BreakingTheCycle: A Heuristic Solver for Calculating the Minimum Directed Feedback Vertex Set in the 2022 PACE Challenge

Jonathan Gutermuth
Goethe University Frankfurt, Germany

Lars Huth
Goethe University Frankfurt, Germany

Marius Lotz
Goethe University Frankfurt, Germany

Johannes Meintrup
THM, University of Applied Sciences Mittelhessen, Giessen, Germany

Timo Mertin
Goethe University Frankfurt, Germany

Manuel Penschuck
Goethe University Frankfurt, Germany

Lukas Schwarz
Goethe University Frankfurt, Germany

Hung Tran
Goethe University Frankfurt, Germany

Abstract

The goal of the directed feedback vertex set problem is to, given a directed graph, find the smallest set of nodes to be removed to render the input acyclic. In this report, we give a short description of our solver *BreakingTheCycle*. This is the report for our submission to the heuristic track of the 2022 PACE Challenge.

2012 ACM Subject Classification Mathematics of Computation → Graph algorithms

Keywords and phrases PACE Challenge 2022, Directed Feedback Vertex Set, Heuristic

Supplementary Material The project's source code can be found on GitHub `https://github.com/goethe-tcs/breaking-the-cycle` and Zendono `https://doi.org/10.5281/zenodo.6602946`.

Funding This work was partially supported by the Deutsche Forschungsgemeinschaft (DFG) under grants ME 2088/4-2 (SPP 1736 — Algorithms for BIG DATA) and ME 2088/5-1(FOR 2975 — Algorithms, Dynamics, and Information Flow in Networks).
Johannes Meintrup: Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 379157101.

Acknowledgements We would like to thank Holger Dell, Anselm Haak, and Frank Kammer for valuable discussions, the PACE committees for organizing the challenge and curating the benchmark data, as well as the Center for Scientific Computing at Goethe University Frankfurt for making their high-performance computing facilities available.

1 Preliminaries

Let $G = (V, E)$ be a directed graph. Denote the open in- and out-neighborhoods of node $u \in V$ as $N_{\text{in}}(u) = \{v \mid (v, u) \in E\}$ and $N_{\text{out}}(u) = \{v \mid (u, v) \in E\}$, respectively, and let $N_{\text{und}}(u) = N_{\text{in}}(u) \cap N_{\text{out}}(u)$. The directed feedback vertex set problem (DFVS) asks for a minimum set of nodes $V' \subseteq V$, such that the graph resulting from removing all nodes of V' from G is acyclic.

2 Solver Summary

Given a graph G, we begin by applying our data reduction rules. This may decrease the problem size by removing redundant nodes and already adding required nodes to the DFVS and possibly splits the input into several strongly connected components (SCCs) which can be processed independently. We then spend 1 minute to solve these SCCs exactly as described in [4]. If unsolved SCCs remain after the timeout, a makeshift heuristic DFVS is calculated by applying the *Weakest Link* algorithm Section 5. The *Weakest Link* runs are also subject to a time budget. This heuristic DFVS is then refined using a *Simulated Annealing* algorithm Section 6. Since *Simulated Annealing* calculates a DFVS right after its initialization, it will almost always find a non-trivial DFVS for every SCC.

Should time run out prior to the application of *Weakest Link* and *SimulatedAnnealing*, we establish a trivial DFVS containing all but one node of yet unprocessed SCCs.

3 Reduction Rules

The following data reduction rules are used in the heuristic solver we submitted. Observe that DFVS on undirected instances degenerates into the well studied Vertex Cover problem. Thus, we can adopt rules for the Vertex Cover and Independent Set problems for DFVS.

3.1 Existing Kernelization Rules

We use the following kernelization rules previously described in literature:

- Rules 1 to 5 due to Fleischer et al. [2]
- *PIE Rule* due to Lin and Jou [5]
- *Crown Rule* which is inspired by Abu-Khzam et al. [1]

3.2 Novel Kernelization Rules

Our solver also includes the following novel rules:

DiClique Rule

For every node $u \in G$, the DiClique Rule checks whether $N_{\text{und}}[u]$ is a clique. If additionally u is not part of a cycle in $G \setminus N_{\text{und}}(u)$, we add $N_{\text{und}}(u)$ to the DFVS and remove $N_{\text{und}}[u]$ from G. This rule is a variation of the CORE Rule from Lin and Jou [5].

DOMN Rule

If the contraction $G \times v$ of a node $v \in G$ does not add any new edges to the graph G, we remove node v from G.

