Orodruin: A Heuristic 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

This document describes the techniques we used and implemented for our submission 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. Our approach first generates an initial greedy solution. This solution is then checked for minimality under exclusion of a single node and exchange of two solution nodes for one new node.

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.6630460`
Source code (Software): `https://github.com/BenBals/mount-doom/tree/heuristic`

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 $\mathrm{PIE} \subseteq E$ 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 two reduction rules known from literature. These rules can be found in [4] and we adopt their nomenclature.

42nd Conference on Very Important Topics (CVIT 2016).
Editors: John Q. Open and Joan R. Access; Article No. 23; pp. 23:1–23:3
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

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.

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 [4]. 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.

3 Solver Description

After exhaustively applying the reductions we described in Section 2, we produce two solutions, one by a greedy procedure and another by a reduction to vertex cover. The better solution of both approaches is minimized further by applying 2-1 swaps until the timelimit is hit.

3.1 Greedy Routine

For the initial solution, we start with an empty set and greedily add to it the vertex of highest degree until we obtain a feasible DFVS. Checking for feasibility here means searching for a cycle in the graph. To speedup this procedure, we only search for a new cycle, once the old one is covered. Additionally, after a fixed number of nodes are taken into the solution, we reapply all reduction rules. After the initial solution is generated, we remove nodes that do not reintroduce a cycle when added back to the graph. This ensures that we create an inclusion-minimal solution.

3.2 Reduction to Vertex Cover

First note that if a 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 a vertex cover S in $G[\text{PIE}]$ and check, whether S is a DFVS for G. If not, 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. When we hit an internal timelimit before finding a feasible DFVS, we apply our greedy approach described in Section 3.1 for the remaining graph.

Let $G = (V, E)$ be an undirected graph and $S \subseteq V$ be a cycle. 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 a optimal vertex cover of G that also contains a vertex of S.

To solve the vertex cover instance, we first use a kernelization procedure implemented by the winning solver of the 2019 PACE challenge [3]. Then, we use a local-search solver [1] on this kernel.

3.3 2-1 swaps

We apply another greedy approach to improve our current solution, named 2-1 swaps. As the name suggests, its goal is to find 2 vertices from a given DFVS-solution and replace them with a vertex to form a smaller solution. The idea uses the notion of skew separator from Chen et al. [2].

Consider a feasible and minimal solution set S to the DFVS problem, i.e. for all $v \in S$ the set $S - v$ is infeasible. Our goal is to find a triple of vertices (v, w, u) where $v, w \in S$ and $u \notin S$ such that $(S - \{v, w\}) \cup \{u\}$ is a feasible solution. First, for each vertex $v \in S$, we find all vertices $u \in G - S$, called candidates, such that $(S - \{v\}) \cup \{u\}$ remains feasible. Let the candidates of a given solution vertex v be C_v.

In order to find C_v, we split the vertex v into two vertices $v_{\text{out}}, v_{\text{in}}$ with out- or ingoing edges of v only. Note that there is no edge between v_{out} and v_{in} in either direction. The candidates are now separator vertices for all $v_{out} \rightsquigarrow v_{in}$ paths in $(G - S) \cup \{v_{out}, v_{in}\}$.

Note that in $G - S$, the vertices in C_v have a topological ordering and since there is at least one cycle that is containing C_v, the ordering is unique. In $G - C_v$, we then mark all vertices u not reachable from v, with the first candidate (in increasing topological order) c of C_v so that $c \rightsquigarrow u$ in $G - C_v \cup \{c\}$, meaning that all candidates $c' \preceq c$ in C_v cover all paths $v \rightsquigarrow u$.

Let $w \in S, w \neq v$ be another solution vertex and $c_w \in C_v$ be the node that marked w. We can then check if there exists a node $y \in C_w \cap C_v$ for which $y \preceq c_w$ in the topological ordering of C_v holds, meaning that all $v \rightsquigarrow w$ paths have the subpath $y \rightsquigarrow c_w$. Therefore, y prevents all paths from $v_{out} \rightsquigarrow v_{in}, w_{out} \rightsquigarrow w_{in}$, and $v_{out} \rightsquigarrow w_{in}$ in $G - S \cup \{v, w\}$, covering all cycles with v, w, or v and w. Thereby, $S - \{v, w\} \cup \{y\}$ is still a feasible solution.

Note that this procedure is conducted for both sides, i.e. the procedure will also take place for swapped roles of v and w.

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. Jianer Chen, Yang Liu, Songjian Lu, Barry O'Sullivan, and Igor Razgon. A fixed-parameter algorithm for the directed feedback vertex set problem. *J. ACM*, 55:21:1–21:19, 2008.
3. Demian Hespe, Sebastian Lamm, Christian Schulz, and Darren Strash. Wegotyoucovered: The winning solver from the pace 2019 challenge, vertex cover track. In *CSC*, 2020.
4. 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`.