277.6	8456	274	276.1	6004	274	275.6	8759	273	275.2	13299
le450_25a	306	306	306.6	2881	306	306	142	306	161	306	169	306	131			
le450_25b	307*	307	307.6	95	307	307	23	307	53	307	28	307	19			
le450_25c	342	348	352.8	583	348	349.1	16413	347	348.1	180	346	347.8	5652	346	348	588
le450_25d	330	335	339.4	232	337	338.7	14212	333	334.4	5904	333	334.2	6282	333	334.2	9648
queen14_14	215	218	223.8	568	215	216.4	9862	216	216.6	7956	215	216.2	6384	214	215.3	8624
wap01a	545	557	577	995	547	550.1	20531	552	559.1	8178	549	553.6	14094	549	552.8	8874
wap02a	538	554	572.1	16183	536	541	21912	550	557.1	13884	541	546.1	7654	541	545.5	12994
wap03a	562	569	575.5	17878	572	575.5	22637	577	579.7	6992	573	576.3	8096	573	575.9	2944
wap04a	563	567	578.9	13939	567	570.5	7346	573	575.6	3152	570	573.2	1970	569	572.5	13790
wap05a	541	542	543.8	7719	542	542.2	11809	542	542.9	4471	542	543	12056	542	543.2	2772
wap06a	516	519	526.1	1575	516	519.5	6264	519	520.7	12180	518	521	9100	520	521.2	5978
wap07a	555	554	573	8460	565	569.2	16299	557	559.4	3360	558	559.8	12040	557	559.2	12460
wap08a	529	536	543.7	19557	543	546.9	19271	539	540.8	7452	539	541.2	1800	538	540.1	10608
#BKS		15/48			23/48			19/48			21/48			24/48		
#Best		24/48			25/48			19/48			22/48			28/48		
# Best Avg		11/48			21/48			11/48			19/48			19/48		

WVCP - Selections



No large differences in selections between the different crossovers.

Choice of the local search is more important.



GCP - Comparison between methods

1 point/instance if the average is significantly better for the method on the line compared to the one in the column (Wilcoxon signed-rank, p-value < 0.001)

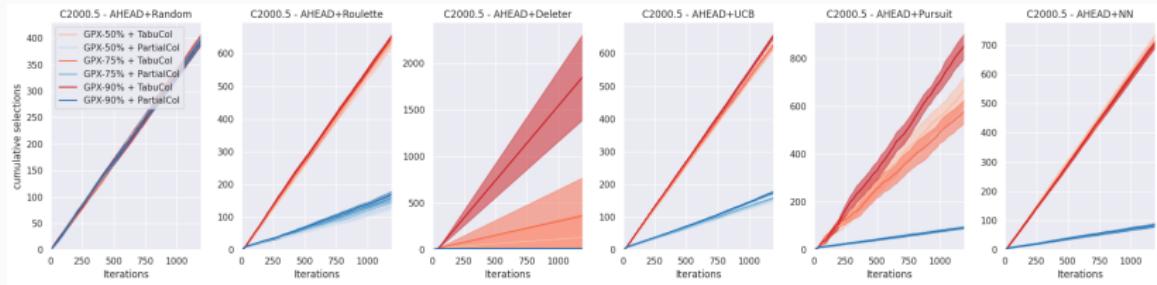
/31	PartialCol	TabuCol	HEAD+PC	HEAD+TC	Random	Roulette	Deleter	UCB	Pursuit	NN	# BKS	# Best	# Best Avg
	PartialCol	TabuCol	HEAD+PC	HEAD+TC	Random	Roulette	Deleter	UCB	Pursuit	NN			
PartialCol	-	2	3	2	1	2	1	2	2	2	5	8	11
TabuCol	14	-	11	2	2	1	0	2	0	1	8	14	7
HEAD+PC	8	6	-	1	0	0	1	0	0	0	6	10	7
HEAD+TC	18	12	20	-	4	2	1	2	2	2	7	17	15
Random	17	11	19	1	-	0	1	1	0	0	9	17	9
Roulette	17	11	19	1	0	-	0	0	0	0	11	19	12
Deleter	19	15	20	5	8	3	-	5	1	1	13	24	20
UCB	19	11	20	1	1	0	0	-	0	0	10	18	10
Pursuit	19	13	20	3	5	2	0	1	-	0	11	20	14
NN	19	12	20	2	4	0	0	0	0	-	12	23	16

