Algorithm Engineering – Exercise 4

Team 02: Julian Fechner, Julio Cesar Perez Duran, Denis Koshelev

1. Implemented Features

We implemented six heuristics, one of them from the lecture (**Greedy Maximum Degree H0**) to approximate the minimum vertex cover:

- H1 **Random edge:** Take both vertices of uncovered edge
- H2 **Clever Steady Strategy Algorithm (CSSA)** [1]: Take a minimum degree neighbor of the minimum degree vertex.
- H3 **Modified Clever Steady Strategy Algorithm**: Take a maximum degree neighbor of the minimum degree vertex.
- H4 **Savage's Vertex Cover Algorithm:** [2]: Construct a depth-first search spanning tree with the graph and then add all vertices to the vertex cover with the exception of the leaves.
- H5 **ConstructVC** [3]: The algorithm consists in iterating through the edges of the graph and taking the vertex with the highest degree into the vertex cover

We also modeled our problem to solve it with a **Weighted SAT-Solver** (python-sat library) and with an **LP-Solver** (CPLEX ILP).

2. Data Structures

For our best heuristic we need to track the minimum degree vertex with its neighbors. Thus, we implemented a data structure that keeps track of it. It is a combination of a hash map with degrees and corresponding vertices and a sorted set of degrees. This allows us to efficiently find and remove the minimum degree vertex from the graph.

3. Highlights

- With the improved data structure we managed to solve all graphs within the 60 seconds timeout, which we couldn't accomplish before.
- Our solver from the last submission solves 119 instances, which is 14 instances more than the SAT-Solver but 40 instances less than the ILP-Solver.

4. Experiments

- **Comparison: SAT, ILP and Reduction**
  We tested our weighted SAT-Solver and ILP-Solver with and without reduction rules. The SAT-Solver with them solved 12 additional test cases more than without, but we could not see an improvement for the ILP-Solver with them. Also, we compared the best solver versions with each other and from Figure 1 it's clear that ILP solved significantly more test cases.
- **Comparison: All Heuristics** (Figure 2)
  We compared all six heuristics and took the average of the percentage error as the metric to evaluate them. The heuristic with the minimum average error among all instances is the modified CSSA with the CSSA coming in a close second. The ConstructVC heuristic comes close in third place while the other heuristics are significantly worst than the Greedy Maximum Degree with the smallest error among them being 14,209 %, especially the random edge one, in which 56 test cases are on the border of the 2 approximation factor.

Heuristic	H0	H1	H2	H3	H4	H5
Avg.Error	1,585	52,116	0,607	0,602	14,209	3,178
Max. Error	8,633	100	5,147	5,147	82,715	37,007

- **Comparison: Modified CSSA and Greedy**
  On the left plot of Figure 3 we picked out the top four graphs where Modified CSSA performed better than Greedy. For instance, the biggest difference was observed on the `vc157.dimacs` graph with 132 vertices less in the optimal solution which is $4,4\%$ of the graph size. Only three graphs where Greedy performed better can be seen on the right plot with the maximum difference of 4 vertices for `vc049.dimacs` which is 2% of the graph size. Even though both algorithms' solutions are always not so far from each other, Modified CSSA managed to solve 17 instances more accurate.

References

[1] M. Fayaz, S. Arshad, Clever steady strategy algorithm: A simple and efficient approximation algorithm for minimum vertex cover problem, 2015, pp. 277–282. doi:10.1109/FIT.2015.55.

[2] C. Savage, Depth-first search and the vertex cover problem, Information Processing Letters 233–235.
URL https://www.sciencedirect.com/science/article/pii/0020019082900229

[3] S. Cai, Balance between complexity and quality: Local search for minimum vertex cover in massive graphs, AAAI Press, 2015, p. 747–753.

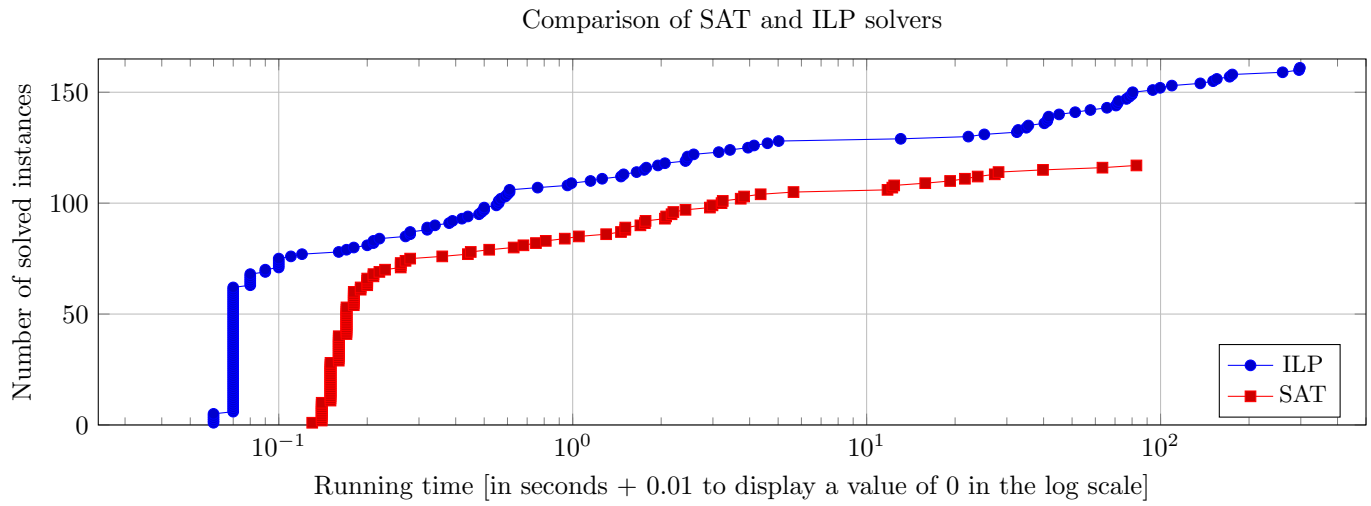

Figure 1: Comparison of ILP and SAT solvers. Timeout was set to 300 and 10 instances were allowed to fail.

