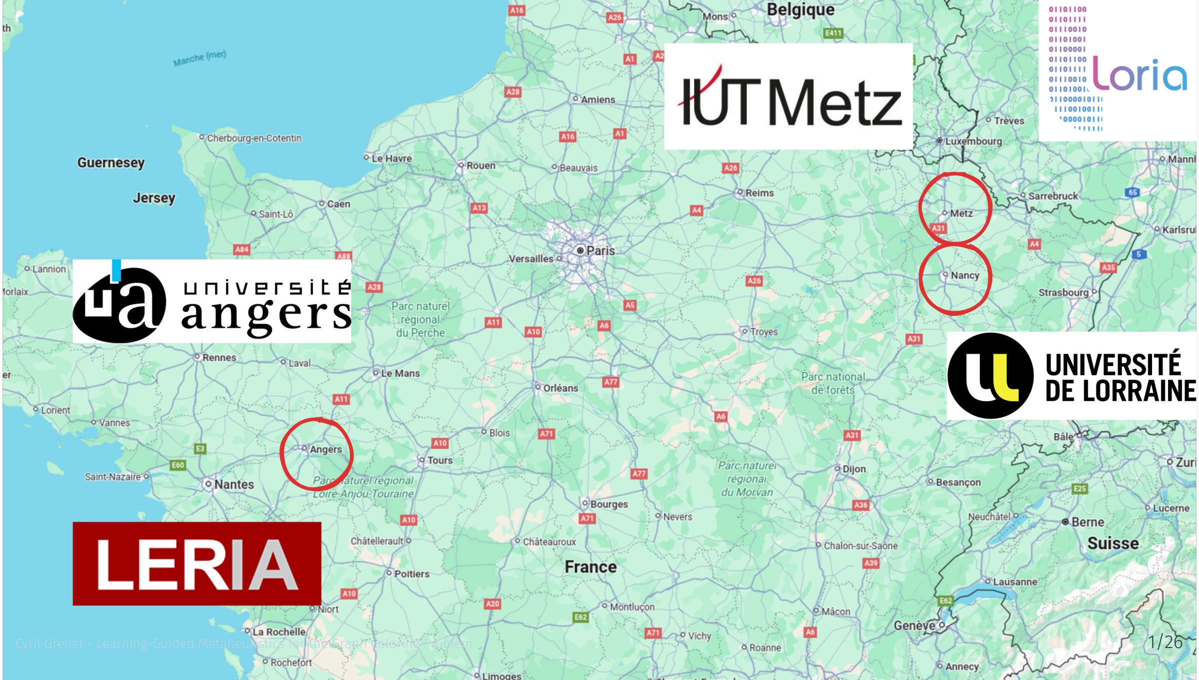
Learning-Guided Metaheuristics for the Graph Coloring Problem

Seminar - LORIA - Department 3

Cyril Grelier


IUT Metz
Loria
université angers
UNIVERSITÉ DE LORRAINE
LERIA
Belgique
Manche (mer)
Guernesey
Jersey
Mons
Amiens
Cherbourg-en-Cotentin
Le Havre
Rouen
Beauvais
Trèves
Luxembourg
Reims
Sarrebruck
Metz
Nancy
Strasbourg
Karlsruhe
Caen
Saint-Lô
Paris
Versailles
Lannion
Morlaix
Parc naturel régional du Perche
Troyes
Rennes
Laval
Le Mans
Orléans
Parc national de forêts
Lorient
Vannes
Blois
Angers
Tours
Bâle
Saint-Nazaire
Nantes
Parc naturel régional Loire-Anjou-Touraine
Parc naturel régional du Morvan
Dijon
Besançon
Bourges
Nevers
Neuchâtel
Berne
Lucerne
Suisse
Châtellerault
Châteauroux
Chalon-sur-Saône
France
Poitiers
Lausanne
Niort
Montluçon
Mâcon
Genève
La Rochelle
Vichy
Roanne
Rochefort
Limoges
Annecy

Education and Career

2013	Higher National Diploma - Plastics Industry	La Baronnerie - Angers
2014	Year of scientific upgrading	Université Catholique de l'Ouest - Angers
2016	University Diploma of Technology - Electrical Engineering & Industrial Computing	IUT Angers
2016	ERASMUS Sweden - Computer Science	Halmstad University - Sweden
2018	Bachelor's Degree - Computer Science	Université d'Angers
2020	Master's Degree - Computer Science - Artificial Intelligence	Université d'Angers
2023	PhD - Computer Science *Learning-Guided Metaheuristics for the Graph Coloring Problem* with Olivier Goudet and Jin-Kao Hao	Université d'Angers - LERIA
2024	Research Engineer work on *QuChemPedIA* project with Benoit Da Mota and Thomas Cauchy	LERIA
2025	PostDoc *Optimisation and learning algorithms for vehicle routing problems* with Ammar Oulamara	LORIA
2025	Associate Professor	LORIA - Université de Lorraine - IUT de Metz

PhD Thesis

Olivier Goudet - Jin-Kao Hao

10/2020–09/2023 – Defense 12/12/2023

Learning-Guided Metaheuristics for Graph Coloring

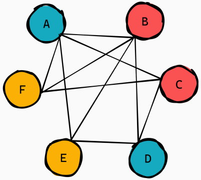
PhD Thesis

Olivier Goudet - Jin-Kao Hao

10/2020–09/2023 – Defense 12/12/2023

Learning-Guided Metaheuristics for Graph Coloring

Combinatorial optimization problem – assigning colors to the vertices of a graph
(Assignment – Scheduling)



PhD Thesis

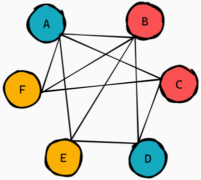
Olivier Goudet - Jin-Kao Hao

10/2020–09/2023 – Defense 12/12/2023

Learning-Guided Metaheuristics for Graph Coloring

Combinatorial optimization problem – assigning colors to the vertices of a graph
(Assignment – Scheduling)

Optimization algorithms for hard problems
(Operations Research – Combinatorial Optimization)



PhD Thesis

Olivier Goudet - Jin-Kao Hao

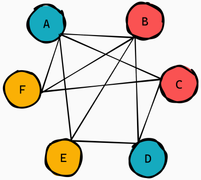
10/2020–09/2023 – Defense 12/12/2023

Learning-Guided Metaheuristics for Graph Coloring

Combinatorial optimization problem – assigning colors to the vertices of a graph
(Assignment – Scheduling)

Optimization algorithms for hard problems
(Operations Research – Combinatorial Optimization)

Algorithms that learn from data
(Machine Learning)



PhD Thesis

Olivier Goudet - Jin-Kao Hao

10/2020–09/2023 – Defense 12/12/2023

How to link Metaheuristics and Learning?

Olivier Goudet - Jin-Kao Hao

10/2020–09/2023 – Defense 12/12/2023

How to link Metaheuristics and Learning?

- ✓ Learning guides the search
  - ❖ Monte Carlo Tree Search (MCTS)

PhD Thesis

Olivier Goudet - Jin-Kao Hao

10/2020–09/2023 – Defense 12/12/2023

How to link Metaheuristics and Learning?

- ✓ Learning guides the search
  - ❖ Monte Carlo Tree Search (MCTS)

- ❖ Greedy Algorithms ➟ Solution initialization
- ❖ Local Search (LS) ➟ Iterative improvements

Olivier Goudet - Jin-Kao Hao

10/2020–09/2023 – Defense 12/12/2023

How to link Metaheuristics and Learning?

- ✓ Learning guides the search
  - ❖ Monte Carlo Tree Search (MCTS)
- ✓ Learning selects operators
  - ❖ Hyper-heuristics

- ❖ Greedy Algorithms ➟ Solution initialization
- ❖ Local Search (LS) ➟ Iterative improvements

PhD Thesis

Olivier Goudet - Jin-Kao Hao

10/2020–09/2023 – Defense 12/12/2023

How to link Metaheuristics and Learning?

- ✓ Learning guides the search
  - ❖ Monte Carlo Tree Search (MCTS)
- ✓ Learning selects operators
  - ❖ Hyper-heuristics

- ❖ Greedy Algorithms ➟ Solution initialization
- ❖ Local Search (LS) ➟ Iterative improvements
- ❖ Memetic Algorithms ➟ Population management

PhD Thesis

Olivier Goudet - Jin-Kao Hao - David Lesaint

10/2020–09/2023 – Defense 12/12/2023

How to link Metaheuristics and Learning?

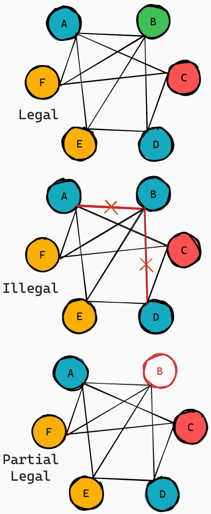
- ✓ Learning guides the search
  - ❖ Monte Carlo Tree Search (MCTS)
- ✓ Learning selects operators
  - ❖ Hyper-heuristics

- ❖ Greedy Algorithms ➟ Solution initialization
- ❖ Local Search (LS) ➟ Iterative improvements
- ❖ Memetic Algorithms ➟ Population management

- ✚ Vertex reduction ➟ Search space reduction
- ✚ Bounds on the number of colors / score ➟ Filtering / pruning
- ✚ Constraint Programming (CP) ➟ Propagation and filtering

Graph Coloring

GCP & WVCP

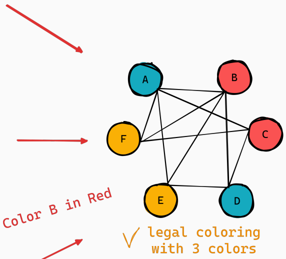


GCP – Graph Coloring Problem

Find the chromatic number χ_G of the graph

k-coloring

Find a legal coloring with k colors

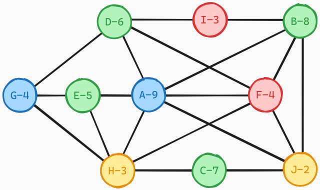


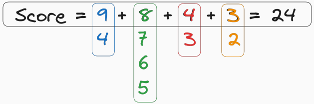
Applications GCP:
- Register Allocation
- Frequency Assignment
- Scheduling
- Sudoku
- ...

WVCP – Weighted Vertex Coloring Problem

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

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

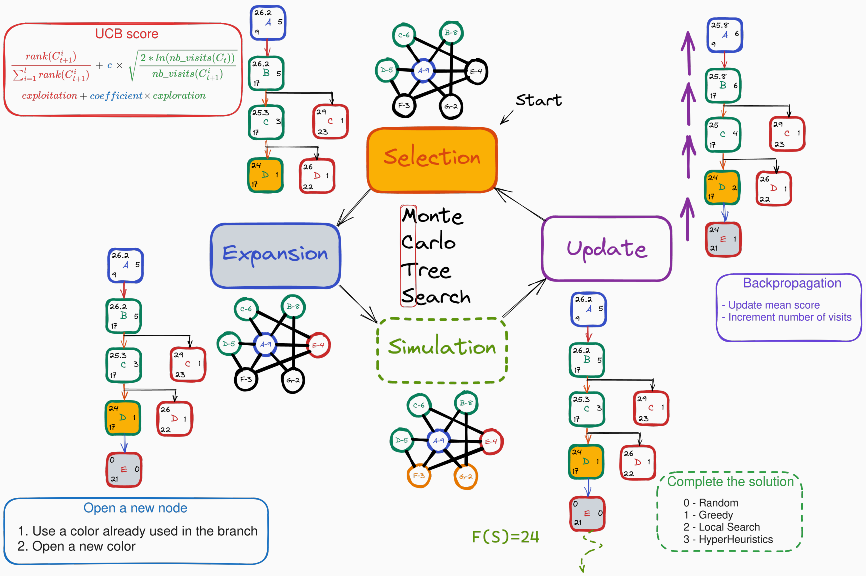




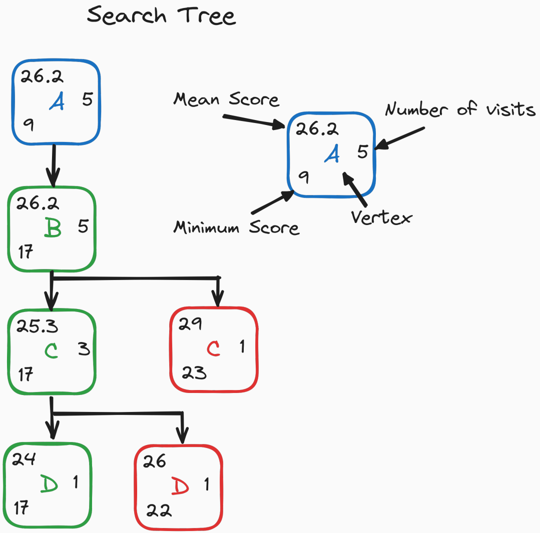
Applications WVCP:
- Traffic management in satellite communications
- Matrix decomposition problem
- Scheduling batch job in parallel