PC = PartialCol – TC = TabuCol

GCP - Results

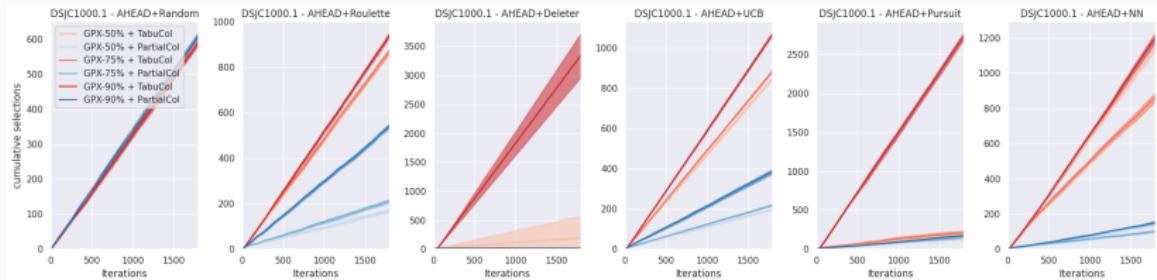
instance	BKS	PartialCol			TabuCol			HEAD+TC			AHEAD+Random			AHEAD+Deleter		
		best	mean	time	best	mean	time	best	mean	time	best	mean	time	best	mean	time
C2000.5	145	164	165.2	5313	162	162.8	4628	148	149.2	3330	150	150.7	3101	149	150.7	3152
C2000.9	408	420	420.8	5171	411	412.5	4786	<u>405</u>	406.4	2328	<u>405</u>	407.7	2956	404	405.6	2988
C4000.5	259	304	305.6	6690	303	304.2	5567	278	279.6	3580	280	281.6	3651	279	280.8	3404
DSJC500.1	12	12		128	12		75	12		86	12		80	12		56
DSJC500.5	47	50	50.1	2227	49		460	48		819	48		1258	48		850
DSJC500.9	126	128		975	126	126.3	2988	126		1027	126	126.1	1379	126		632
DSJC1000.1	20	21		1	21		0	21		0	21		1	20	20.9	2391
DSJC1000.5	82	90	90.5	3516	88		1760	83	83.3	2290	83	83.5	2372	83	83.5	2511
DSJC1000.9	222	227	228.4	3630	224	224.9	3345	223		224	223	224.2	2734	223	223.8	1589
DSJR500.5	122*	125	126.2	1666	124	127	1155	123		124	123	124.2	2245	123	123.8	2289
flat300_28_0	28*	28		896	28	29.5	3220	30	30.8	1916	28	28.5	702	28	30.4	5
flat1000_50_0	50*	50		44	50		69	50		28	50		8	50		8
flat1000_60_0	60*	60		213	60		233	60		54	60		28	60		29
flat1000_76_0	76*	89	89.1	2845	86	87	3096	82	82.3	1905	82	82.8	2775	82	82.8	1969
latin_square_10	97	107	110.2	4875	100	100.8	4377	102	103.7	93	103	103.8	1996	99	100.7	1729
le450_25c	25*	27		69	26		0	26		0	25	25.9	1407	25	25.3	1022
le450_25d	25*	27		50	26		0	26		0	26		0	25	25.3	1537
r250.5	65*	67		134	66	67.2	462	65	66	3378	65	66	1638	66		549
r1000.1c	98	141	149.1	61	134	155.2	77	100	101.6	264	100	101.6	1674	100	101.6	1621
r1000.5	234	247	248.1	5638	244	245.6	3622	246	247.6	1479	246	247.4	2134	245	245.5	2009
wap01a	41*	42		1088	42	43	2160	42		137	42		143	41	42	1958
wap02a	40*	41	41.7	4275	40	41.1	6499	41		15	41		15	40	40.8	1634
wap03a	43	44		91	44	45.9	4342	45		261	45		87	43	44.3	2387
wap04a	41	43		61	42	43.1	4869	43		880	43		1186	43		293
wap06a	40*	41		98	40	41.3	4248	40		909	40	40.8	1549	40		246
wap07a	41	44		41	41	42.3	5046	42	42.1	1771	42	43	2526	42	42.1	494
wap08a	40*	43	43.2	2750	41	41.5	2967	42		48	42		365	41	41.9	2146
#BKS		5/31			8/31			7/31			9/31			13/31		
#Best		8/31			14/31			17/31			17/31			24/31		
#Best Avg		11/31			7/31			15/31			9/31			20/31		