Redundant Cycles Rule

This rule searches for small cycles of length two (undirected edges), three, and four. Let $C \subseteq V$ be such a cycle. If there exists an edge outside of C that is not part of any cycle every time any of C's nodes is deleted, we can delete that edge safely since we know that we have to delete at least one of C's nodes eventually.

C4 rule

We search four pairwise disjoint nodes u, v, w, x such that there are undirected edges $\{u, v\}$, $\{u, w\}$, $\{v, x\}$, and $\{w, x\}$. In such a structure, we have to delete at least $A = \{u, x\}$ or $B = \{v, w\}$. If –wlog– the nodes A are not in any cycle in $G \setminus B$, we add B to the DFVS and remove $A \cup B$.

4 Branch and Bound

The branch-and-bound algorithm used for the calculation of an optimal solution has been taken from our exact track submission [4].

5 Weakest Link

While G is not acyclic, we remove what we call the *weakest link* with each iteration. Firstly, we exhaustively remove all nodes u with $N_{\text{in}}(u) = \emptyset$ or $N_{\text{out}}(u) = \emptyset$. All remaining nodes are part of at least one cycle. Let $d_{\min} = \min_{v \in V}(|N_{\text{in}}(v)|, |N_{\text{out}}(v)|)$ be the smallest degree in the graph. We find a node $u \in \{v \mid v \in V \colon \min(|N_{\text{in}}(v)|, |N_{\text{out}}(v)|) = d_{\min}\}$ with largest total degree $|N_{\text{in}}(u)| + |N_{\text{out}}(u)|$.

The *weakest link* is now the set of all edges between u and either its in- or out-neighborhood, depending on which one is smaller. All the neighbors in said neighborhood are then added to the DFVS and removed. Since this makes u either a sink or a source, u can be removed as well before checking for a cycle and going into the next iteration.

6 Simulated Annealing

We apply simulated annealing as described in [3], which tries to reach a minimum DFVS starting from the heuristic DFVS calculated using the weakest-link procedure.

Simulated Annealing is an iterative algorithm, therefore it has two following beneficial characteristics. First, it provides a DFVS right after initialization, which may subsequently decrease in size as iteration steps continue. Therefore, is it easy to stop the algorithm at any time and still obtain a feasible and best-known solution so far. Secondly, *Simulated Annealing* can start and try to improve any given precalculated DFVS. This can save time, if the precalculation is faster than Simulated Annealing for the first approximate solution.

Simulated Annealing is a metaheuristic, which can leave a local optimum, which gives the chance to find another better local optimum. The probability to leave a local optimum (increase the DFVS size) depends on a temperature factor. The temperature factor is a variable which decreases over runtime, thereby decreasing the probability to leave a local minimum.

References

1 Faisal N. Abu-Khzam, Rebecca L. Collins, Michael R. Fellows, Michael A. Langston, W. Henry Suters, and Christopher T. Symons. Kernelization algorithms for the vertex cover problem: Theory and experiments. In Lars Arge, Giuseppe F. Italiano, and Robert Sedgewick, editors, *Proceedings of the Sixth Workshop on Algorithm Engineering and Experiments and the First Workshop on Analytic Algorithmics and Combinatorics, New Orleans, LA, USA, January 10, 2004*, pages 62–69. SIAM, 2004.

2 Rudolf Fleischer, Xi Wu, and Liwei Yuan. Experimental study of FPT algorithms for the directed feedback vertex set problem. In Amos Fiat and Peter Sanders, editors, *Algorithms - ESA 2009, 17th Annual European Symposium, Copenhagen, Denmark, September 7-9, 2009. Proceedings*, volume 5757 of *Lecture Notes in Computer Science*, pages 611–622. Springer, 2009. doi:10.1007/978-3-642-04128-0_55.

3 Philippe Galinier, Eunice Adjarath Lemamou, and Mohamed Wassim Bouzidi. Applying local search to the feedback vertex set problem. *J. Heuristics*, 19(5):797–818, 2013. doi:10.1007/s10732-013-9224-z.

4 Jonathan Guthermuth, Lars Huth, Marius Lotz, Johannes Meintrup, Timo Mertin, Manuel Penschuck, and Hung Tran. Breakingthecycle: An exact solver for calculating the minimum directed feedback vertex set in the 2022 pace challenge. Technical report, Goethe University Frankfurt and University of Applied Sciences Mittelhessen, 2022.

5 Hen-Ming Lin and Jing-Yang Jou. Computing minimum feedback vertex sets by contraction operations and its applications on CAD. In *Proceedings of the IEEE International Conference On Computer Design, VLSI in Computers and Processors, ICCD '99, Austin, Texas, USA, October 10-13, 1999*, page 364. IEEE Computer Society, 1999. doi:10.1109/ICCD.1999.808567.