Algorithm Engineering – Exercise 4

Team 2: Denis Koshelev, Julian Fechner, Julio Cesar Perez Duran

Presentation, January 11

Overview

In this presentation we will talk about:

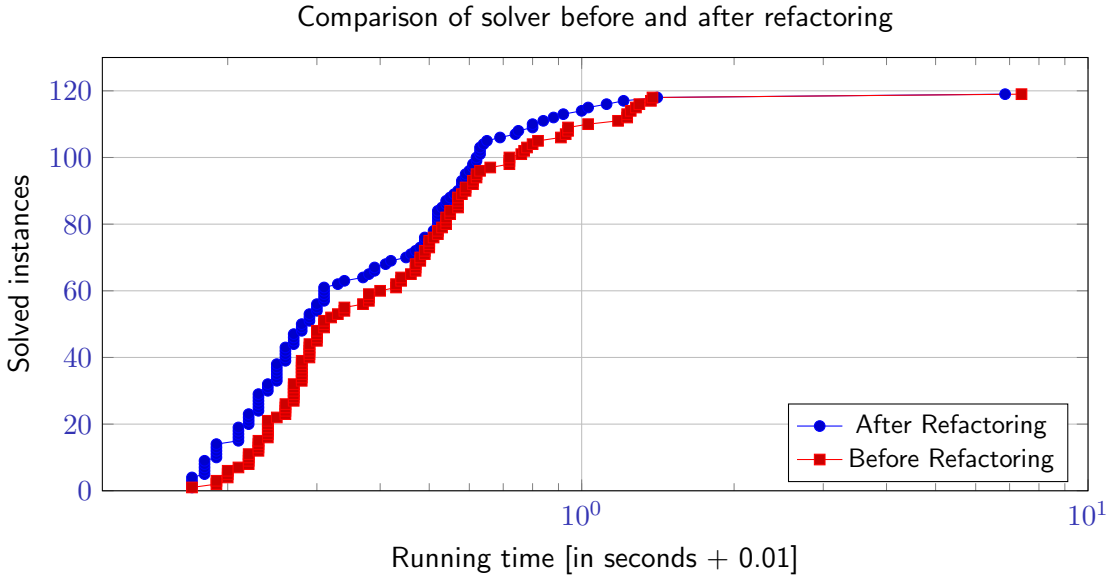
1. Overview
2. Refactoring
3. ILP & SAT Solvers
4. Implemented Heuristics:
   - Max-Degree Greedy Heuristic
   - Random-Edge Heuristic
   - Savage Heuristic
   - ConstructVC Heuristic
   - (Modified) CSSA Heuristic
5. Analysis of best heuristic

Refactoring

We decided to do some refactoring:

- Re-insert custom data-structure (degree-map)
  - Small speed-up (compared to before)
  - Max-Degree vertex needed for Heuristics
- Update bipartite graph while branching
  - Avoid creating new graph in each iteration
  - Speed-up for: LP-bound and LP-Reduction
- Apply reduction rules only in each n-th iteration
  - Some rules very "expensive"
  - Avoid executing "expensive" rules in each iteration

Refactoring



SAT & ILP Solvers

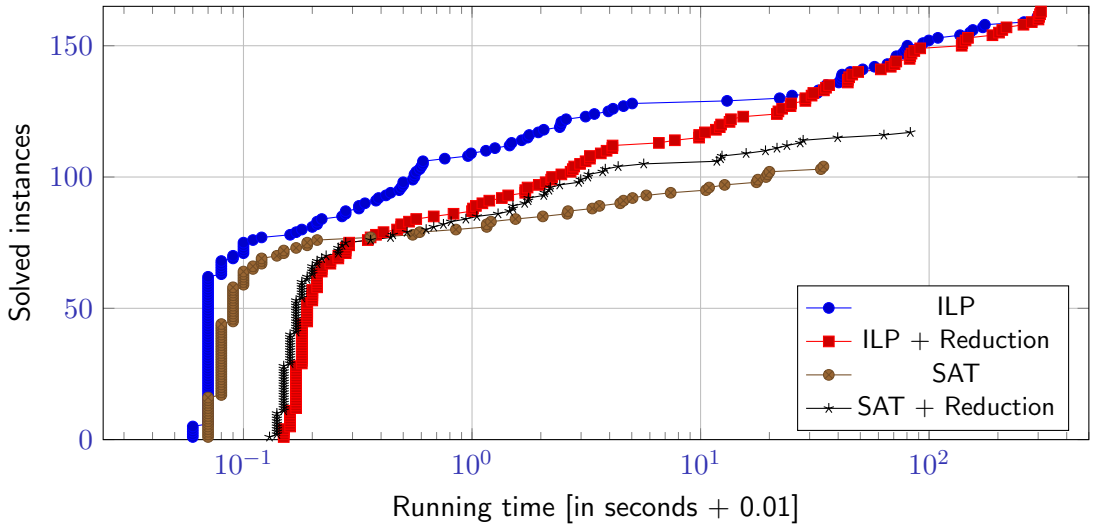
Implemented SAT Solver:

