Mount Doom: An Exact Solver for Directed Feedback Vertex Set

Sebastian Angrick
Hasso Plattner Institute, University of Potsdam

Ben Bals
Hasso Plattner Institute, University of Potsdam

Katrin Casel
Hasso Plattner Institute, University of Potsdam

Sarel Cohen
The Academic College of Tel Aviv-Yaffo, Israel

Tobias Friedrich
Hasso Plattner Institute, University of Potsdam

Niko Hastrich
Hasso Plattner Institute, University of Potsdam

Theresa Hradilak
Hasso Plattner Institute, University of Potsdam

Davis Issac
Hasso Plattner Institute, University of Potsdam

Otto Kißig
Hasso Plattner Institute, University of Potsdam

Jonas Schmidt
Hasso Plattner Institute, University of Potsdam

Leo Wendt
Hasso Plattner Institute, University of Potsdam

Abstract

In this document we describe the techniques we used and implemented for our submission to the Parameterized Algorithms and Computational Experiments Challenge (PACE) 2022. The given problem is *Directed Feedback Vertex Set* (DFVS), where you are given a directed graph $G = (V, E)$ and want to find a minimum $S \subseteq V$ such that $G - S$ is acyclic. We approach this problem by first exhaustively applying a set of reduction rules. In order to find a minimum DFVS on this remaining instance, we create and solve a series of Vertex Cover instances.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms

Keywords and phrases directed feedback vertex set, vertex cover, reduction rules

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

Supplementary Material Source code (Software): `https://doi.org/10.5281/zenodo.6645235`
Source code (Software): `https://github.com/BenBals/mount-doom/tree/exact`

1 Preliminaries

Let $G = (V, E)$ be a directed graph. The Directed Feedback Vertex Set problem asks to find a minimum $S \subseteq V$, such that $G - S$ is acyclic.

Let $v, w \in V$. We define $N^+(v)$ as the outgoing neighbors of v and $N^-(v)$ as the incoming neighbors. We call an edge $vw \in E$ bidirectional if $wv \in E$ as well. Let PIE $\subseteq E$ be the set of all bidirectional edges and let $B \subseteq V$ be the set of all vertices only incident to bidirectional edges. We define the bidirectional neighbors $N(v)$ as those which are incident using bidirectional edges. Additionally, we call $D \subseteq V$ a diclique, if all $u \in D$ have $D \setminus \{u\} \subseteq N(u)$.

Finally, let $v \in V$ be given. Let $V' = V \setminus \{v\}$ and $E' = (E \cap (V' \times V')) \cup (N^-(v) \times N^+(v))$. We call $G' = (V', E')$ the graph obtained from G by short cutting v. In light of the DFVS, this is equivalent to adding the assumption $v \notin S$.

2 Reduction rules

We apply a number of reductions known from the literature and introduce one rule which to the best of our knowledge is new. The known rules can be found in [5] and we adopt their

nomenclature. Additionally, we adapt some reduction rules, that have already been used for the Vertex Cover problem. Before running the reduction rules, all nodes with self loops are collected into the solution and removed from the graph. Isolated nodes are removed as well.

PIE. Recall that PIE is the set of bidirectional edges. Now consider any edge uv between different strongly connected components in $G - \text{PIE}$. Any cycle using this edge must therefore use at least one bidirectional edge, which must be covered anyways, so we can safely delete uv.

DOME. We call an edge $ab \in E - PIE$ dominated if all outgoing neighbors of b are also outgoing neighbors of a or if all incoming neighbors of a are also incoming neighbors of b. It is well known [2] that such a dominated edge can safely be deleted.

Improved CORE. A vertex a is a core of a diclique if the graph induced by a and its neighbors is a diclique. Traditionally, one now deletes $N(a)$ from G since if S' is optimal for $G - N(v)$ then $S' \cup N(v)$ is optimal for G [5]. We proceed differently and shortcut the node a if $N^+(a)$ or $N^-(a)$ are dicliques. While this extension is easy to prove, it is, to the best of our knowledge, novel.

SHORTONE. Consider any node v which has exactly one incoming edge uv and one outgoing edge vw in $G - PIE$ such that any bidirectional neighbor of v is also a bidirectional neighbor of at least one out of u or w. In that case we remove the edges uv and vw and add uw. Call the reduced graph G'. For correctness, take any solution S in G. If $v \notin S$, then S is also a solution for G'. If $v \in S$, assume S is not a solution in G'. Then $u, w \notin S$ since any cycle introduced by the reduction must use uw. Since all bidirectional neighbors of v are also bidirectional neighbors of u or w, those must all be in S. Thus we can simply replace v by u (or w) and obtain a solution of the same size. Also, any solution in G' is always a solution in G.

2.1 Vertex Cover Reductions.

Note that if we have a bidirectional edge between a and b, we have to take at least one of a and b into the DFVS. Therefore we can regard $G[PIE]$ as a vertex cover subproblem. Any DFVS for G must necessarily be a vertex cover for that subgraph. For that reason some vertex cover reduction rules [3] translate well to DFVS.

VC-DOME. Recall, that we denote by $B \subseteq V$ the set of vertices only incident to bidirectional edges. Consider $v \in B$, with $u \in N(b)$ with $N(v) \setminus \{u\} \subseteq N(u)$. Then, we add u to the solution and consider $G - u$.