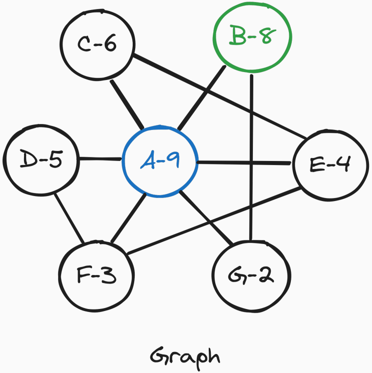
Monte Carlo Tree Search - MCTS

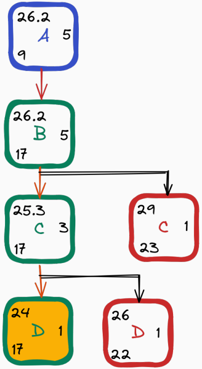
MCTS - Monte Carlo Tree Search

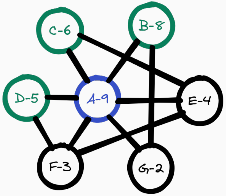


MCTS - A Search Tree and a Graph







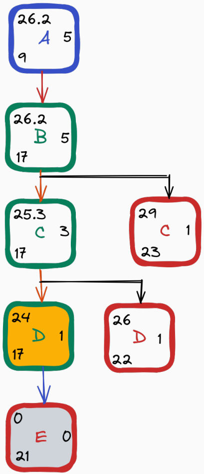


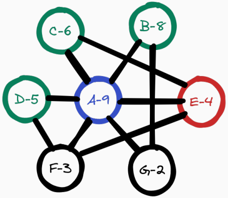
Use UCB score (Jooken *et al.* [2023])

Exploitation + coefficient * Exploration

$$\frac{rank(C^i_{t+1})}{\sum_{i=1}^{l} rank(C^i_{t+1})} + c \times \sqrt{\frac{2 * ln(nb\_visits(C_t))}{nb\_visits(C^i_{t+1})}}$$

MCTS - Phase 2 : Expansion





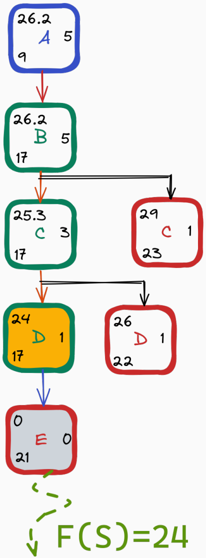
Expansion

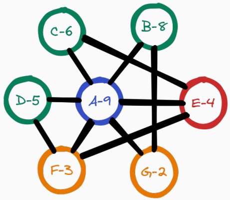
Open a new node:

1 - Use a color already in use in the branch

2 - Open a new color (limit with bounds and score)

MCTS - Phase 3 : Simulation





Simulation

Complete the coloring

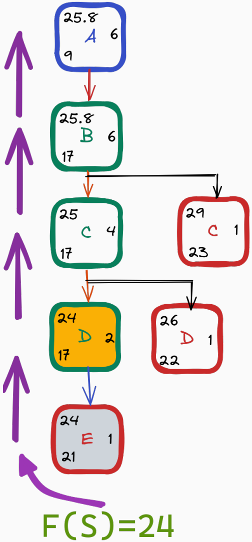
0 - Purely random choice

1 - Greedy Algorithm

2 - Local Search (Grelier *et al.* [2022])

3 - Hyperheuristics (Grelier *et al.* [2023])

MCTS - Phase 4 : Update

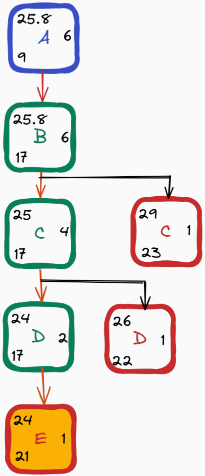


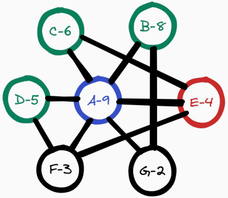
Backpropagation of the score to update :

- Mean score

- Number of visits

MCTS - Phase 1 : Selection...

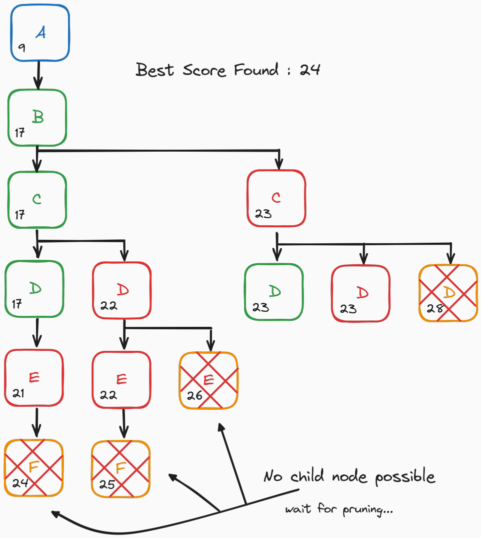


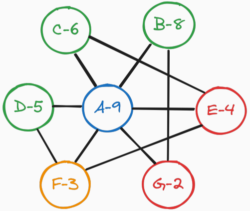


Selection

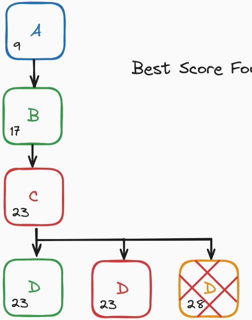
New Iteration...

MCTS - Pruning 1

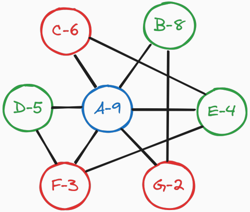




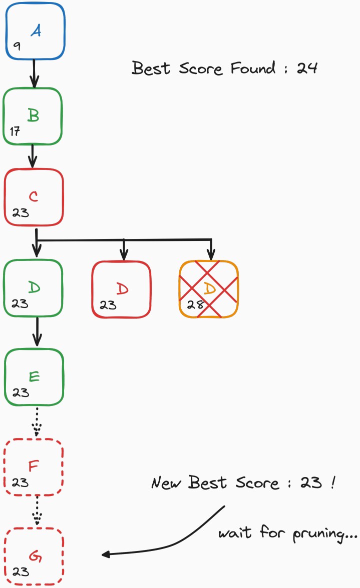
MCTS - Pruning 2

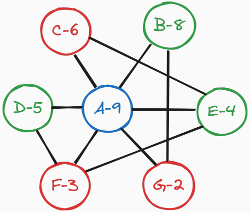


Best Score Found : 24

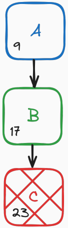


MCTS - Pruning 3

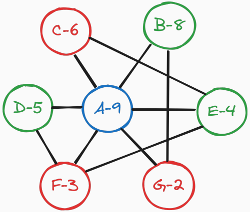




MCTS - Pruning 4



Best Score Found : 23



Tree fully explored -> 23 is the optimal score!

MCTS + Greedy - Color the remaining vertices

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 100/244 GCP instances and 50/188 WVCP instances
- MCTS is very good on small instances and some medium ones
- Less efficient on larger instances

MCTS + Local Search - Improve the solution after the Greedy Algorithm

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 the results of several local searches
- Different results depending on the instance and the local search

MCTS + Hyperheuristics - Adapt the search to the instance

Why?

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

How does it work?

1. Complete the solution with a greedy algorithm
2. Ask a **criteria** which local search to use
3. Perform the local search for limited time
4. **Update** the criteria with the obtained score (**reward**)

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]

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)

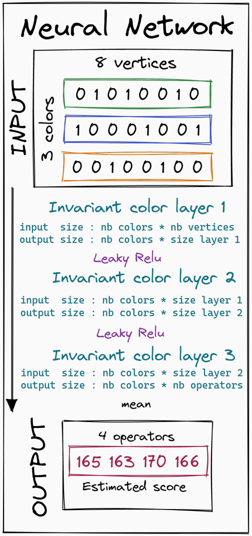
$$\begin{cases} proba[b] = proba[b] + \beta(p\_max - proba[b]) \\ proba[o] = proba[o] + \beta(p\_min - proba[o]) \end{cases}$$

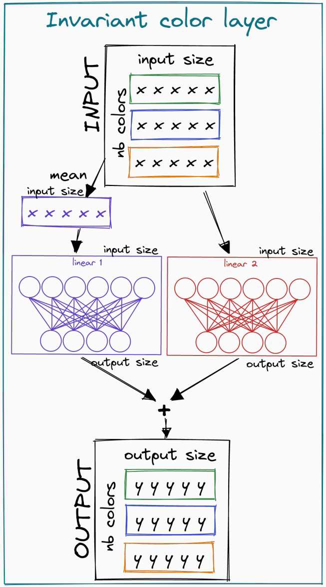
- **UCB** : Focusing on the best while encouraging exploration

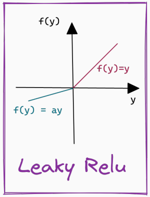
$$score[o] = r[o] + c * \sqrt{2 * \frac{\log(\sum visits)}{visits[o]}}$$

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

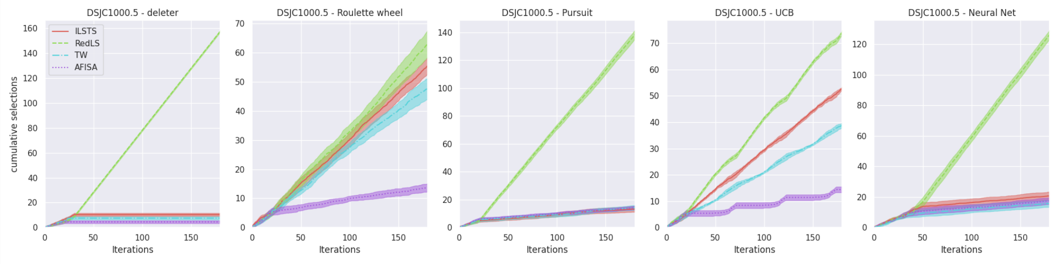
Neural Network - NN - Deep sets







MCTS + Hyperheuristics - Results



Selection of Local Search Operators

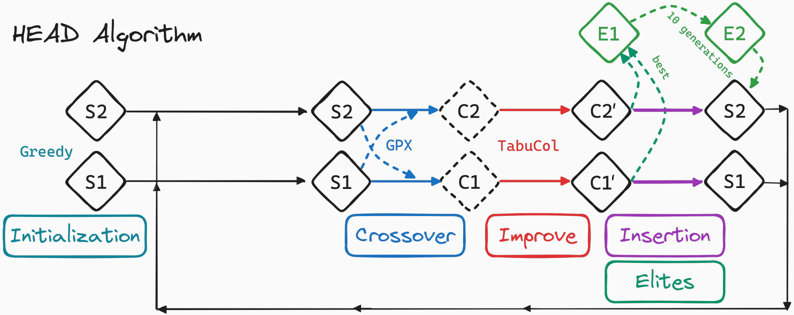
- Generally 1 or 2 good operators per instance
- Importance of having complementary operators
- No change in the choice during search

Results

- 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



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

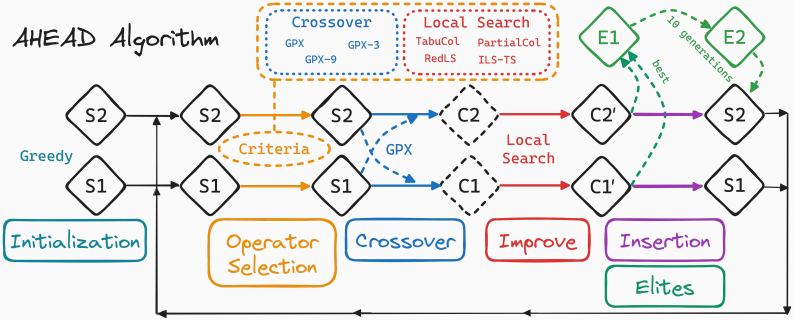
Why HEAD?