- Weighted SAT
- Python Module: python-sat

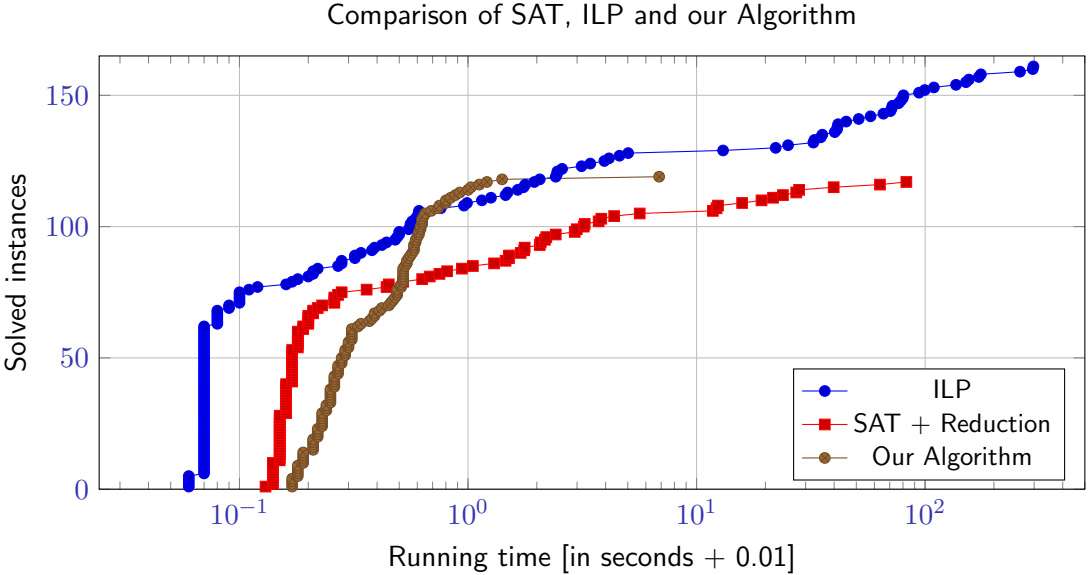
Implemented ILP Solver:

- CPLEX ILP Solver
- Python Module: cplex

SAT vs ILP


Comparison of SAT and ILP solvers
Solved instances
150
100
50
0
10^{-1}
10^{0}
10^{1}
10^{2}
Running time [in seconds + 0.01]
ILP
ILP + Reduction
SAT
SAT + Reduction

SAT vs ILP vs Algorithm



Implemented Heuristics

From the lecture:

- Max-Degree Greedy Heuristic

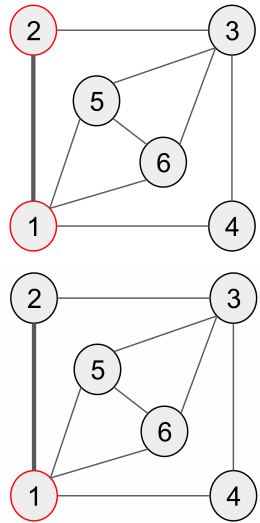
New heuristics:

- Random-Edge Heuristic
- Savage Heuristic
- ConstructVC Heuristic
- (Modified) CSSA Heuristic

Underperforming Heuristics

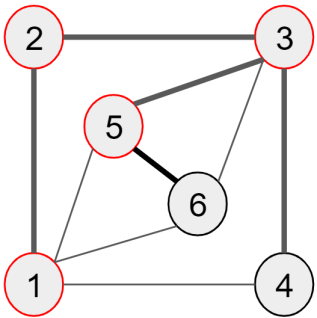
- **Random-Edge Heuristic**
  Iterate through all edges and take both vertices in the solution, if edge is uncovered.

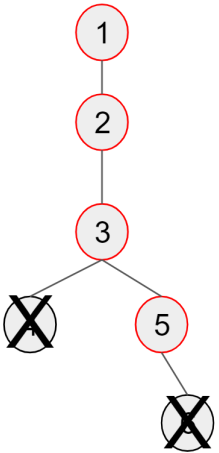
- **ConstructVC Heuristic** [1]
  Iterate through all edges and take vertex with a greater degree in the solution, if edge is uncovered.



Underperforming Heuristics

- **Savage Heuristic** [3]
  Construct a DFS spanning tree and take all non-leaves in the solution.



