PACE Solver Description: A Simplified Threshold Accepting Approach for the Cluster Editing Problem*

Martin Josef Geiger
Logistics Management Department, University of the Federal Armed Forces Hamburg, Germany

Abstract

We present a simple heuristic for the Cluster Editing Problem as presented in the Parameterized Algorithms and Computational Experiments (PACE) 2021. Our method makes use of a simple Threshold Accepting strategy and employs single neighborhood moves only.

Despite its simplicity, the results of the method are encouraging. However, and this has to be expected, the approach cannot ultimately win in a competitive setting such as PACE 2021. Nevertheless, some interesting insights can be derived from such a simple method, as this gives an idea of how good results can be by a comparable basic approach with a reasonable implementation effort.

2012 ACM Subject Classification Theory of computation → Randomized local search

Keywords and phrases Cluster Editing Problem, Threshold Accepting, Local Search

Digital Object Identifier 10.4230/LIPIcs.IPEC.2021.34

Supplementary Material *Software (Source Code)*: `https://doi.org/10.5281/zenodo.4891323`

1 Introduction

In the fall of 2020, the Parameterized Algorithms and Computational Experiments (PACE) competition invited to work on the Cluster Editing Problem. In this problem, an undirected, weighted graph $G = (V, E)$ is to be transformed into a cluster graph by a minimum number of edge modifications (additions and/ or deletions).

Following our contribution to the PACE 2018, and based on our experiences with simple, yet effective local search algorithms in other competitions (with a local search technique reduced to a single-operator search for the VeRoLog Solver Challenge 2019 as a prominent example), we decided to concentrate on what works best with little effort.

2 The actual algorithm

The method implemented for PACE 2021 is best characterized as a multi-start Threshold Accepting-type local search.

2.1 Preprocessing

We did experiment with some preprocessing techniques that allow for a reduction of the graph G. However, in our limited experiments, we did not find them to be overly beneficial, so for the actual submission, all of them have been removed.

* This is a brief description of one of the highest ranked solvers of PACE Challenge 2021. It has been made public for the benefit of the community and was selected based on the ranking. PACE encourages publication of work building on the ideas presented in this description in peer-reviewed venues.

2.2 Creating an initial solution

The initialization of the first alternative is straight-forward: Each node is placed in a separate cluster. Hence, $|V|$ clusters are initially created, and the objective function value of this initial solution is (always) $|E|$.

2.3 Improvements by Local Search

The only move implemented in our contribution is a *single-node-move* operator. This operator investigates the effect of removing a single node from its current clusters, and placing it either

- in a new, i. e., a previously empty cluster, or
- in a "neighboring" cluster.

The first sub-type is important as otherwise the number of clusters would monotonically decrease – and can only decrease – throughout the runs. Note that the definition of a neighboring cluster is based on the elements in E: A node might join another cluster if (and only if) there is an incident edge in E connecting it to a node of this cluster. Effectively, this sub-type can be implemented on the edges as given in G.

Moves are considered in random order.

2.4 Acceptance condition

Any move improving the current solution is accepted right away. Moreover, moves that do not change the current evaluation are accepted with $p = 0.5$. Furthermore, moves that are within a threshold T of the current best-know-solution are accepted with, again, $p = 0.5$. T is, during search, randomly changed every 1,000,000 moves by choosing it from $\{0, 1, 2\}$.

A value of $T = 0$ corresponds to a pure hillclimbing algorithm, while values of $\{1, 2\}$ allow the search to escape local optima. In our experience, and based on the limited work put into this project, this mechanism contributes to the quality of the identified solutions. It has also been observed that accepting moves that do not change the objective function value (but merely maintain it) are beneficial in terms of diversifying the search.

2.5 Restarts

Once the algorithm is unable to improve the current best-known solution for 50 percent of its running time, a restart is triggered. See subsection 2.2 on the initial solution.

The best solution identified through all runs is kept and reported once the termination criterion is reached.

3 Source-code

The source-code of our contribution has been published under the Creative Commons Attribution 4.0 International Public License and made available under `https://doi.org/10.5281/zenodo.4891323`.