- **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



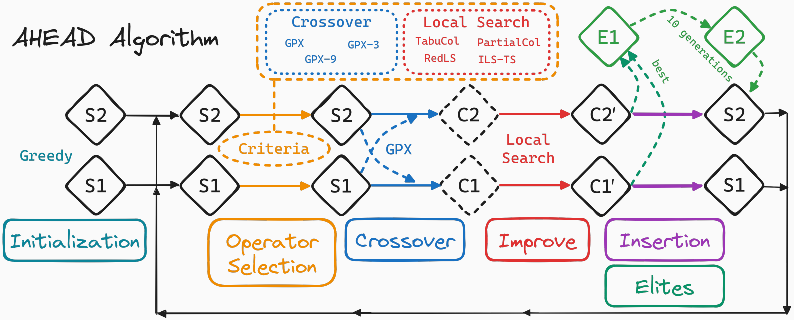
Why AHEAD?

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

Which Operators?

- 3 GPX crossovers (± conservative)
- 2 local searches/problem
  - GCP: TabuCol, PartialCol
  - WVCP: RedLS, ILS-TS

AHEAD - Adaptive Hybrid Evolutionary Algorithm in Duet



Results on GCP

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

Results on WVCP

- Better results than LS/MCTS/HEAD
- 3 new best scores

Hyperheuristics?

- Conservative crossover preferred
- Local Search choice more important
- Best are Deleter and Pursuit

Conclusion & Perspectives

Conclusion & ⋆ Perspectives

- Theoretical contributions
  - Vertex reduction - IJCAI 2023
    - ⋆ New rules of reduction
    - ⋆ Exact Model for reduction
  - Bounds on colors and score - IJCAI 2023
- Implemented methods
  - Constraint programming models - IJCAI 2023
  - MCTS + Greedy - EvoCOP 2022
    - ⋆ DSatur/Branch&Bound 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
- Other methods
  - ⋆ Branch&Bound + LS/MCTS
    →Nicolas Dupin - Univ Angers/LERIA
  - ⋆ TabuEdges - ROADEF 2025
  - ⋆ EvoWeight (Guided Memetic Algorithm)
  - ⋆ LNS + Exact + Learning
  - ⋆ Hybridations exact + metaheuristics

Thank you for your attention!

Questions?

Publications, source code, results :

`https://cyril-grelier.github.io/`

References

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, Kaile Su, and Abdul Sattar. Local search with edge weighting and configuration checking heuristics for minimum vertex cover. *Artificial Intelligence*, 175(9-10):1672–1696, 2011.

Shaowei Cai and Jinkun Lin. Fast solving maximum weight clique problem in massive graphs. In *IJCAI*, pages 568–574, 2016.

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.

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.

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.

Bibliography iii

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.

Alain Hertz and D de Werra. Using tabu search techniques for graph coloring. *Computing*, 39(4):345–351, 1987.

Marijn JH Heule, Anthony Karahalios, and Willem-Jan van Hoeve. From cliques to colorings and back again. In *28th International Conference on Principles and Practice of Constraint Programming (CP 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.

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.

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.

Laurent Moalic and Alexandre Gondran. Variations on memetic algorithms for graph coloring problems. *Journal of Heuristics*, 24:1–24, 2018.

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.

Graph Coloring

GCP - Introduction

Graph Coloring Problem

Objective: find a legal coloring that minimizes the number of colors

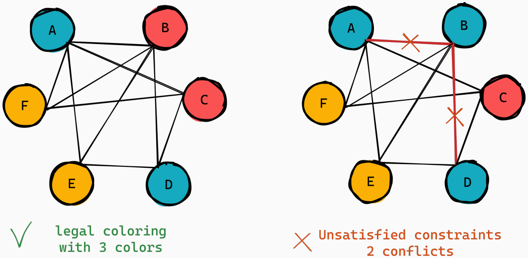
Score:

- Number of colors k (legal)
- Number of conflicts $|C|$ (illegal)
- Number of uncolored vertices $|U|$ (partially legal)

NP-Hard problem

Applications:

- Scheduling problems
- Register allocation
- Frequency assignment
- Sudoku



GCP - Graph Coloring Problem

Example of k-coloring with k=3

Search Space

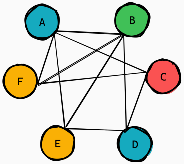
Unsatisfied constraints Score

GCP : Find the chromatic number of the graph

k-coloring : Find a legal coloring with k colors

NP-Hard Problems

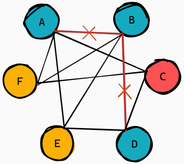
Legal



4 colors used

Score = 4

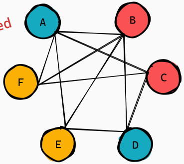
Illegal



2 conflicts

Score = 2

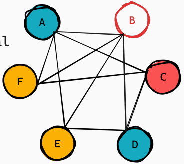
Color B in Red



√

legal coloring
with 3 colors

Partial-Legal



vertex B not colored

Score = 1

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

WVCP - Weighted Vertex Coloring Problem

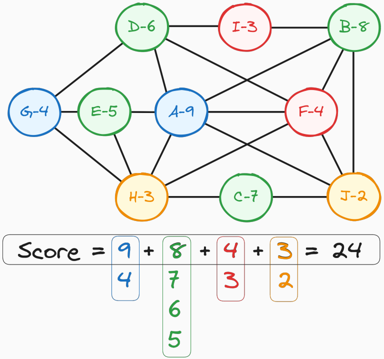
Objective: find a legal coloring that minimizes the sum of the weights of the heaviest vertices in each color

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

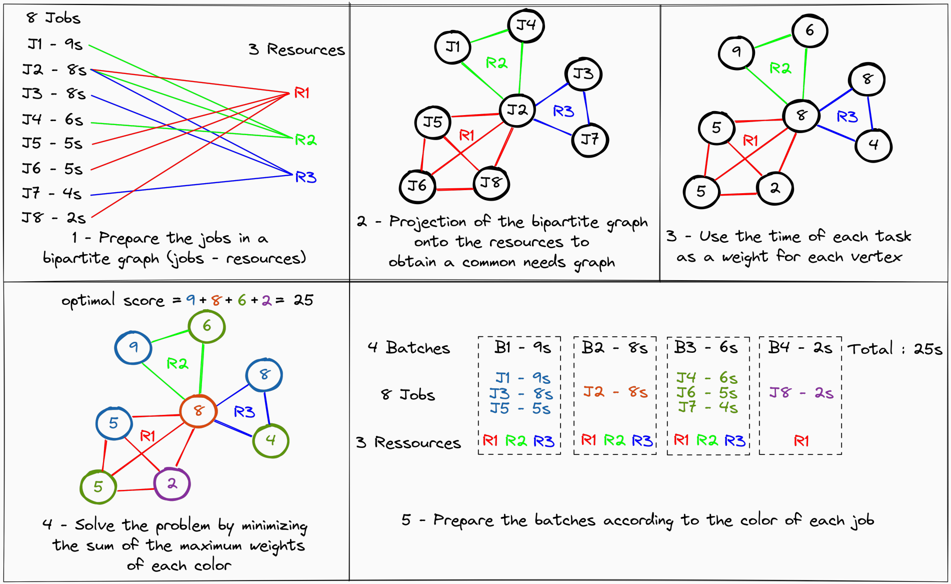
NP-Hard problem

Applications :

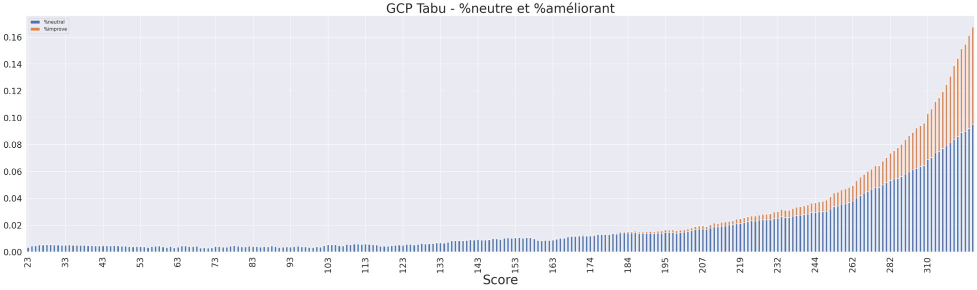
- Traffic management in satellite communications
- Matrix decomposition problem
- Scheduling batch job in parallel

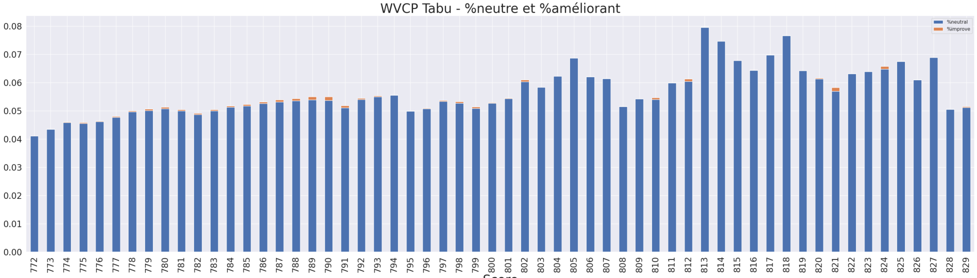


WVCP - Scheduling Parallel Batch Jobs

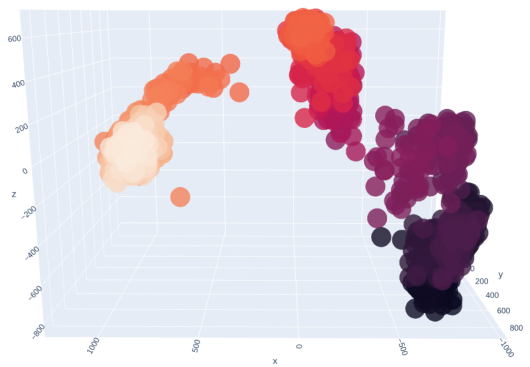


GCP/WVCP - Exploration

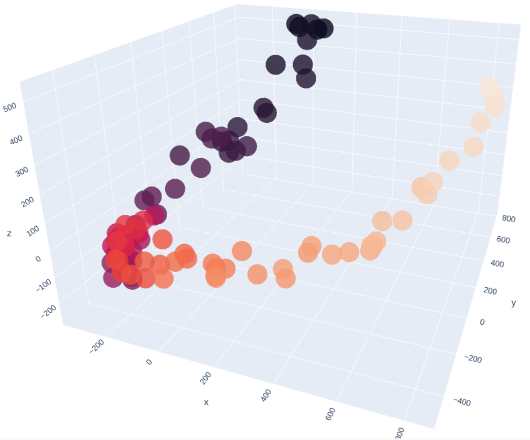




GCP/WVCP - Exploration



Search path WVCP



Search path GCP

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, no simulation

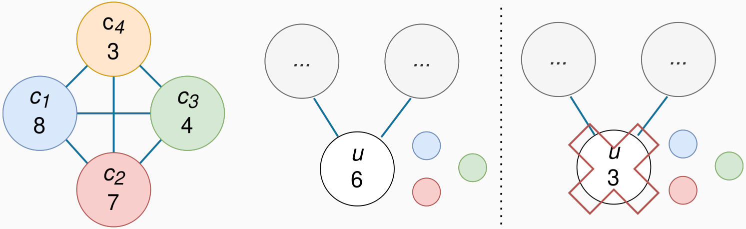
WVCP - State of the Art and Contributions

- 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

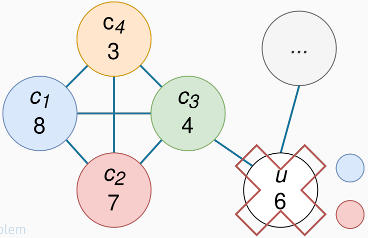
Vertex Reduction Rules and Iterative Reduction Procedure

Reduction Rules R0 and R1

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

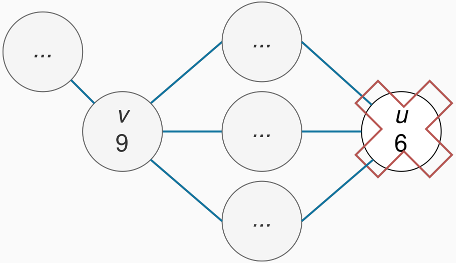


- R1, takes into account that u may have neighbors in C (WVCP):



Reduction Rule R2 and Iterative Reduction Procedure

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



- Iterative Reduction Procedure : demo
  1. Compute a clique for each vertex with FastWClq 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!

