HCPS42

Temirkhan Zimanov
Student at Higher School of Economics

1 Abstract

In this paper we describe a solver submitted to the heuristic track of PACE 2024. This solver was also submitted to the parametrized track with minor changes.

Digital Object Identifier: https://doi.org/10.5281/zenodo.12087453
GitHub Repository: https://github.com/HCPS42/PACE2024

2 Problem

We are given a bipartite graph $G = (U, V, E)$ as an input. We need to output a permutation π of vertices from V that minimizes

$$f(\pi) = |\{((u_1, v_1), (u_2, v_2)) \in E^2 \mid u_1 < u_2, \pi^{-1}(v_1) > \pi^{-1}(v_2)\}|.$$

3 Heuristic approximate solution

For $v \in V$, let

$$N(v) = \{u \in U \mid (u, v) \in E\}.$$

If $|V| > 15000$, then just order the vertices of V with respect to

$$\frac{\sum_{u \in N(v)} u}{|N(v)|}.$$

Otherwise, for any $i \neq j \ \in V$ let

$$c_{i,j} = |\{(u_1, u_2) \in N(i) \times N(j) \mid u_1 > u_2\}|.$$

Note that

$$f(\pi) = \sum_{\substack{\pi_i, \pi_j \in V \\ \pi_i < \pi_j}} c_{\pi_i, \pi_j}.$$

Then a trivial lower bound of $f(\pi)$ is

$$f(\pi) \geq L(G) = \sum_{\substack{i,j \in V \\ i<j}} \min(c_{i,j}, c_{j,i}).$$

Let us consider a directed graph $H = (V, S)$, where $(i, j) \in V^2, i \neq j$ belongs to S if and only if $c_{i,j} < c_{j,i}$. That is, we would prefer i to be before j. Now we can run Kosaraju's algorithm to find the strongly connected components of H and their topological order:

$$(V_1, V_2, \ldots, V_k).$$

Clearly, it is optimal to leave the relative order of the strongly connected components V_i as is and only to work with individual components from now on.

We can try to improve a component V_i in the following way. First, sort the vertices in V_i according to

$$\frac{\sum_{u \in N(v)} u}{|N(v)|}.$$

Next, split it into blocks of 40 consecutive vertices (there might be less than 40 vertices in the last block). Last, apply a divide-and-conquer algorithm that splits the current block into two halves, processes each half, and then orders them to minimize $f(\pi)$. If the current block is of size not bigger than 10, then iterate over all possible permutations of the vertices in the block.

Finally, return the improved components in the order obtained by the Kosaraju's algorithm.