

Memetic Algorithms for Tree Decomposition

Masterstudium:
Software Engineering & Internet Computing

Kevin Bader

Technische Universität Wien
Institut für Informationssysteme
Abteilung für Datenbanken und Artificial Intelligence
Betreuer: Priv.-Doz. Dr. Nysret Musliu

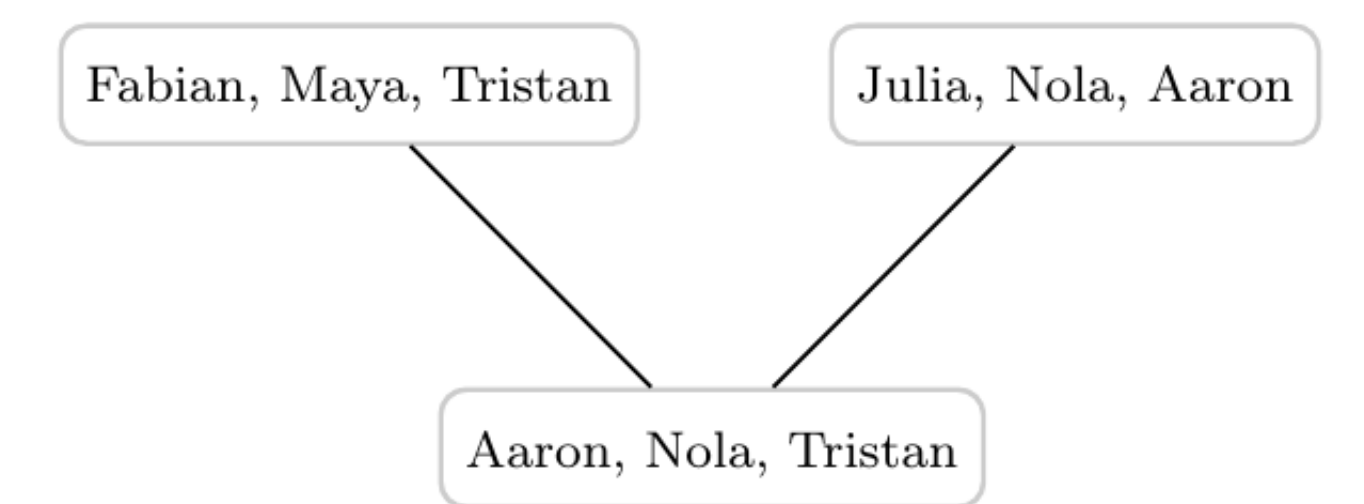
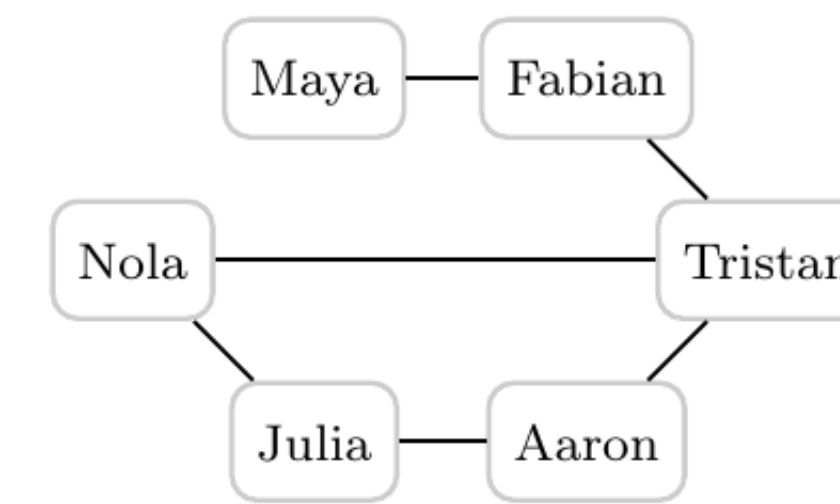
Motivation

- Many instances of NP-hard problems can be solved efficiently if their treewidth is small
- Finding a tree decomposition of minimal width is NP-hard
- Exact and heuristics algorithms exist but there is still a big gap between lower and upper bounds

Objectives

- The development of new memetic algorithms for minimizing treewidth upper bounds
- An evaluation of the new algorithms on benchmark instances from the literature
- A comparison with state-of-the-art algorithms

