Algorithm Engineering – Exercise 2

Denis Koshelev, Julian Fechner, Julio Cesar Perez Duran

Presentation for Exercise 2, November 23th

Overview

In this presentation we will talk about:

1. Overview
2. Implemented features
   - Modified branching algorithm
   - Clique bound
   - Linear Programming bound
3. Data structures
4. Experiments
5. Performance evaluation

Implemented Features

In this exercise we've managed to implement following features:

- Modified branching algorithm
- Clique bound
- Linear Programming bound

Clique bound

Two different approaches:

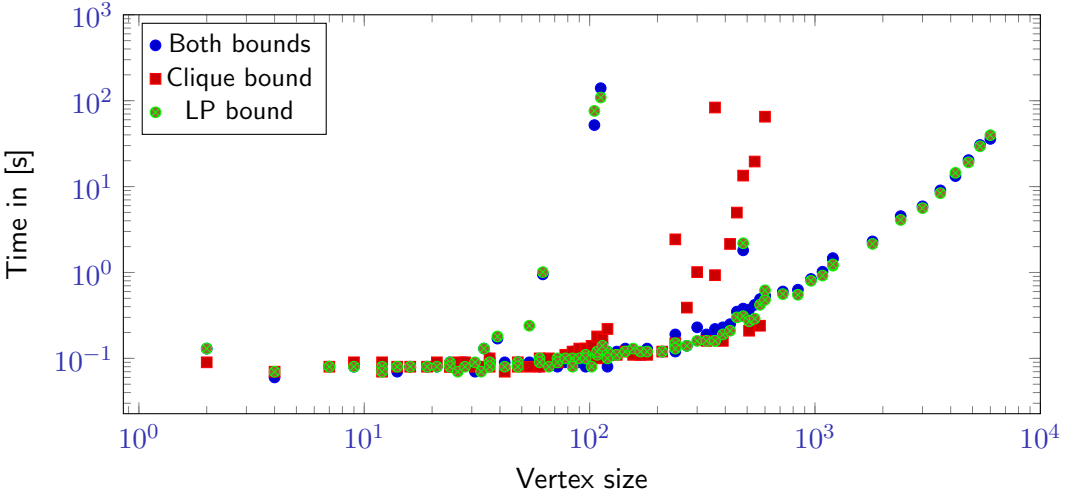
- Construct cliques one by one, taking the next node available with the highest degree
- Random shuffle the list of vertices and build cliques in that order x 10 and taking best result (lower amount of cliques)

Linear Programming bound

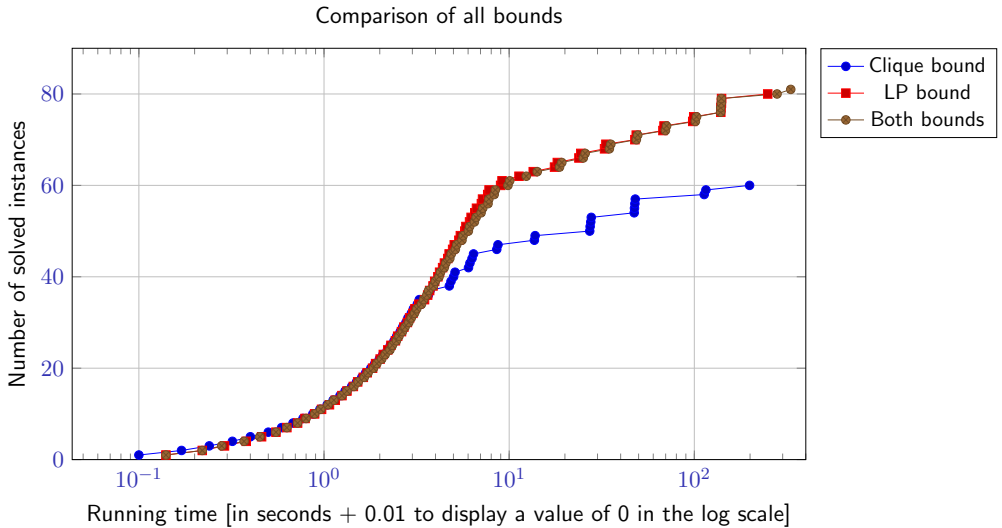
For computing a linear programming bound:

- Construct a bipartite graph from original one
- Run Hopcroft–Karp algorithm for maximum matching
- Divide result by two due to the König's theorem

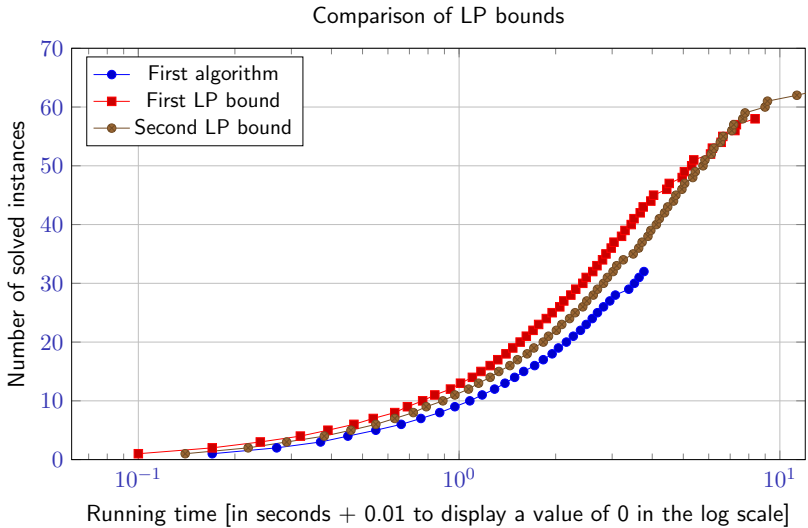
Comparing all bounds



Linear Programming bound

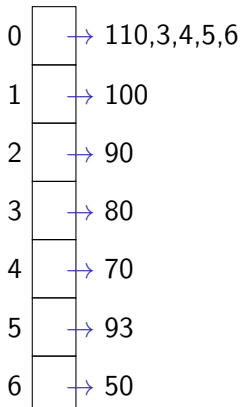


Linear Programming bound



Data Structure

Hashmap of hashsets of vertices according to degree



Pointer = 6

Experiments

We have conducted the following experiments for this assignment:

- Implementation of additional (naive) Bounds:
  - Complete Graphs (Kn)
  - Max-Subcomponents (disconnected Graphs)
  - Max-Node-Degree ≤ 2 (Circles and Paths)

Performance Evaluation

- Better Branching: $O(1.4565^k \cdot n^{O(1)})$
- LP Bound:
  - Generate Bipartite Graph: $O(V + E)$
  - Create Maximum Matching: $O(E \cdot \sqrt{V})$
- Clique Bound:
  - Approach 1: Max-Degree $O(V^2)$
  - Approach 2: Random $O(V^2)$

Thank you for your attentention!
Questions or Feedback?