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(=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:

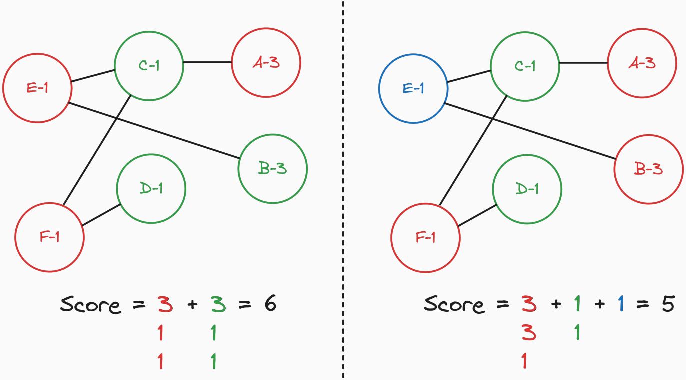
- 67 instances out of 244 of the GCP keep only one clique.
- Help for exact and approximate resolution methods
- Help for the calculation of bounds

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

Bounds WVCP - Introduction

Why ?

- No limit on the number of colors - a solution with more colors can be better
- Necessary for exact methods
- Reduce the search space



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 [1]

[1] Best 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 :

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

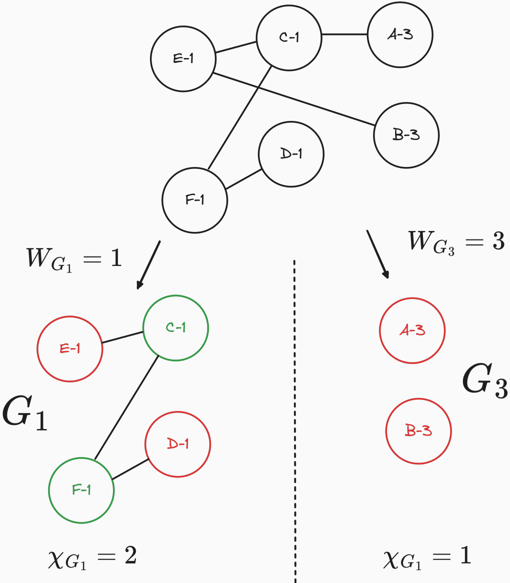
- on the score :

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

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 - 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 - Results

Instance	$\lvert V' \rvert$	density	h_W	$\Delta + 1$	bounds colors lb	bounds colors ub	bounds score lb	bounds score 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 score	primal time(s)	primal ub color score	primal ub color time(s)	primal all bounds score	primal all bounds time(s)
DSJC125.1g	23	23*	862	23*	435	23*	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

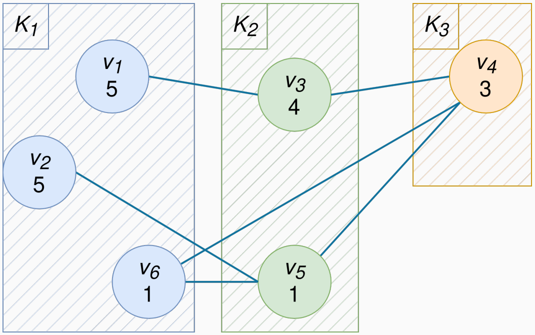
- 88/188 instances with a better upper bound on the number of colors compared to $\Delta(G) + 1$
  $\hookrightarrow$ On average 13 fewer colors
- In practice :
  - Better results with a CP model Goudet *et al.* [2023]
    $\hookrightarrow$95 proven optimal instances versus 72 without the bounds
  - Low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

Three Constraint Programming Models for WVCP

CP models - 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 $K = \{1, \ldots, \kappa\}$.
- 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.
- $\mathcal{S}_{P_\kappa}$: set of d-solutions using a number of colors smaller than κ.

CP models - Example of d-sorted solution



CP models - Primal model for P_κ

minimize x^o *s.t.*

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

$$\forall v_i \in U : x_i^U \in K \tag{P2}$$

$$\forall k \in K : x_k^K \in 2^U \tag{P3}$$

$$\forall k \in K : x_k^D \in U \tag{P4}$$

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

$$\forall k \in K : x_{|V|+k}^U = k \tag{P6}$$

$$\forall \{v_i, v_j\} \in E : x_i^U \neq x_j^U \tag{P7}$$

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

$$x^o = \sum_{k \in K} w[x_k^D] \tag{P9}$$

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

CP models - 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.

CP models - Primal model results and impact of pre-computed bounds

		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
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.

CP models - 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

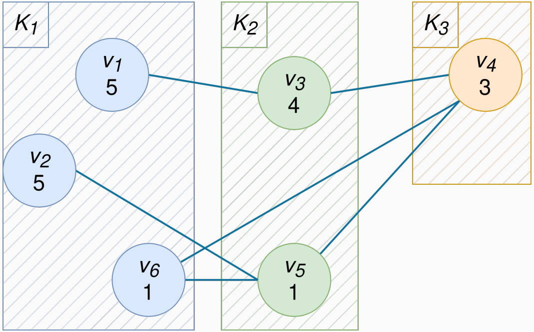
Let P_κ be a satisfiable WVCP instance. There exists $g_{P_\kappa} : \mathcal{S}_{P_\kappa} \mapsto \mathcal{S}_{P_\kappa}$ such that, for all $s \in \mathcal{S}_{P_\kappa}$, $g_{P_\kappa}(s)$ is compact, $f(g_{P_\kappa}(s)) \leq f(s)$ and $g_{P_\kappa}(g_{P_\kappa}(s)) = g_{P_\kappa}(s)$.

Corollaries

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

CP models - 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.



CP models - 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 ($n \geq 0$). 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**:

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

CP models - Implementation of MAX_LEFT_SHIFT

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

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

$$\forall i \in \{1, \ldots, n\} : z_i \in \{0, \ldots, n+1\} \tag{M1}$$

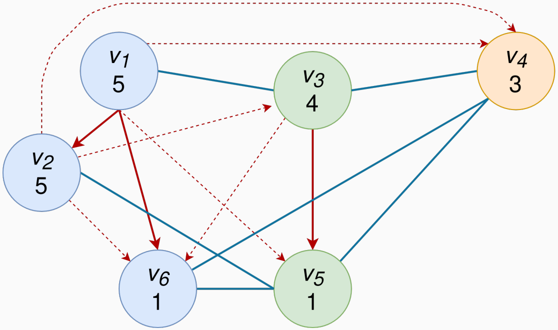
$$\forall i \in \{1, \ldots, n\} : z_i = (y > x_i) \times x_i \tag{M2}$$

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

CP models - Impact of this symmetry breaking rule on the results

		primal		primal + P11	
instance	BKS	score	time(s)	score	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 reached		101/188		102/188	
nb optim		72/188		76/188	

CP models - Dual graph from Cornaz et Jost [2008]



Primal

$$Score_P = w(v_1) + w(v_3) + w(v_4) = 12$$

Dual

$$Score_D = w(v_1) + w(v_3) + w(v_4) = 7$$

$$Score_P + Score_D = \sum_{i=1}^{n} w(v_i) = 19$$

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

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

maximize y^o s.t.

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

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

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

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

$$y^A_{ij} + y^A_{ik} \leq 1 \tag{D4}$$

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

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

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

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

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

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

CP models - Joint model

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

minimize x^o *s.t.*

$$\forall ij \in \overrightarrow{E^c} : y^A_{ij} \leq (x^U_i = x^U_j) \tag{J1}$$

$$\textsc{gcc}([x^D_k \mid k \in K],\ V,\ [z^V_i \mid v_i \in V]) \tag{J2}$$

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

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

$$\left(\bigwedge_{v_h \in N(v_i) \cap \overline{N(v_j)}} x^U_h \neq x^U_j \right) \Rightarrow x^U_i \leq x^U_j \tag{J4}$$

CP models - Results

instance	BKS	primal score	primal time(s)	primal + P11 score	primal + P11 time(s)	dual score	dual time(s)	joint + J4 score	joint + J4 time(s)
DSJC125.1g	23	<u>23*</u>	862	<u>23*</u>	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	

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

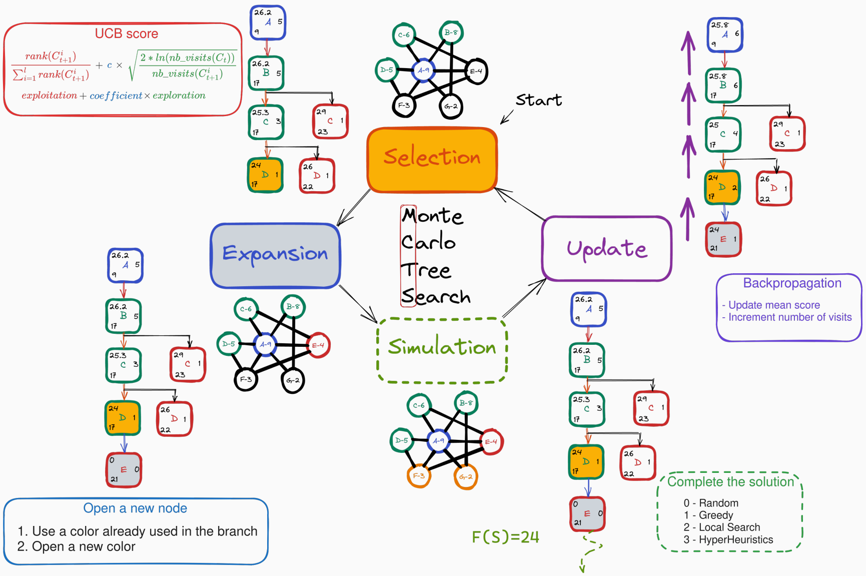
Table 2: New optimality proofs for difficult benchmark instances.

CP models - Conclusion

- Iterative reduction procedure and new upper bounds on the score and the number of colors.
  $\rightarrow$ 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 - Monte Carlo Tree Search



MCTS - Why ?

Particularities of the WVCP

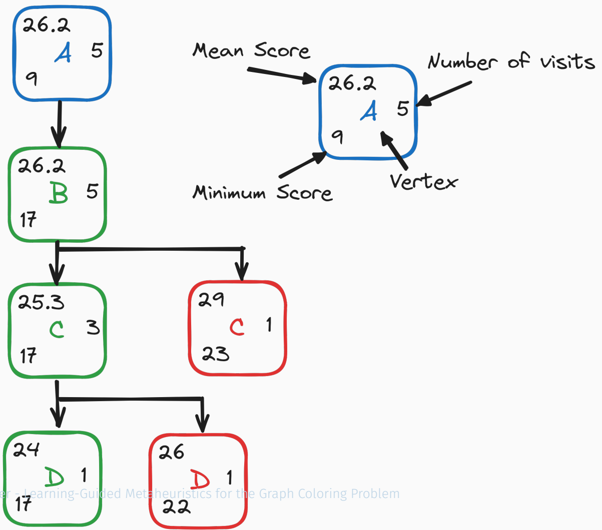
- Some vertices are more important (weight, degree)
- Only the heaviest vertex of a color has an impact on the score

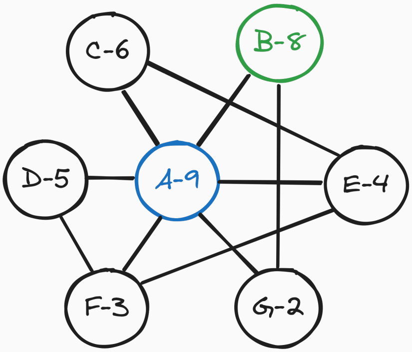
Why MCTS ?

- Find a good color for the heaviest vertices
- Continuously explore new areas of the search space
- Cut symmetries and prune well to prove optimality
- Hybridation with other methods (Greedy, Local Searches,...)