Tree decomposition by example

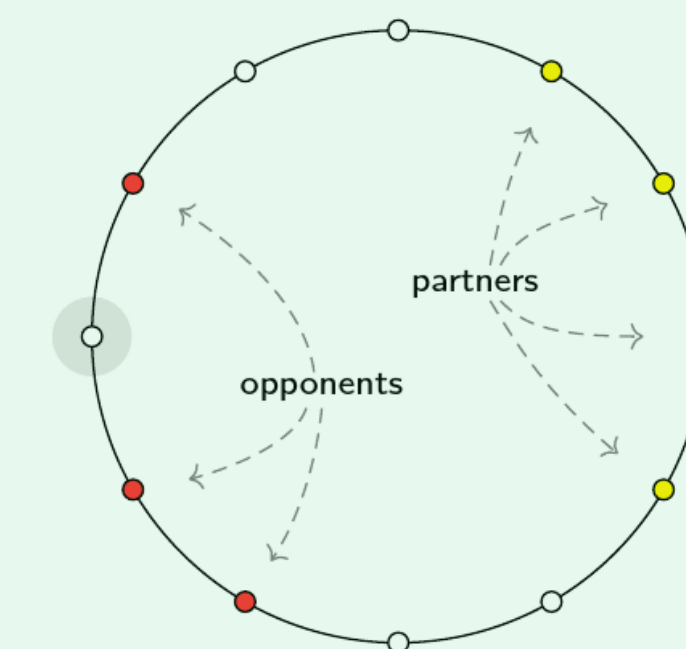


Imagine that this graph represents children and their relationships, and that two children know each other if there is an edge between them. Suppose the children should be assigned to rooms such that no two children in a room know each other. Is this possible with two rooms?

For 3 rooms or more this is an NP-complete problem (the graph coloring problem). Still, it is solvable efficiently if the treewidth of the instance is small. The graph above is one possible tree decomposition of the graph on the left. The tree nodes represent subproblems of the instance; if they are of small width (have few vertices in them), the subproblems can be solved more efficiently, and ultimately the original instance.

Three new memetic algorithms for treewidth upper-bound minimization have been proposed:

- MA1 organizes a population of solutions in a randomly initialized ring structure, used for employing selection and recombination among the individual solutions according to their vicinity
- MA2 is a hybrid between a genetic algorithm that uses the plus strategy in combination with elitism for selection, and a local search heuristic that is used to improve the best individuals in every generation
- MA3 reuses an existing implementation of a genetic algorithm, and applies a local search based heuristic to a random fraction of the population at each iteration

