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 least amount of nodes required to be removed which make the graph 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 for valuable discussions, 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 to decrease the problem size. This decreases the problem size by removing redundant nodes and already adding required nodes to the DFVS as well as then removing them. With our G likely split into multiple strongly connected components (SCCs), the solver gets a total of 1 minute to iterate over all of them and calculate exact solutions to the problem using our *BranchAndBound* described in [5]. If there are unsolved SCCs left when the minute has run out, a makeshift heuristic DFVS is calculated by applying the *WeakestLink* algorithm. This heuristic DFVS is then refined using a *SimulatedAnnealing* algorithm.

Should time run out prior to the application of *WeakestLink* and *SimulatedAnnealing*, all the the remaining nodes of all remaining SCCs get packed into the makeshift heuristic DFVS and used as output.

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 6 due to Fleischer et al.[3]
- *PIE Rule* due to Lin and Jou [8]
- *Unconfined vertices* due to from Kneis et al.[7]
- *Crown Rule* which is inspired by Abu-Khzam et al.[1]
- *Vertex folding* due to Chen et al.[2]
- *Twin Rule* due to Xiao and Nagamochi [9] as described by Hespe et al.[6]
- *Desk Rule* and *Funnel Rule* due to Hespe et al.[6]

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 [8].

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 [5]. It maintains lower- and upper-bounds to prune the search tree.

5 Weakest Link

While G is not acyclic, we remove what we call the *weakest link* with each iteration. Firstly, we remove all nodes with *in_degree* of 0 or *out_degree* of 0, is applied until no such nodes are left. This leaves only SCCs, meaning all nodes are part of at least one cycle. Next, we find a node $u \in V$ such that it is one of the possible results of $\operatorname{argmin}_{v \in V(G)}(|N_{\text{in}}(u)|, |N_{\text{out}}(v)|)$ and its total degree $|N_{\text{in}}(u)| + |N_{\text{out}}(u)|$ is maximal among all possible argmax outputs.

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 [4], using which we try to reach a minimum DFVS starting from the heuristic DFVS calculated using the weakest-link procedure.

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. Jianer Chen, Iyad A. Kanj, and Weijia Jia. Vertex cover: Further observations and further improvements. *J. Algorithms*, 41(2):280–301, 2001.
3. 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.
4. 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.
5. 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.
6 Demian Hespe, Sebastian Lamm, Christian Schulz, and Darren Strash. Wegotyoucovered: The winning solver from the PACE 2019 challenge, vertex cover track. In *CSC*, pages 1–11. SIAM, 2020.
7 Joachim Kneis, Alexander Langer, and Peter Rossmanith. A fine-grained analysis of a simple independent set algorithm. In Ravi Kannan and K. Narayan Kumar, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2009, December 15-17, 2009, IIT Kanpur, India*, volume 4 of *LIPIcs*, pages 287–298. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2009. doi:10.4230/LIPIcs.FSTTCS.2009.2326.
8 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.
9 Mingyu Xiao and Hiroshi Nagamochi. Confining sets and avoiding bottleneck cases: A simple maximum independent set algorithm in degree-3 graphs. *Theor. Comput. Sci.*, 469:92–104, 2013.