MCTS - A Search Tree and a Graph

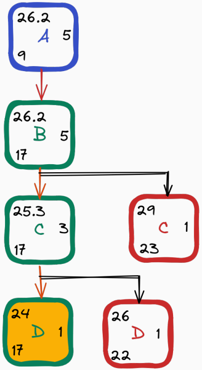
Search Tree

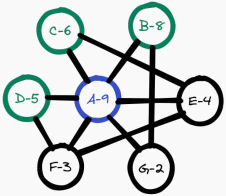




Graph

MCTS - Phase 1 : Selection



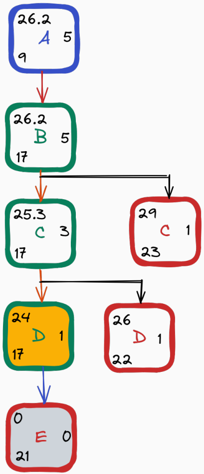


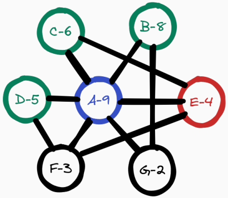
Use UCB score (Jooken *et al.* [2023])

Exploitation + coefficient * Exploration

$$\frac{rank(C^i_{t+1})}{\sum_{i=1}^{l} rank(C^i_{t+1})} + c \times \sqrt{\frac{2 * ln(nb\_visits(C_t))}{nb\_visits(C^i_{t+1})}}$$

MCTS - Phase 2 : Expansion





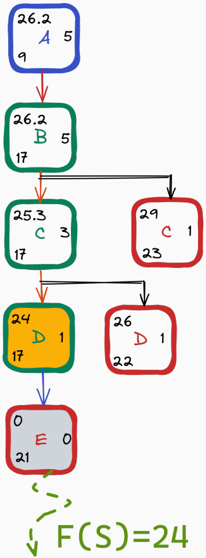
Expansion

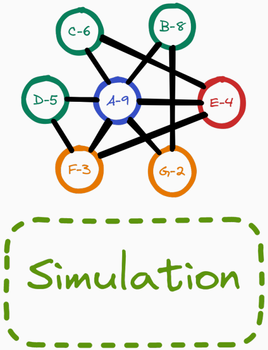
Open a new node:

1 - Use a color already in use in the branch

2 - Open a new color (limit with bounds and score)

MCTS - Phase 3 : Simulation





Complete the coloring

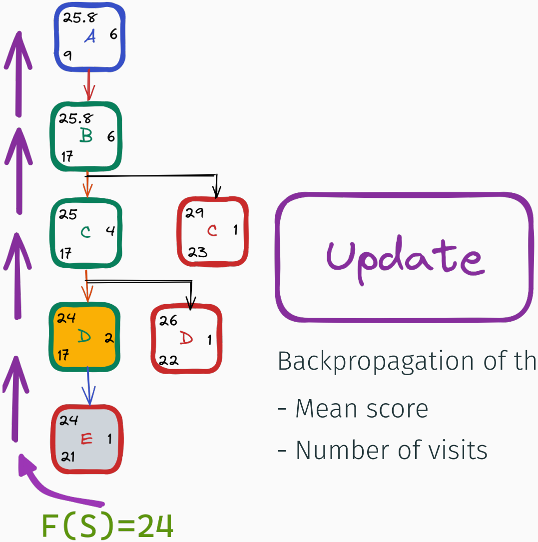
0 - Purely random choice

1 - Greedy Algorithm

2 - Local Search (Grelier *et al.* [2022])

3 - Hyperheuristics (Grelier *et al.* [2023])

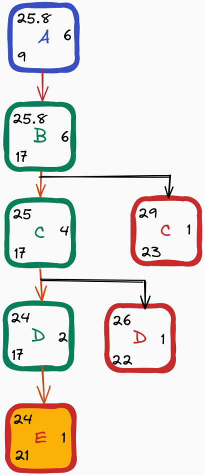
MCTS - Phase 4 : Update

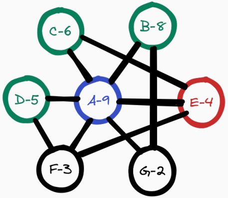


Backpropagation of the score to update :

- Mean score
- Number of visits

MCTS - Phase 1 : Selection...

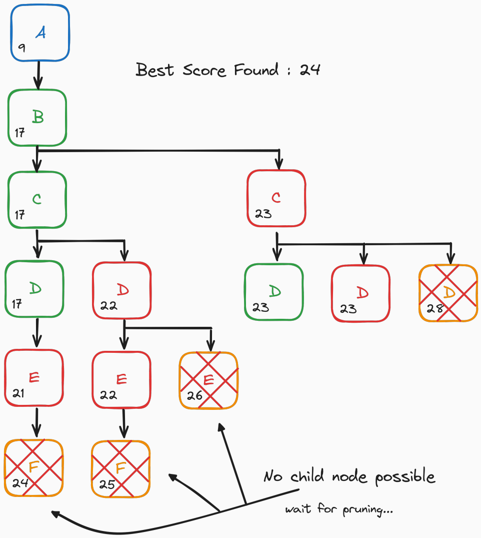


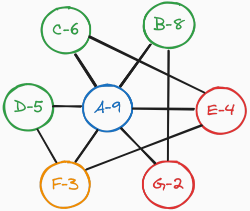


Selection

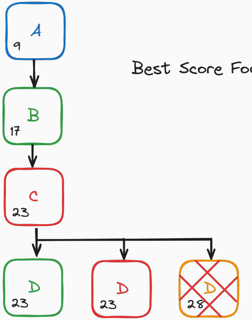
New Iteration...

MCTS - Pruning 1

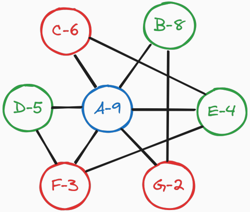




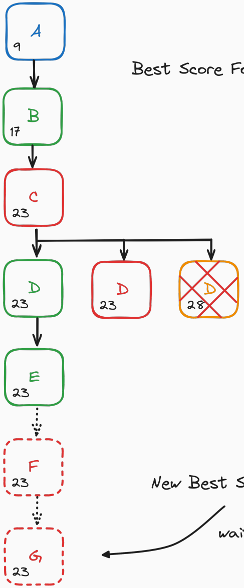
MCTS - Pruning 2



Best Score Found : 24



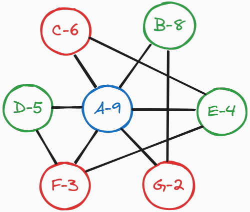
MCTS - Pruning 3



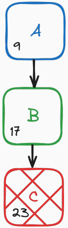
Best Score Found : 24

New Best Score : 23 !

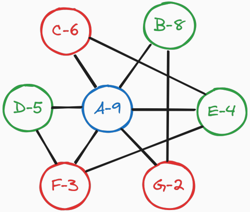
wait for pruning...



MCTS - Pruning 4

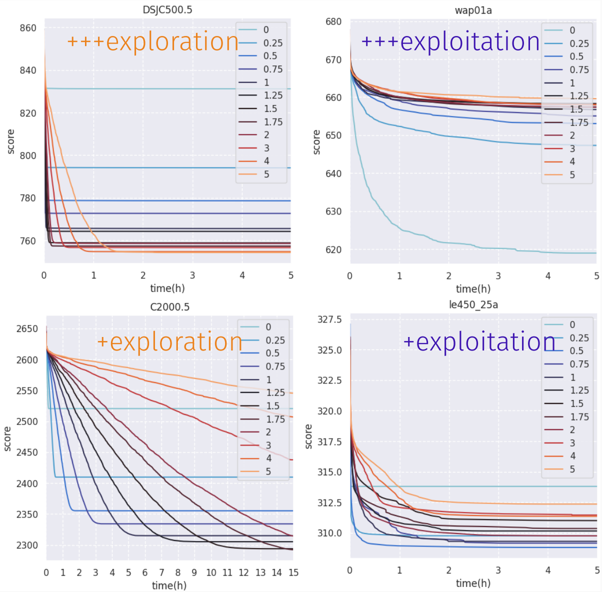


Best Score Found : 23



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

MCTS + Greedy - Results GCP

Instance	BKS	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

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

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	2062	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
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]		[illegible]	[illegible]		[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

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] [2]):
  - 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

[2] optimized by Moalic et Gondran [2018]

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	1558	2332	2340.8	1271	2337	2346.2	6924
C2000.9	5477	**5502**	**5528.1**	3303	6060	6093.6	1107	5910	5969.9	3587	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**	127.8	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	398.4	1746	396	397.8	1143	396	398.2	2701
DSJC250.9	934*	**934**	**935.6**	719	935	935.9	2406	936	942.1	3053	936	936.8	1593	935	936	2643	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	685	**706**	**716.1**	2840	716	719.2	3234	724	735.5	1744	715	720.2	964	715	719	3234	715	720.7	1045	712	720.6	880	715	719.8	715	713	718.9	631
DSJC500.9	1662	**1670**	**1675.1**	946	1687	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	300	305	307.2	1235	307	307.9	1799	305	307.4	1575	304	306.2	3570	304	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	1428	1241	1251.4	2142	1242	1250	1722	1239	1247.8	1554	1240	1247.6	3159
DSJC1000.9	2836	**2840**	**2858.5**	2953	2974	2992.2	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	1499
flat1000_50_0	924	**1155**	**1173.8**	3238	1199	1206.2	336	1222	1235	1713	1203	1210.7	3045	1191	1206.2	630	1201	1208.1	945	1205	1210.6	756	1198	1207.8	3654	1196	1204.6	4370
flat1000_60_0	1162	**1191**	**1205.7**	1080	1238	1244.5	1491	1250	1270	125	1235	1249.5	3045	1242	1247.5	525	1241	1248.2	1995	1232	1247.8	882	1239	1245.3	1701	1238	1245.1	1393
flat1000_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	2121	1211	1222.5	4602
GEOM120a	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
GEOM120b	35*	**35**	35.5	0	**35**	**35**	1	**35**	**35**	0	**35**	**35**	4	**35**	**35**	5	**35**	**35**	4	**35**	**35**	5	**35**	**35**	7	**35**	**35**	3
GEOM120	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**	3	**72**	**72**	2
latin_square_10	1480	**1505**	**1523**	2369	1533	1544.8	1938	1559	1581.2	1369	1546	1552.9	1406	1533	1544.2	1606	1540	1550.6	3268	1522	1546.8	456	1529	1547.2	798	1538	1546.2	4558
le450_25a	306	**306**	307.4	504	**306**	**306**	132	**306**	**306**	175	**306**	**306**	453	**306**	**306**	330	**306**	**306**	349	**306**	**306**	406	**306**	**306**	292	**306**	**306**	424
le450_25b	307*	**307**	313.4	56	**307**	**307**	87	**307**	**307**	10	**307**	**307**	63	**307**	**307**	75	**307**	**307**	59	**307**	**307**	52	**307**	**307**	61	**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	**335.9**	1436	339	342.5	1999	336	336.8	1403	334	336.1	1935	335	336.7	1512	333	336	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	1293	**172**	172.8	1370	**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	222.4	18	**216**	**216.9**	918	**216**	217.3	2018	217	217.3	1408	**216**	**216.9**	2173	**216**	217.2	1868	**216**	217.3	3268	**216**	217.3	1168	217	217.3	1669
queen15_15	223	227	229.7	1110	225	226.2	718	227	228.4	2167	226	226.9	2274	226	226.2	1392	226	226.6	2088	225	226.4	2408	**224**	**226.1**	2590	225	226.2	2647
queen16_16	234	237	239.9	91	237	237.7	1813	238	240.2	2938	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	586.6	2860	**548**	**552**	3263	560	590.3	2596	587	591.6	2420	567	589.6	352	570	590.1	3080	586	591.5	1672	567	589.5	440
wap02a	538	552	586.5	589	551	556.6	90	**540**	**542.7**	2658	552	560.2	1215	554	558	945	551	559.8	945	549	558.5	2520	548	556.5	3105	551	559.1	5815
wap03a	562	**569**	**573.1**	2893	587	588.8	2085	576	578.6	3102	587	589.5	1656	587	589.5	3404	587	588.9	2024	587	589.3	3266	586	588.9	2300	588	589.9	8953
wap04a	563	**564**	**572.5**	3280	581	583.9	1287	569	574.6	3529	583	585.2	1906	583	584.8	2138	582	584	2525	582	584.1	2624	581	584.1	2277	582	585.1	1685
wap05a	541	543	544.5	970	543	544.8	1326	**541**	**543.4**	3306	544	545.8	1139	544	545.7	1397	545	545.6	1929	545	546	1649	543	545.4	578	543	545.3	1571
wap06a	516	**518**	590.4	1005	521	523.5	1332	519	**522.4**	1123	521	526.1	1188	526	529.7	270	523	527.1	756	522	526.6	972	524	528.3	2196	521	527.4	252
wap07a	555	729	745.7	0	566	632.6	770	**564**	**569.5**	2680	567	595.9	210	605	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	1836	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**	9	**2688**	**2688**	10	**2688**	**2688**	7	**2688**	**2688**	10	**2688**	**2688**	8
p42	2466*	**2466**	2522.5	0	**2466**	**2466**	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**	30	**9407**	**9407**	26	**9407**	**9407**	37	**9407**	**9407**	26	**9407**	**9407**	25
r29	8693*	8696	8817.6	0	**8693**	**8693**	539	**8693**	**8693**	1	**8693**	**8693**	23	**8693**	**8693**	14	**8693**	**8693**	18	**8693**	**8693**	13	**8693**	**8693**	17	**8693**	**8693**	21
r30	9816*	9836	9988.2	1	**9816**	9824	2570	**9816**	**9816**	6	**9816**	**9816**	47	**9816**	**9816**	30	**9816**	**9816**	35	**9816**	**9816**	35	**9816**	**9816**	27	**9816**	**9816**	20