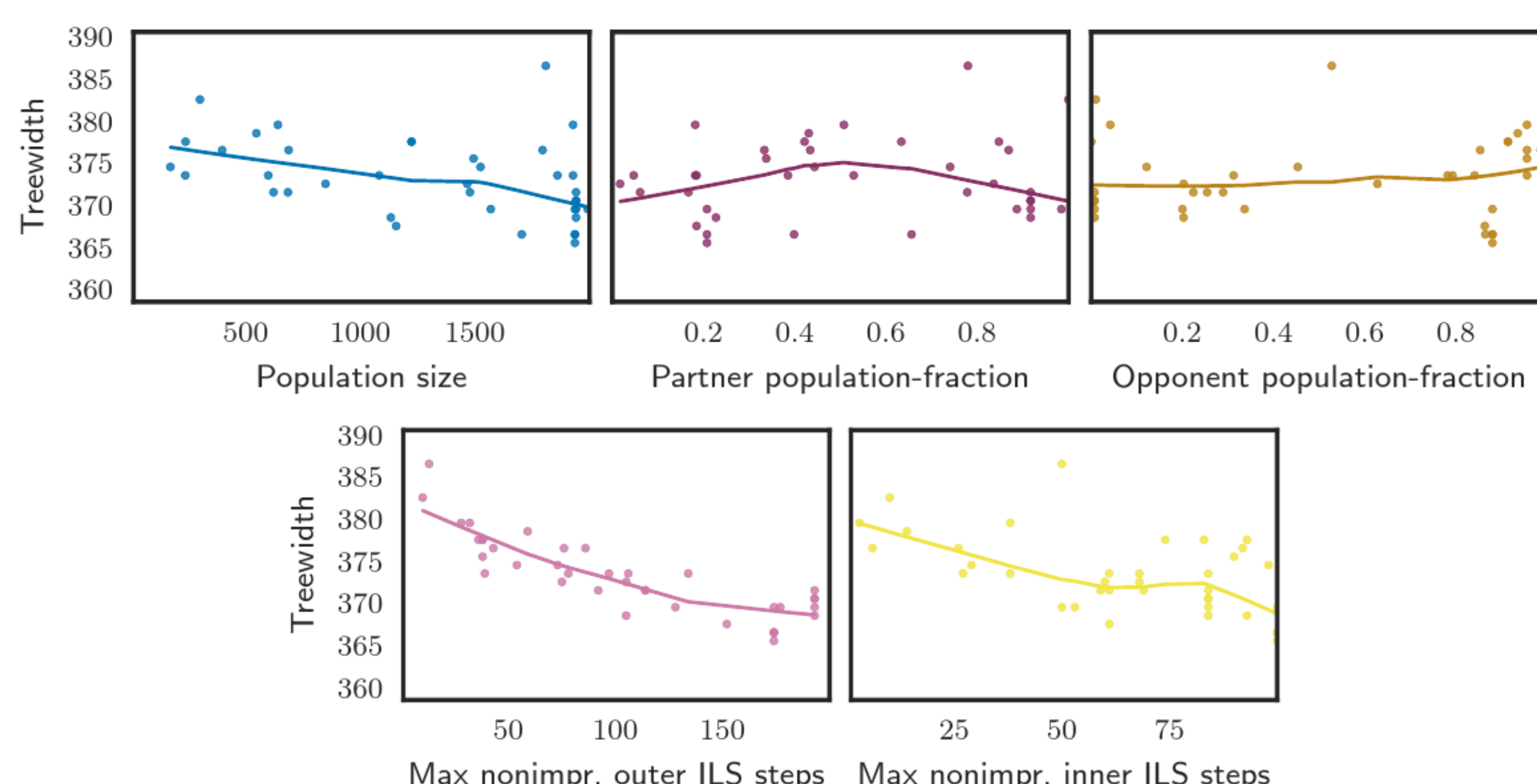


- Improved best known upper bounds on 8 instances of the DIMACS benchmark
- Comparison with state-of-the-art methods on training and validation instance sets
- Parameter tuning and parameter correlation study

Experimental Evaluation

Training Instances: Second DIMACS Implementation Challenge

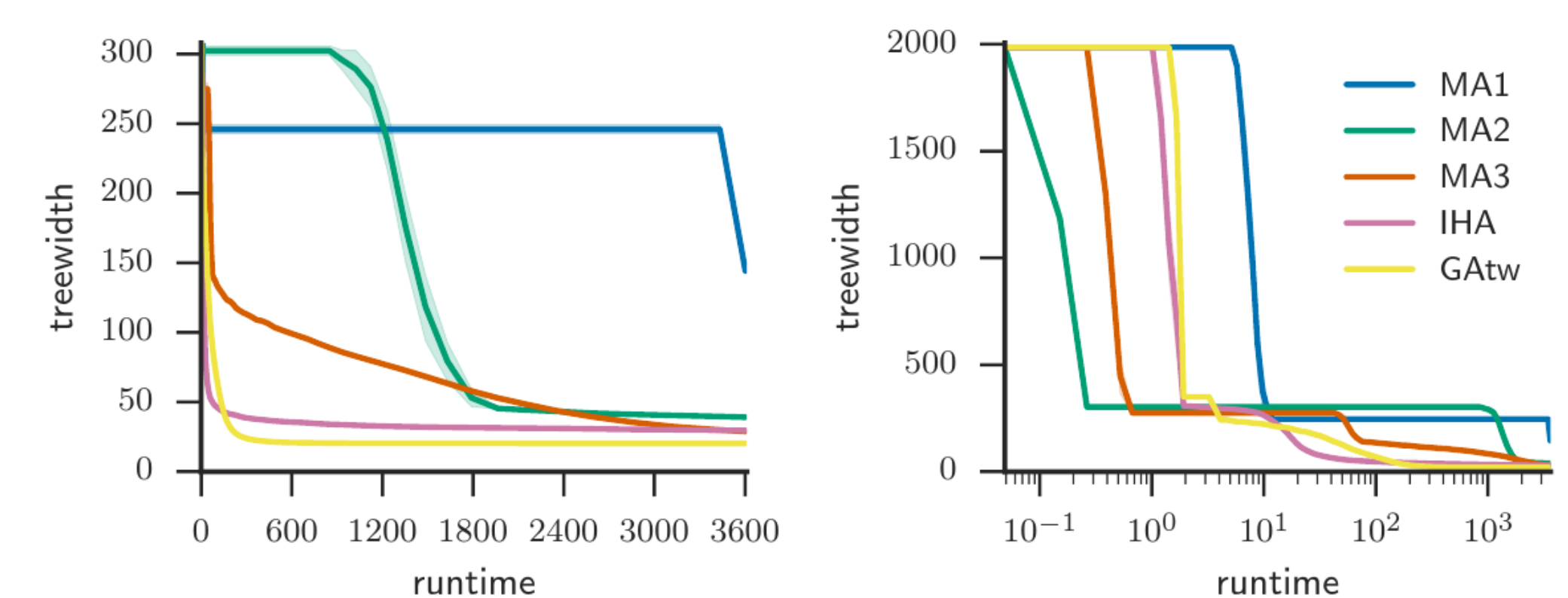
- 79 instances, 20 runs with 1h runtime each
- Comparison with state-of-the-art algorithms:
 - BB-tw (Bachooore and Bodlaender, 2006)
 - QuickBB (Gogate and Dechter, 2004)
 - TabuTW (Clautiaux et al., 2004)
 - ACS+ILS (Hammerl, 2009)
 - IHA (Musliu, 2008)
 - GA-tw (Schafhauser, 2006)
- For the two best state-of-the-art algorithms, IHA and GA-tw, the experiments on the training instances have been repeated using the same hardware, platform and time limit
- Memetic algorithms competitive on most instances
- They improve 8 best known upper bounds
- Evaluation of parameter correlations for the memetic algorithms