GCP - Selections



No large differences in selections between the different crossovers.

Choice of the local search is more important.



DLMCOL

DLMCOL - Parameter settings

Table 5: Parameter setting in DLMCOL

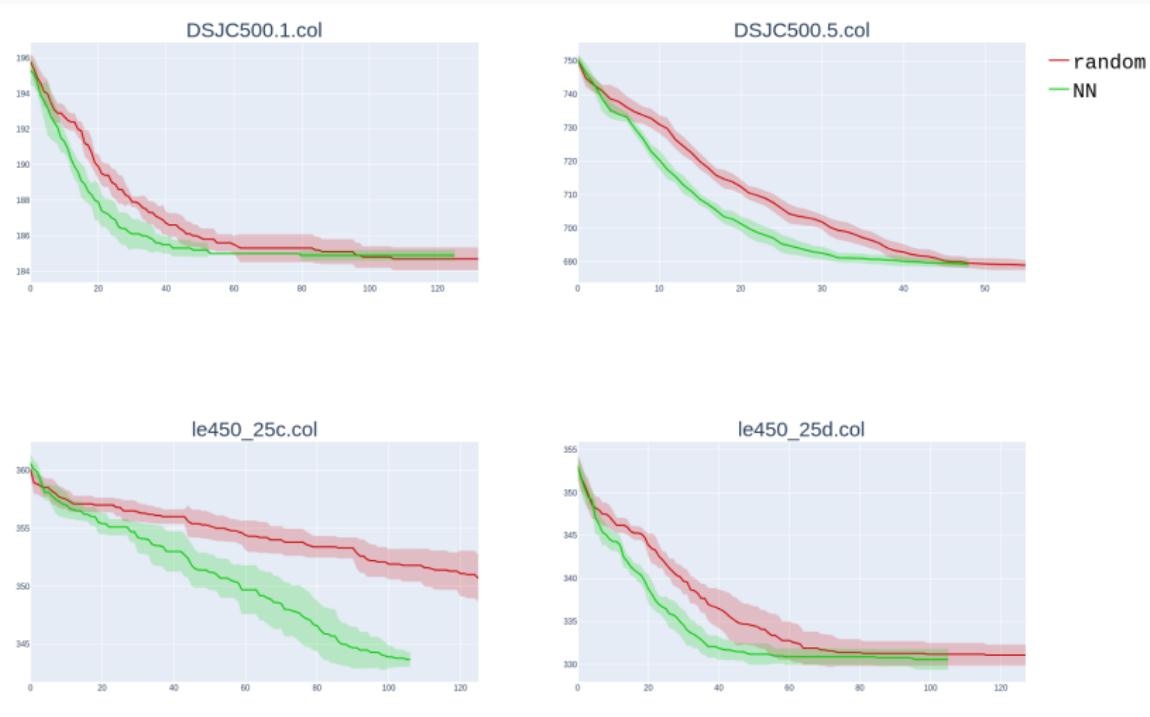
Parameter	Description	Value
p	Population size	20480 (8192)
$maxLSItrs$	Number of iterations local search	10
$nblter_{TS}$	Number of iterations tabu search	$10 \times V $
α	Tabu tenure parameter	0.2
MS	Minimum spacing between two individuals	$\frac{ V }{10}$
l_r	Learning rate of the neural network	0.001
N	Number of epochs of the training	20
K	Number of considered neighbors for crossover selection	32