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 A
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

MCTS + Hyperheuristics - Results WVCP

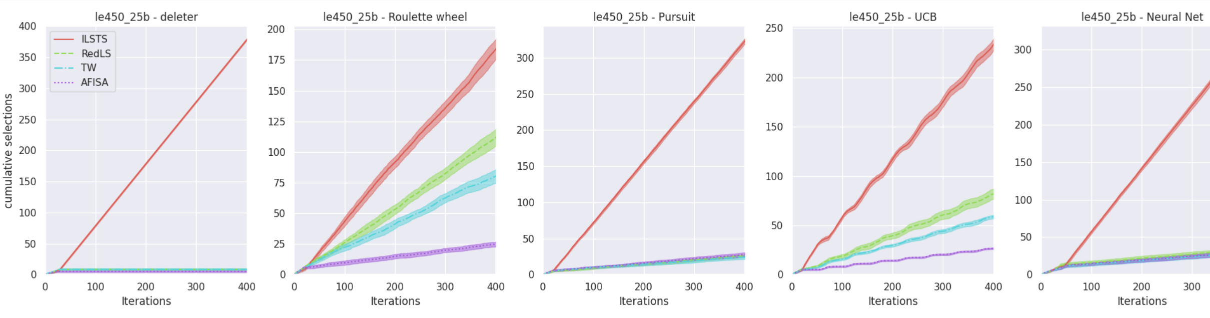
MCTS + LS (Local Search)

- 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)

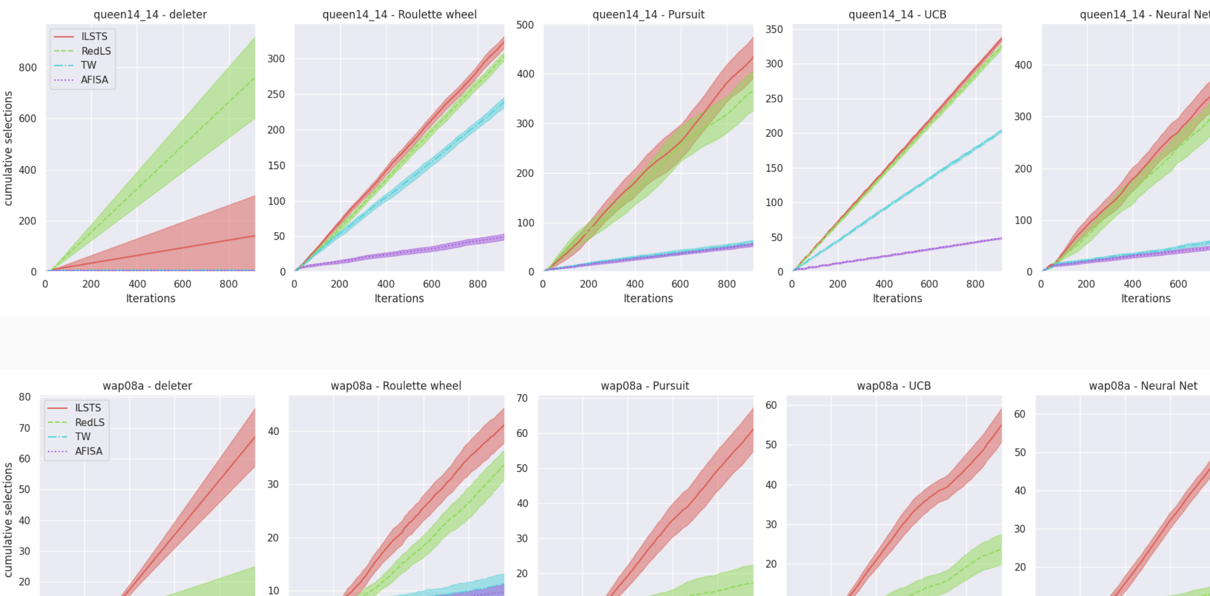
MCTS + HH (Hyperheuristics, with selection criteria)

- 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

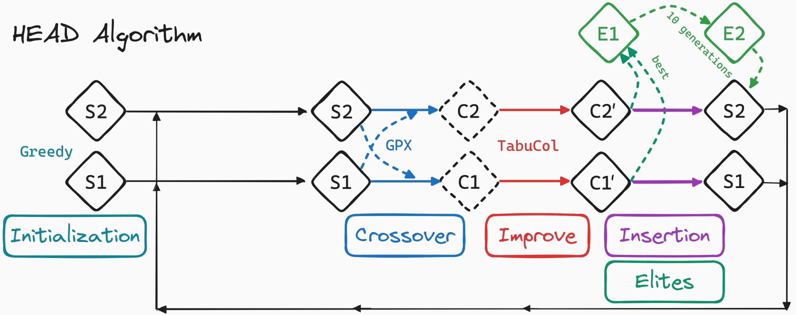


MCTS + Hyperheuristics - Selection 2/2



AHEAD

HEAD - Hybrid Evolutionary Algorithm in Duet



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

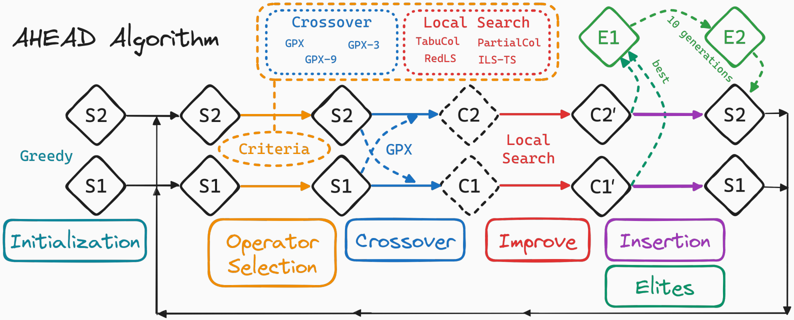
Why HEAD?

- **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



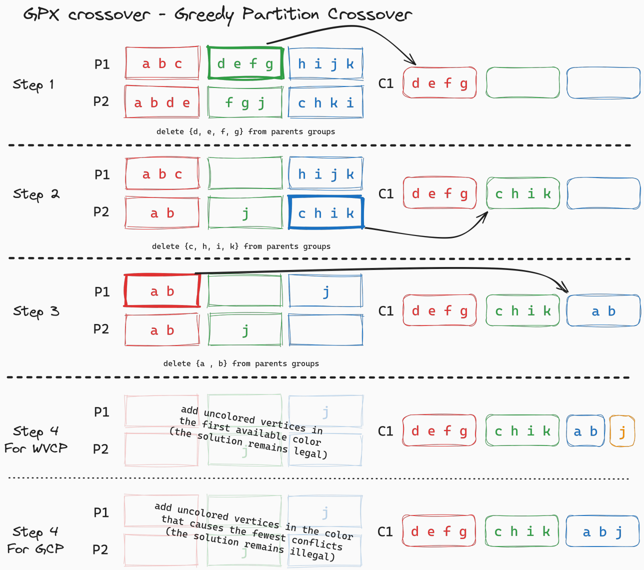
Why AHEAD?

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

Which Operators?

- 3 GPX crossovers (± conservative)
- 2 local searches/problem
  - GCP: TabuCol, PartialCol
  - WVCP: RedLS, ILS-TS

GPX - Galinier et Hao [1999]



AHEAD - Operators Selection

Crossover

- **GPX** : 1 color in P1 for 1 color in P2
- **GPX-3** : 3 colors in P1 for 1 color in P2
- **GPX-9** : 9 colors in P1 for 1 color in P2

GCP - Local Search

- **TabuCol** : Hertz et Werra [1987]
- **PartialCol** : Blöchliger et Zufferey [2008]

WVCP - Local Search

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

Criteria

- Same as for MCTS: Random, Deleter, Roulette, Pursuit, UCB, NN
- Reward : Score of the solution after the local search
- Choice of a pair of operators <crossover, local search>
- Exception : NN : Generate all crossovers and select the best one

AHEAD - WVCP - Comparison between methods

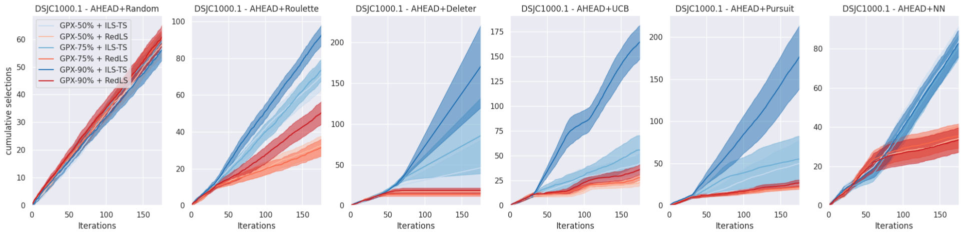
1 point/instance if the average is significantly better for the method on the line compared to the one in the column (Wilcoxon signed-rank, p-value < 0.001)

/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

AHEAD - WVCP - Results

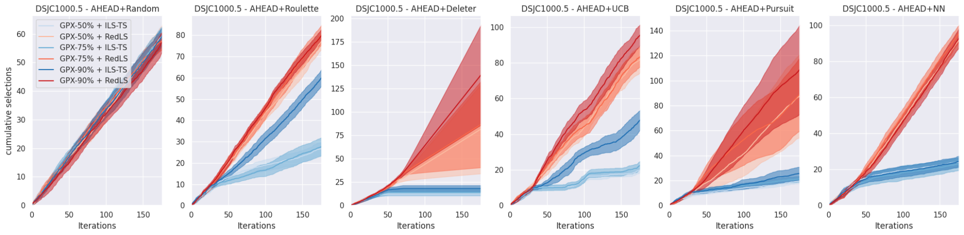
instance	BKS	RedLS			ILS-TS			HEAD+RedLS			AHEAD+Random			AHEAD+Deleter		
		best	mean	time	best	mean	time	best	mean	time	best	mean	time	best	mean	time
C2000.5	2144	2131	2155.7	18367	2244	2264.4	6423	2244	2257.9	7453	2220	2236.8	12962	2218	2236.3	1782
C2000.9	5477	5439	5455.1	23137	5847	5910.1	23014	5732	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_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		

AHEAD - WVCP - Selections



No large differences in selections between the different crossovers.

Choice of the local search is more important.