MA1 applied to instance le450_15c, 38 samples.

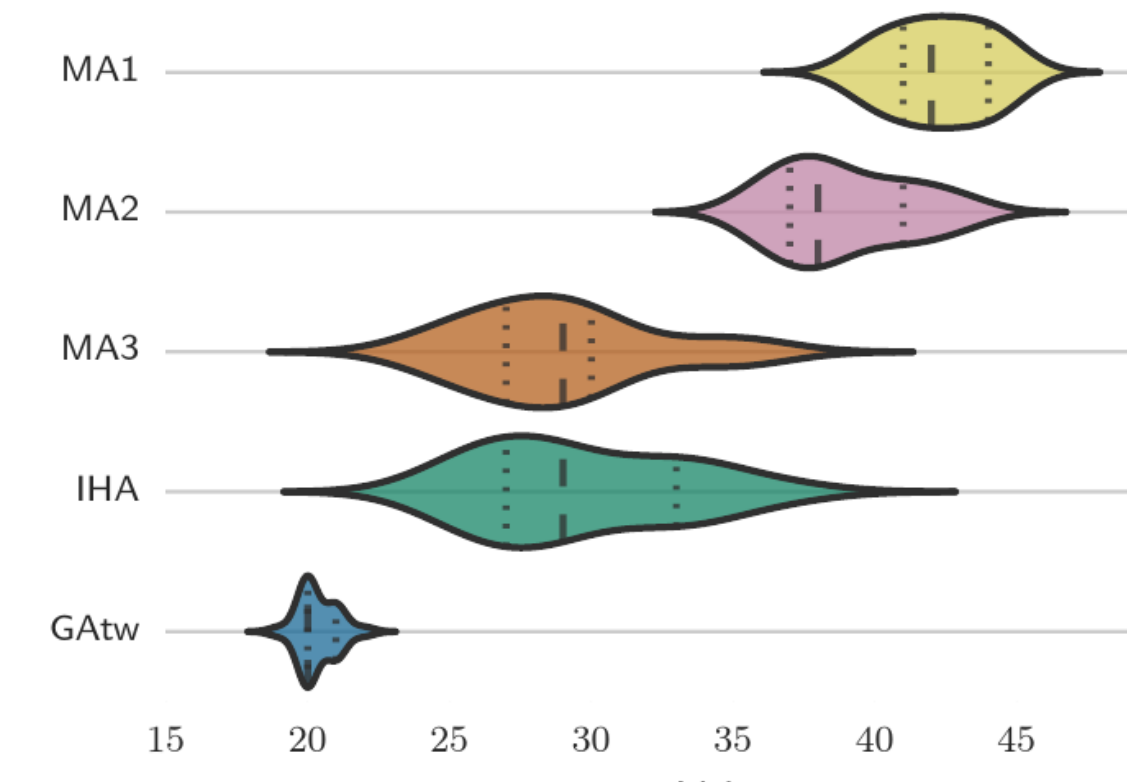
Comparison on Validation Instances

- 27 instances, 20 runs with 1h runtime each
- 5 Algorithms: our memetic algorithms (MA1, MA2, MA3) vs IHA and GA-tw
- Comparing time response



Solution quality over time for instance pcb3038
(linear time scale left, logarithmic time scale right).

- Comparing mean performance



Violin plot for instance pcb3038.

- Overall result

Result of the comparisons on the validation instances. Dark areas represent instances where the respective side performs significantly better than the other (significance level 5%). The remaining light areas in between represent instances where no performance difference could be identified. The vertical black bars signify the result of the respective comparison.