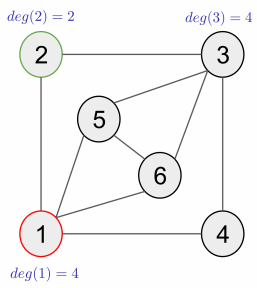


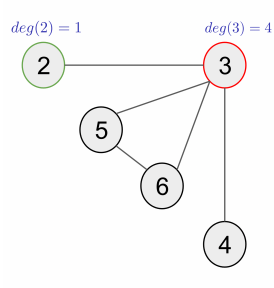
Best heuristic: (Modified) CSSA

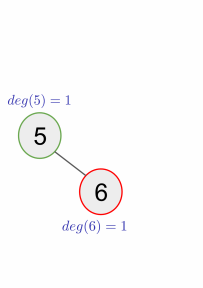
- **Clever Steady Strategy Algorithm** [2]
  Find a vertex with the lowest $deg(v)$ and take in the solution its neighbor with the **lowest** $deg(v)$.

- **Modified CSSA**

Find a vertex with the lowest $deg(v)$ and take in the solution its neighbor with the **highest** $deg(v)$.

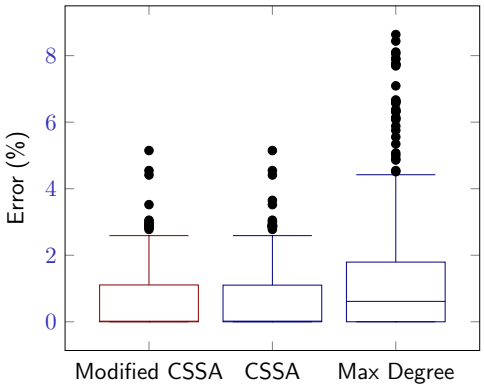


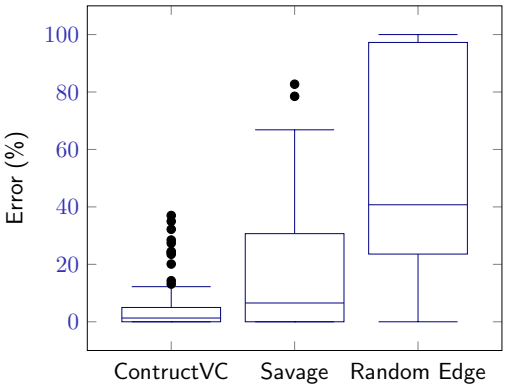




Comparison of Heuristics

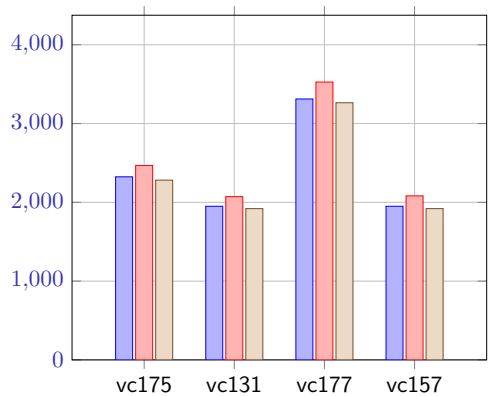
Comparison of heuristics. They are sorted from the best to the worse one.

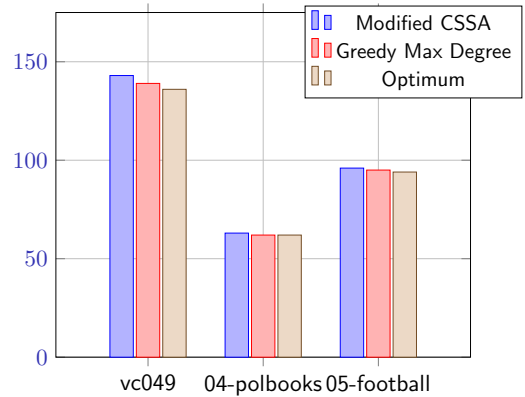




Modified CCSA vs Greedy Maximum Degree

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





1. Cai, S. 2015. Balance between Complexity and Quality: Local Search for Minimum Vertex Cover in Massive Graphs Balance between complexity and quality: Local search for minimum vertex cover in massive graphs. (747–753). AAAI Press.
2. Fayaz, M. Arshad, S. 2015. Clever Steady Strategy Algorithm: A Simple and Efficient Approximation Algorithm for Minimum Vertex Cover Problem Clever steady strategy algorithm: A simple and efficient approximation algorithm for minimum vertex cover problem. (277-282). 10.1109/FIT.2015.55
3. Savage, C. . Depth-first search and the vertex cover problem Depth-first search and the vertex cover problem. Information Processing Letters233-235. https://www.sciencedirect.com/science/article/pii/0020019082900229

Thank you for your attention!
Questions or Feedback?