Zygosity - Solver for the PACE 2023 Challenge on Twin-Width - Heuristic Track

Emmanuel Arrighi
University of Trier, Germany

Pål Grønås Drange
University of Bergen, Norway

Kenneth Langedal
University of Bergen, Norway

Farhad Vadiee
University of Bergen, Norway

Martin Vatshelle
University of Bergen, Norway

Petra Wolf
University of Bergen, Norway

Abstract

Submission for the PACE challenge 2023 - heuristic track. The heuristic starts by looking for and contracting twins. It then contracts the rest of the graph in a randomized greedy manner.

2012 ACM Subject Classification Theory of computation → Random search heuristics

Keywords and phrases PACE, twin-width, randomized greedy algorithm

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

Supplementary Material *Solver for the heuristic track*:
`https://bitbucket.org/Petra2334/pace2023-code/src/master/Zygosity_Heuristic/`

1 Description

Our randomized greedy heuristic considers multiple random contractions at each step and selects the best one. The exact number of contractions to consider is determined based on the remaining time and the time required to check each contraction. We employ two measures to evaluate the quality of a contraction.

- **Red degree:** This is the primary aspect we aim to minimize. Thus, for a potential contraction, we identify the highest red degree among the vertices whose red degree would increase as a result of this contraction. In the case where two neighboring vertices are being contracted, if the edge between them is red, the red degree of the new vertex will only be considered increased if it is strictly greater than the red degrees of both original vertices.
- **Intersection size:** The vertices in the intersection of the two neighborhoods will have their degree reduced by one. Additionally, the total number of edges in the graph will decrease by the intersection size. For these reasons, having a larger intersection size is considered better. If there is an edge between the two contracted vertices, it will add one to the intersection size.

We prioritize the smallest red degree and use intersection size as a tiebreaker. The only exception to this rule is when both contractions being compared have a red degree lower

than the current width of the sequence generated so far. In such cases, only the intersection size is taken into account.

Instead of randomly picking a pair of vertices, we randomly pick the first vertex and then do a random walk from there. The random walk performs either one or two steps. We initially flipped a coin to decide between one and two. However, we later discovered that favoring one over the other improves the results in certain instances. In the final version, we favor distance one walks on sparser graphs and distance two on dense ones.