AHEAD - GCP - Comparison between methods

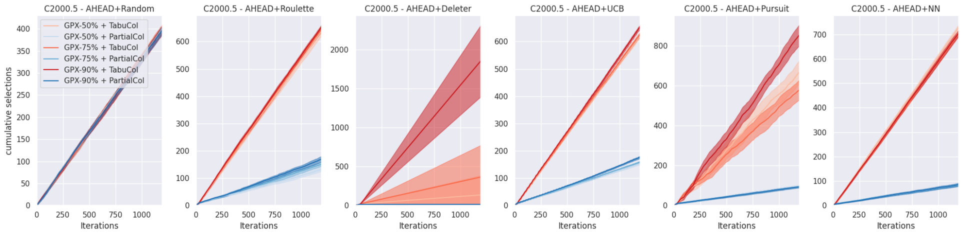
1 point/instance if the average is significantly better for the method on the line compared to the one in the column (Wilcoxon signed-rank, p-value < 0.001)

/31	PartialCol	TabuCol	HEAD+PC	HEAD+TC	Random	Roulette	Deleter	UCB	Pursuit	NN	# BKS	# Best	# Best Avg
PartialCol	-	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

AHEAD - GCP - Results

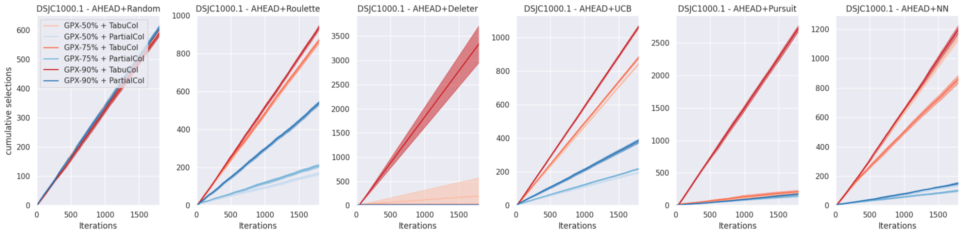
instance	BKS	PartialCol			TabuCol			HEAD+TC			AHEAD+Random			AHEAD+Deleter		
		best	mean	time	best	mean	time	best	mean	time	best	mean	time	best	mean	time
C2000.5	145	164	165.2	5313	162	162.8	4628	**148**	**149.2**	3330	150	150.7	3101	149	150.7	3152
C2000.9	408	420	420.8	5171	411	412.5	4786	405	406.4	2328	405	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	1616	**223**	224.2	2734	**223**	**223.8**	1589
DSJR500.5	122*	125	126.2	1666	124	127	1155	**123**	124	1766	**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		

AHEAD - GCP - Selections



No large differences in selections between the different crossovers.

Choice of the local search is more important.



AHEAD - Conclusion

GCP

- 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

- 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)

Hyperheuristics

- **Conservative crossover** preferred
- **Local Search choice more important**
- Best criteria are Deleter and Pursuit, no changes of choice during search

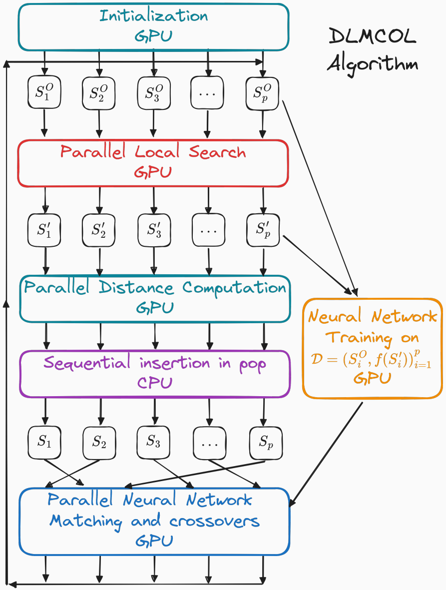
DLMCOL

DLMCOL - Deep Learning Guided Memetic framework

- **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



DLMCOL - Parameter settings

Table 3: Parameter setting in DLMCOL

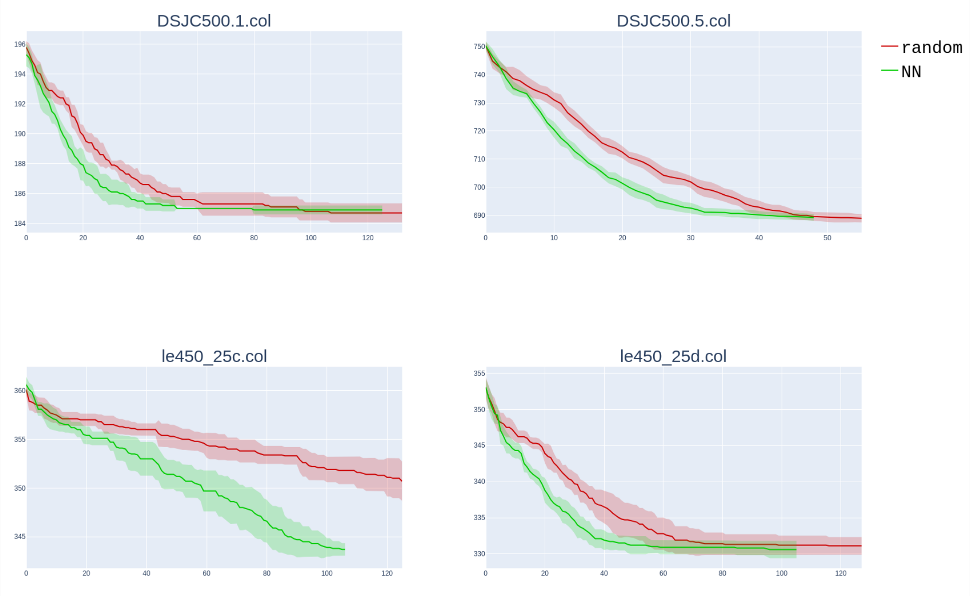
Parameter	Description	Value
p	Population size	20480 (8192)
$maxLSIters$	Number of iterations local search	10
$nbIter_{TS}$	Number of iterations tabu search	$10 \times \lvert V \rvert$
α	Tabu tenure parameter	0.2
MS	Minimum spacing between two individuals	$\frac{\lvert V \rvert}{10}$
l_r	Learning rate 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 $10|V|$, $5|V|$, $2|V|$, $2|V|$, $2|V|$, $2|V|$, $2|V|$, $|V|$ and $|V|//2$.

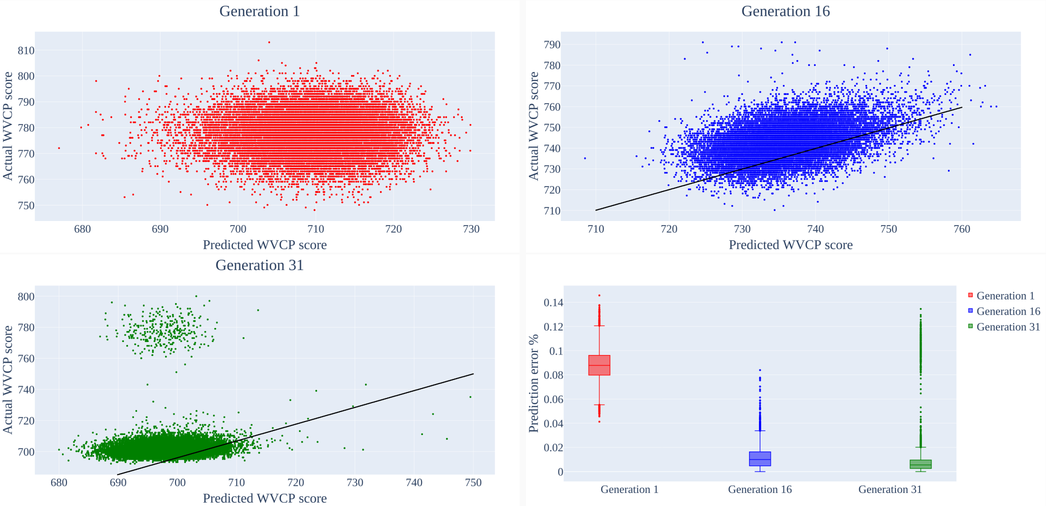
DLMCOL - New Best Scores for WVCP

Instance	$\lvert V \rvert$	previous $\text{score}_{\text{best}}$	new $\text{score}_{\text{best}}$	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

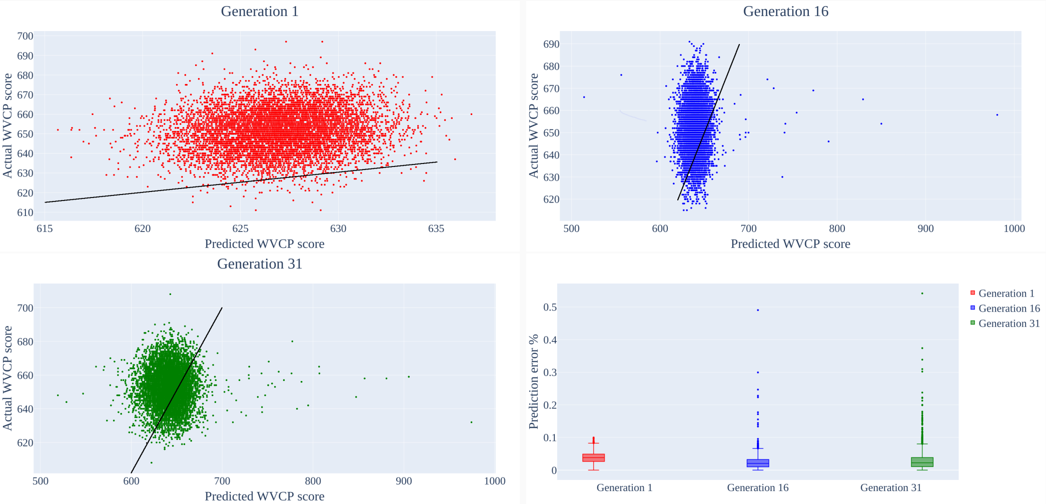
DLMCOL - Learning vs Random crossover choice



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

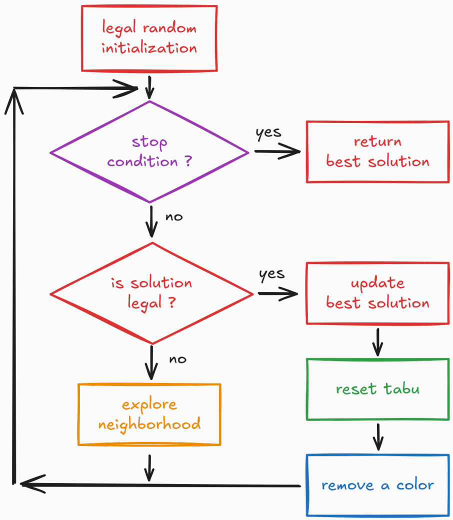


DLMCOL - Predicted vs Actual Score (wap05a)



TabuEdges

TabuEdges - Algorithm



- Color removing
  - Fusion: merge two colors
  - Divide: separate to multiple colors
- Tabu Management
  - Tabu List
  - Tabu Matrix
  - Configuration Checking
- Neighborhood Exploration (one-move)
  - TabuCol
  - TabuEdge
  - TabuDouble
- Guided Component
  - Increment weight/penality of edges in conflict
  - Use weight as delta penalty when searching moves

TabuEdges -Color Removing

To decrease the number of colors when a legal solution is found

- Fusion: merge two colors groups
- Divide: separate vertices from a color group to multiple color groups

Low impact on the search

TabuEdges -Tabu Strategy

Different solutions exists to manage the tabu aspect.