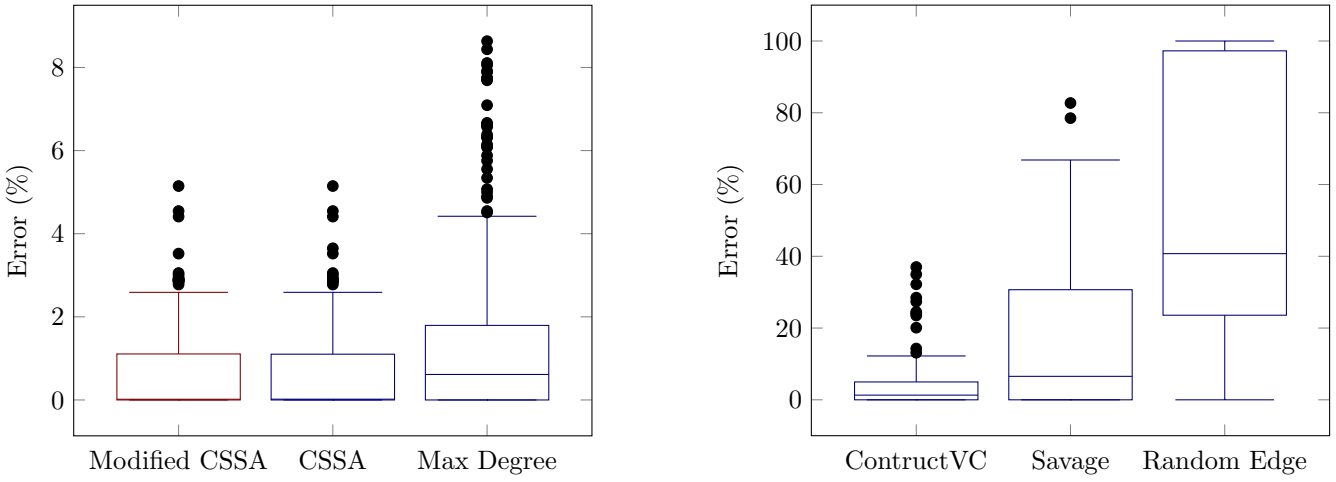

Figure 2: Comparison of Heuristics. They are sorted from the best to the worse one.

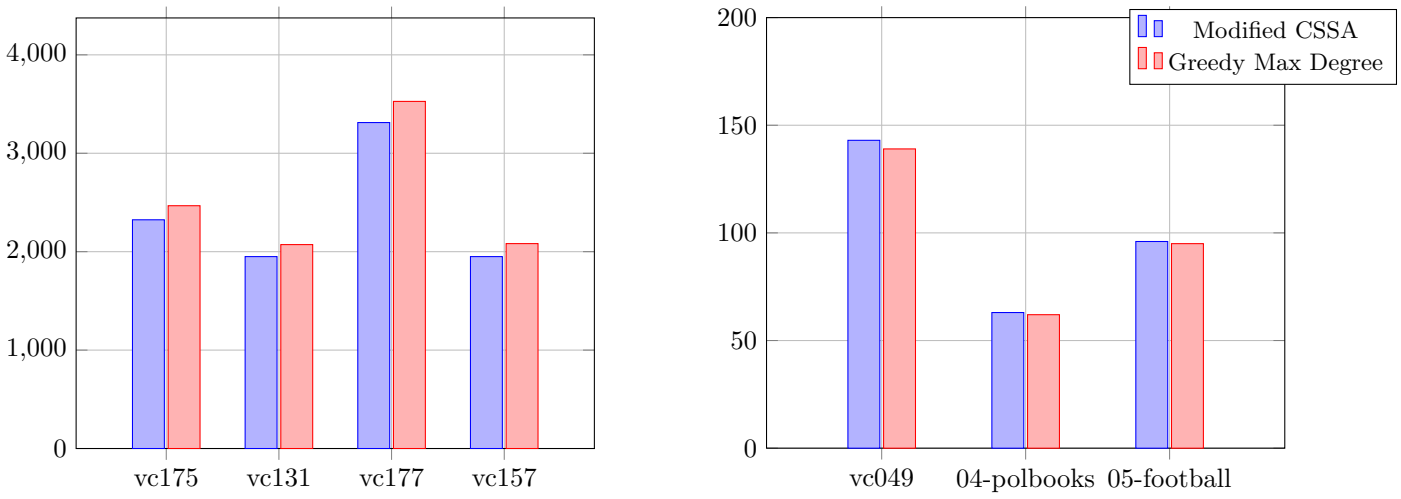

Figure 3: Comparison of Modified CSSA and Greedy Maximum Degree heuristics on specific graphs.