Neural network architecture : 9 hidden layers of size $10|V|$, $5|V|$, $2|V|$, $2|V|$, $2|V|$, $2|V|$, $|V|$ and $|V|//2$.

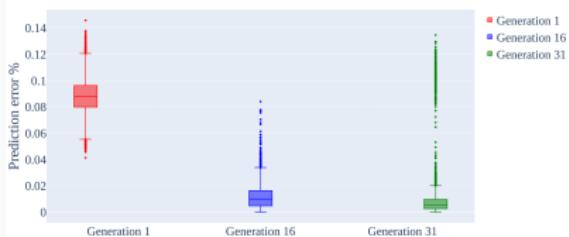
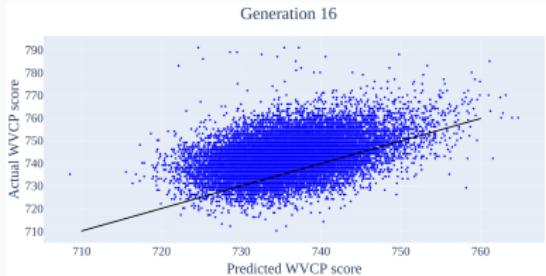
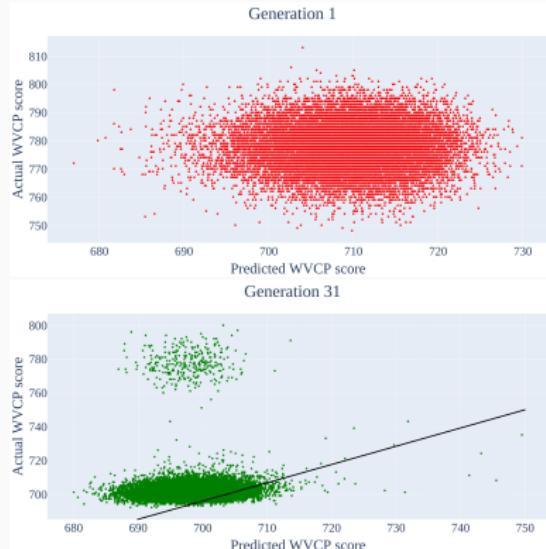
DLMCOL - New Best Scores for WVCP

Instance	V	previous score _{best}	new score _{best}	Improvement	Time (s)
DSJC500.1	500	186	184	-2	23049
DSJC500.5	500	707	685	-22	47064
DSJC500.9	500	1667	1662	-5	121518
DJSC1000.5	1000	1220	1185	-35	
flat1000_50_0	1000	1184	924	-260	82068
queen14_14	196	216	215	-1	16621
flat1000_60_0	1000	1220	1162	-58	
flat1000_76.0	1000	1200	1165	-35	
queen14_14	196	216	215	-1	16621
queen15_15	225	224	223	-1	16621
queen16_16	256	238	234	-4	14751
latin_square_10	900	1542	1480	-62	149656
le450_15c	450	277	275	-2	43854
le450_15d	450	274	272	-2	22917
le450_25c	450	349	342	-7	57924
le450_25d	450	339	330	-9	45128

DLMCOL - Learning vs Random crossover choice



DLMCOL - Predicted vs Actual Score (DSJC500.5.col)



DLMCOL - Predicted vs Actual Score (wap05a)