Let's consider the vertex v that move from the color c to another color

$|V|$: number of vertices in the graph

$|C|$: number of conflicts in the solution

- **Tabu List**: size $|V|$ - Hertz et Werra [1987]
  $tabu[v] = turn + random(0, 10) + \alpha * |C|$
- **Tabu Matrix**: size $|V| * k_{max}$ - Moalic et Gondran [2018]
  $tabu[v][c] = turn + random(0, 10) + \alpha * |C|$
- **Configuration Checking**: size $|V|$ - Cai *et al.* [2011]
  $tabu[v] = true$
  $tabu[n] = false, \forall n \in neighbors(v)$

TabuEdges - Neighborhood Exploration and Guided Component

Neighborhood: 1-opt / one-move : one vertex move to another color

Neighborhood Selection Strategy:

1. TabuCol, Hertz et Werra [1987]
    - select the best non tabu move (can degrade the solution)
    - aspiration critera to accept a tabu move that lead to a better solution
2. TabuEdges
    1. apply a non tabu and non degrading move
    2. if no move is found, penalize conflicting edges*
    3. then move a random vertex in conflict in the least degrading way
3. TabuDouble
    - alternate TabuEdges and TabuCol strategies

***Guided Component**, Wang *et al.* [2020]:

- Increment weight (penality) of the edges in conflict
- Use the weight to compute penalty and delta while searching moves

TabuEdges - Comparison between Tabu and Neighborhood Strategies

20 runs of 1h, 31 hardest GCP instances

Compare with Wilcoxon signed-rank, p-value < 0.001

- TabuCol's best tabu strategy is the Tabu Matrix
- TabuEdges's best tabu strategies are:
  - Tabu List (2 new best scores)
  - Tabu Matrix (1 new best score)
- TabuEdges better than TabuCol on more instances (12 againt 8) and reach more best scores
- TabuDouble (alternate between the two):
  better than TabuCol but not TabuEdges

TabuEdges - Comparison with State of the Art

k-TabuCol, k-HEAD and k-AHEAD solve *k*-coloring (1h max to solve 1 color)

/31 instances	# BKS	# Best Score	# Best Mean
TabuCol TL	4	7	5
TabuCol TM	5	8	6
TabuCol CC	0	0	0
TabuEdge TL	14(**2**)	17	**15**
TabuEdge TM	**15**(1)	18	11
TabuEdge CC	14	15	12
TabuDouble TL	14	15	10
TabuDouble TM	13	14	10
k-TabuCol	5	8	7
k-HEAD	7	17	14
k-AHEAD	11	**19**	12

TL : TabuList, TM : TabuMatrix, CC: Configuration Checking, BKS : Best Known Score

TabuEdges - Comparison between Tabu and Neighborhood Strategies

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	TC L	TC M	TC CC	TE L	TE M	TE CC	TD L	TD M	# BKS	# Score	# Mean
TC L	-	3	**21**	5	6	7	5	6	4	15	9
TC M	**15**	-	**31**	8	8	9	6	9	5	16	14
TC CC	0	0	-	0	0	0	0	0	0	0	0
TE L	**14**	**13**	**31**	-	0	**7**	**7**	**5**	14(2)	21	**19**
TE M	**14**	**12**	**31**	**2**	-	**7**	**6**	**3**	**15**(1)	**22**	14
TE CC	**13**	**11**	**31**	1	1	-	4	1	14	19	14
TD L	**12**	**11**	**31**	2	2	**7**	-	2	14	20	13
TD M	**13**	**11**	**31**	1	1	**4**	**3**	-	13	19	13

TC : TabuCol, TE : TabuEdges, TD : TabuDouble, L : TabuList, M : TabuMatrix

CC: Configuration Checking, BKS : Best Known Score

TabuEdges - Comparison with State of the Art

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	TC M	TE L	TE M	TD	k-TC	k-HEAD	k-AHEAD	# BKS	# Score	# Mean
TC M	-	8	8	6	0	1	0	5	8	6
TE L	**13**	-	0	**7**	**11**	**11**	**10**	14(**2**)	17	**15**
TE M	**12**	**2**	-	**6**	**11**	**11**	**10**	**15**(1)	**18**	11
TD	**11**	2	2	-	**11**	**11**	10	14	15	10
k-TC	**10**	8	8	10	-	1	0	5	8	7
k-HEAD	**15**	8	8	9	**12**	-	1	7	17	14
k-AHEAD	**18**	9	9	10	**14**	1	-	11	19	12

TC : TabuCol, TE : TabuEdges, TD : TabuDouble, L : TabuList, M : TabuMatrix

CC: Configuration Checking, k- : solve k-coloring problem, (GCP otherwise)

TabuEdges - Results - State of the Art

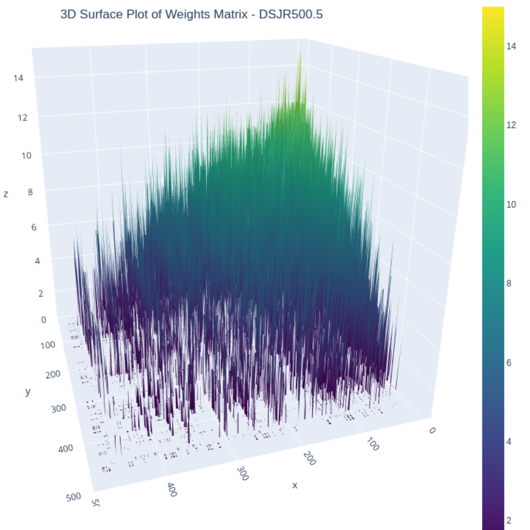
instance	BKS	TabuCol TM		TabuEdge TL		TabuEdge TM		TabuDouble TL		k-TabuCol		k-HEAD		k-AHEAD	
		best	time	best	time	best	time	best	time	best	time	best	time	best	time
C2000.5	145	164	2445	166	3290	166	3167	165	663	162	2564	150	3300	151	3273
C2000.9	404	413	3491	423	3418	421	3184	422	1525	412	2225	406	2527	406	3139
C4000.5	259	306	2821	319	3410	317	3499	307	540	304	2332	283	3492	284	3156
DSJC500.1	12	12	813	13	0	13	0	12	1449	12	58	12	107	12	103
DSJC500.5	47	49	239	50	430	50	751	49	2594	49	573	48	1239	48	1663
DSJC500.9	126	126	1033	127	149	126	819	127	2249	126	1975	126	1188	126	530
DSJC1000.1	20	21	2	21	193	21	366	21	16	21	0	21	1	20	2496
DSJC1000.5	82	88	2919	90	1362	90	2318	90	1315	88	1544	83	2811	83	2793
DSJC1000.9	222	224	1870	224	1894	223	2945	227	409	224	3296	223	2272	223	3079
flat300_28_0	28*	28	1608	28	2099	28	2584	28	620	30	1765	30	1605	30	1228
flat1000_50_0	50*	50	920	50	16	50	25	50	15	50	27	50	69	50	18
flat1000_60_0	60*	60	2845	60	1403	60	1327	60	1468	60	41	60	112	60	57
flat1000_76_0	76*	87	1994	89	2229	89	1728	89	1413	86	2678	82	3064	83	2280
latin_square_10	97	101	2692	101	1014	101	1238	101	3213	100	1014	101	2591	99	2859
le450_25c	25*	26	0	26	0	26	0	26	0	26	0	26	0	25	1328
le450_25d	25*	26	0	26	0	26	0	26	0	26	0	26	0	25	1655
DSJR500.5	122*	125	842	122	1	122	0	122	1	126	231	124	1869	124	1498
r250.5	65*	66	1095	65	0	65	0	65	0	67	129	65	2816	65	1556
r1000.1c	98	115	254	98	21	98	9	98	24	133	0	100	2306	101	1699
r1000.5	234	245	2383	234	49	234	41	234	44	245	1387	246	2545	244	1862
wap01a	41*	42	62	41	7	41	4	41	7	42	1069	42	219	41	2774
wap02a	40*	41	1462	40	7	40	4	40	7	41	231	41	26	40	2844
wap03a	43	47	2380	42	2143	42	2811	43	220	45	332	45	510	44	2008
wap04a	41	43	2036	41	2235	41	1486	42	138	42	755	43	1736	43	612
wap06a	40*	41	778	40	1	40	0	40	0	40	1616	40	1103	40	558
wap07a	41	42	1634	40	1296	41	19	41	27	42	442	42	1667	42	968
wap08a	40*	41	1046	40	130	40	150	40	68	41	915	42	107	41	1568
#BKS		5		14		15		14		5		7		11	
#Best		8		17		18		15		8		17		19	
#Best Avg		6		15		11		10		7		14		12	

TabuEdges - Results - Heterogeneous vs Homogeneous Degree

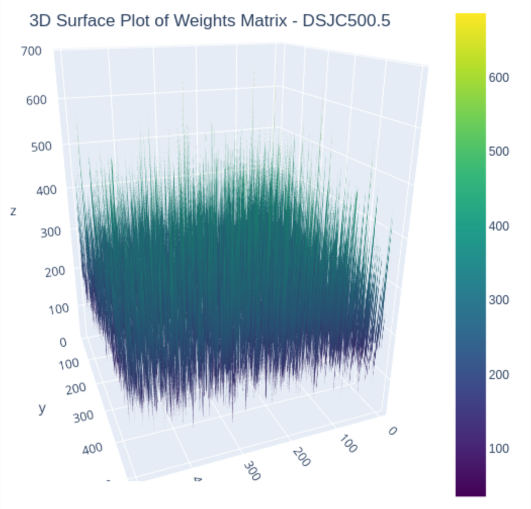
instance	BKS	TabuCol TM best	TabuCol TM time	TabuEdge TL best	TabuEdge TL time	TabuEdge TM best	TabuEdge TM time	TabuDouble TL best	TabuDouble TL time	k-TabuCol best	k-TabuCol time	k-HEAD best	k-HEAD time	k-AHEAD best	k-AHEAD time
C2000.5	145	164	2445	166	3290	166	3167	165	663	162	2564	150	3300	151	3273
C2000.9	404	413	3491	423	3418	421	3184	422	1525	412	2225	406	2527	406	3139
C4000.5	259	306	2821	319	3410	317	3499	307	540	304	2332	283	3492	284	3156
DSJC500.1	12	12	813	13	0	13	0	12	1449	12	58	12	107	12	103
DSJC500.5	47	49	239	50	430	50	751	49	2594	49	573	48	1239	48	1663
DSJC500.9	126	126	1033	127	149	126	819	127	2249	126	1975	126	1188	126	530
DSJC1000.1	20	21	2	21	193	21	366	21	16	21	0	21	1	20	2496
DSJC1000.5	82	88	2919	90	1362	90	2318	90	1315	88	1544	83	2811	83	2793
DSJC1000.9	222	224	1870	224	1894	223	2945	227	409	224	3296	223	2272	223	3079
flat300_28_0	28*	28	1608	28	2099	28	2584	28	620	30	1765	30	1605	30	1228
flat1000_50_0	50*	50	920	50	16	50	25	50	15	50	27	50	69	50	18
flat1000_60_0	60*	60	2845	60	1403	60	1327	60	1468	60	41	60	112	60	57
flat1000_76_0	76*	87	1994	89	2229	89	1728	89	1413	86	2678	82	3064	83	2280
latin_square_10	97	101	2692	101	1014	101	1238	101	3213	100	1014	101	2591	99	2859
le450_25c	25*	26	0	26	0	26	0	26	0	26	0	26	0	25	1328
le450_25d	25*	26	0	26	0	26	0	26	0	26	0	26	0	25	1655
DSJR500.5	122*	125	842	122	1	122	0	122	1	126	231	124	1869	124	1498
r250.5	65*	66	1095	65	0	65	0	65	0	67	129	65	2816	65	1556
r1000.1c	98	115	254	98	21	98	9	98	24	133	0	100	2306	101	1699
r1000.5	234	245	2383	234	49	234	41	234	44	245	1387	246	2545	244	1862
wap01a	41*	42	62	41	7	41	4	41	7	42	1069	42	219	41	2774
wap02a	40*	41	1462	40	7	40	4	40	7	41	231	41	26	40	2844
wap03a	43	47	2380	42	2143	42	2811	43	220	45	332	45	510	44	2008
wap04a	41	43	2036	41	2235	41	1486	42	138	42	755	43	1736	43	612
wap06a	40*	41	778	40	1	40	0	40	0	40	1616	40	1103	40	558
wap07a	41	42	1634	40	1296	41	19	41	27	42	442	42	1667	42	968
wap08a	40*	41	1046	40	130	40	150	40	68	41	915	42	107	41	1568
#BKS		5		14		15		14		5		7		11	
#Best		8		17		18		15		8		17		19	
#Best Avg		6		15		11		10		7		14		12	

TabuEdges - Heterogeneous vs Homogeneous Degree

link DSJR500.5



link DSJC500.5



TabuEdges - Conclusion

Cons

- Pretty bad results on graphs with homogeneous degree
- Doesn't work well on k-coloring

Pros

- Quite fast and very efficient on graphs with heterogeneous degree
- Works well on GCP

Other

- 2 new best scores (wap03a, wap07a)
- Optimal score for wap07a thanks to lower bound from Heule *et al.* [2022]

TabuEdges - What's next?

Elements to explore

- Reduce weights or reset weights
- **Guided Memetic Algorithm**
- **aspiration criteria** for the weighted moves
- other **tabu strategies**
- impact of a **swap move**
- go toward an **iterated local search framework** where the weights on edges are used only during a perturbation phase