Degree 2 Fold. Consider a degree-two vertex $v \in B$ with $N(v) = \{u, w\}$ and $u \in B$. Observe that there is an optimal DFVS D^* with either $D^* \cap \{u, v, w\} = \{v\}$ or $D^* \cap \{u, v, w\} = \{v, w\}$. To reduce the graph, we add a new vertex t to G and connect t in such a way, that $N^+(t) = N^+(w) \cup N(u)$ and $N^-(t) = N^-(w) \cup N(u)$ and we remove u, v and w from the graph. Then we solve the instance on the reduced graph to obtain the solution S. If $t \in S$, the solution to the original instance is $(S \setminus \{t\}) \cup \{u, w\}$. Otherwise, the solution to the original instance is $S \cup \{v\}$.

Funnel Fold. Consider a vertex $v \in B$ with $w \in N(v)$ such that $N(v) \setminus w$ is a diclique. Observe that there is an optimal DFVS D^* with either $D^* \cap \{v, w\} = \{v\}$ or $D^* \cap \{v, w\} = \{w\}$. To reduce the instance, we first add $C = N(u) \cap N(w)$ to the solution and remove these vertices from the graph. Now we add edges, such that each $x \in N(v) \setminus C$ is a bidirectional neighbor of every $y \in N(w) \setminus C$. Finally, we remove u, w from the graph. Then we solve the instance on the reduced graph to obtain the solution S. If $N(v) \setminus \{w\} \subseteq S$, the solution to the original instance is $S \cup \{w\}$. Otherwise, the solution to the original instance is $S \cup \{v\}$.

3 Reduction to Vertex Cover

After applying a set of reduction rules exhaustively we solve the remaining instance by repeatedly solving vertex cover instances. Initially, we choose the reduction rules Improved CORE and PIE. If this doesn't solve the instance within a few seconds, we proceed by using all reduction rules listed above.

First note that if the remaining graph contains only bidirectional edges, we can easily reduce the DFVS instance to a vertex cover instance by turning bidirectional edges into undirected edges. Initially, we find an optimal vertex cover S in G[PIE]. If S is a DFVS, S must be optimal. Otherwise, we find a set of vertex-disjoint cycles C in $G - S - PIE$ using a DFS. All cycles in C are not covered by S, so we add a gadget to each cycle to ensure, that in the modified graph, there is an optimal vertex cover, which includes a $v \in S$. Finally, we iterate on the modified graph until the vertex cover is also a DFVS. Note, that this can happen multiple times since our choice of C does not guarantee that all cycles in G are covered.

Let $G = (V, E)$ be an undirected graph and let $S \subseteq V$. Our goal is to find the minimum vertex cover in G that also contains a vertex in S. To achieve this, we add a clique of size $|S|$ to G and connect it one-to-one with S. We call the modified graph G'. Consider any optimal vertex cover C in G'. Then, C contains at least $|S| - 1$ vertices in the new clique. Also, C must cover all edges between V and the clique, so it must contain at least one vertex in S or all vertices in the clique. If C contains all vertices in the clique, we exchange one of these vertices for a vertex in S and obtain an optimal vertex cover C' in G' with $C' \cap S \neq \emptyset$. Thus, $C' \cap V$ is an optimal vertex cover of G that also contains a vertex of S.

Note that the exactness of our solver only relies on the optimality of the final vertex cover. Therefore, when the exact solver takes too long, we switch to a heuristic solver and verify our solution with an exact solver once we have covered all cycles in G. If the heuristic vertex cover was not optimal, we iterate further. Surprisingly, this did not occur on any public instance.

To solve a vertex cover instance, we initially reduce it using the kernelization procedure, implemented by the winning solver of the 2019 PACE challenge [4]. When solving this instance heuristically, we use a local-search solver [1] on this kernel. To solve the kernel exactly, we use a branch-and-reduce solver [6], which we augment by implementing better upper bounds using the aforementioned local-search solver.

References

1 Shaowei Cai, Kaile Su, Chuan Luo, and Abdul Sattar. Numvc: An efficient local search algorithm for minimum vertex cover. *J. Artif. Int. Res.*, 46(1):687–716, jan 2013. URL: https://dl.acm.org/doi/10.5555/2512538.2512555.

2 Reinhard Diestel. Graph theory 3rd ed. *Graduate texts in mathematics*, 173, 2005.

3 Fedor V. Fomin, Fabrizio Grandoni, and Dieter Kratsch. A measure & conquer approach for the analysis of exact algorithms. *J. ACM*, 56(5), aug 2009. doi:10.1145/1552285.1552286.

4 Demian Hespe, Sebastian Lamm, Christian Schulz, and Darren Strash. Wegotyoucovered: The winning solver from the pace 2019 challenge, vertex cover track. In *CSC*, 2020.

5 Mile Lemaic. *Markov-Chain-Based Heuristics for the Feedback Vertex Set Problem for Digraphs*. PhD thesis, Universität zu Köln, 2008. URL: https://kups.ub.uni-koeln.de/2547/1/Dissertation.pdf.

6 Rick Plachetta and Alexander van der Grinten. *SAT-and-Reduce for Vertex Cover: Accelerating Branch-and-Reduce by SAT Solving*, pages 169–180. doi:10.1137/1.9781611976472.13.