A Heuristic Solver for the Directed Feedback Vertex Set Problem

Bachelor Thesis Presentation

Andrei Arhire

Lect. Dr. Paul Diac

June 29, 2022

Table of Contents

Description

1 Introduction: PACE Challenge

The Parameterized Algorithms and Computational Experiments Challenge (PACE) was conceived in the Fall of 2015 to deepen the relationship between parameterized algorithms and practice.

Topics from multivariate algorithms, exact algorithms, fine-grained complexity, and related fields are in scope.

Motivation

1 Introduction: PACE Challenge

PACE aims to:

- Bridge the divide between the theory of algorithm design and analysis and the practice of algorithm engineering
- Inspire new theoretical developments
- Investigate how far theoretical algorithms from parameterized complexity and related fields are competitive in practice
- Produce universally accessible libraries of implementations and repositories of benchmark instances
- Encourage the dissemination of these findings in scientific papers

Specifications

1 Introduction: PACE Challenge

- Participation is open to anyone
- Competitors develop a solver that will quickly produce good results
- The programs are evaluated on 200 instances generated by the committee
- Candidates write a short paper describing the techniques used
- The team with the best solver is invited to the Award ceremony at the International Symposium on Parameterized and Exact Computation (IPEC)

Table of Contents

2 Directed Feedback Vertex Set

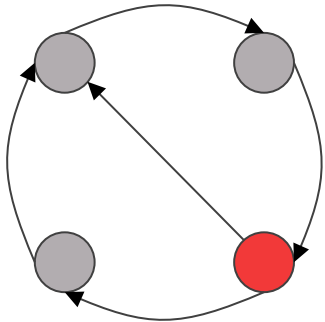
Description

2 Directed Feedback Vertex Set

The PACE challenge for this year is the Directed Feedback Vertex Set. The competition had one track for Exact algorithms and another for Heuristic algorithms.

In the Directed Feedback Vertex Set (DFVS) problem, the input is a directed graph D and an integer k.

The objective is to determine whether there exists a set of at most k vertices intersecting every directed cycle of D.



Description

2 Directed Feedback Vertex Set

Regarding the Minimum Directed Feedback Vertex Set:

- it is part of Karp's 21 $NP - complete$ problems
- the first fixed parameter tractable algorithm was proposed in 2008 [Igo08]
- the first kernelization of the problem was published in 2019 [S.20]
- it has applications deadlock recovery, Bayesian inference, and in control theory of gene regulatory networks

Table of Contents

Heuristic Solver

3 High-Level Description

The Heuristic solver can be structured in three stages:

- a solution is computed following heuristic selections that alternate with reduction operations.
- redundant nodes are removed from the previously found solution
- several local searches are performed starting from a subset of the best-known solution

Algorithm

3 High-Level Description

Algorithm 1 Stage I

Require: $G-$ a directed graph, $\alpha \in (0, 1]$
  $F \leftarrow \emptyset$
  $E \leftarrow getNumberOfEdges(G)$
  while $G \neq \emptyset$ **do**
    $G, F \leftarrow useLightReductions(G, F)$
    if $E \leq \alpha \cdot getNumberOfEdges(G)$ **then**
      $G, F \leftarrow useHeavyReductions(G, F)$;
      $E \leftarrow getNumberOfEdges(G)$
    end if
    if $G \neq \emptyset$ **then**
      $V \leftarrow selectVertex(G)$
      $G \leftarrow remove(G, V)$
      $F \leftarrow F \cup V$
    end if
  end while
  return F

Two processes are repeated until the graph becomes empty:

- reduction rules are applied
- a promising vertex is selected to be part of the DFVS and is removed

A vertex v is heuristically selected if it maximizes one of the functions:

- $(d^+_G(v) + d^\pm_G(v)) \cdot (d^-_G(v) + d^\pm_G(v))$
- $d^\pm_G(v) \cdot \infty + d^-_G(v) \cdot d^+_G(v)$

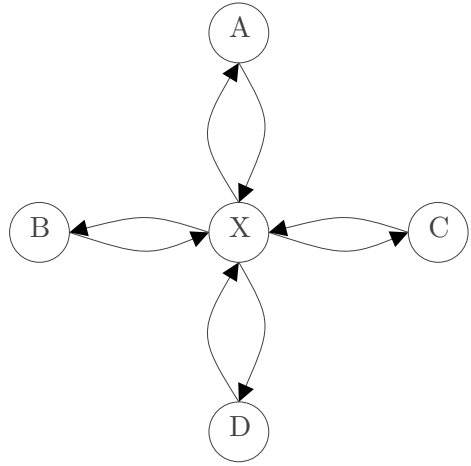
Algorithm

3 High-Level Description

Algorithm 2 Stage II

Require: G - a directed graph, F - a $DFVS$ of G, A - the acyclic remainder of $G \setminus F$

```
for all v ∈ F do
    if isAcyclic(A ∪ {v}) then
        A ← A ∪ {v}
        F ← F \ {v}
    end if
end for
return F
```



Example of a redundant vertex

Algorithm

3 High-Level Description

Algorithm 3 Stage III

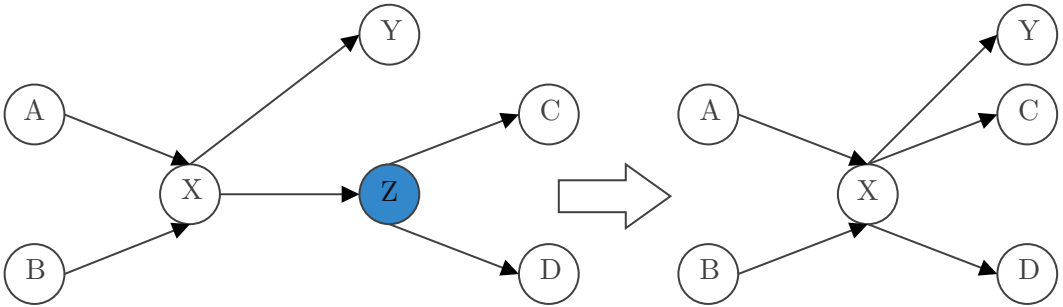
Require: G - a directed graph,
F - a $DFVS$ of G, $\alpha \in (0, 1]$
while $availableTime$ **do**
$F' \leftarrow getRandomSubset(F)$;
$G' \leftarrow G \setminus F'$;
$S \leftarrow StageI(G', \alpha)$
$S \leftarrow StageII(G', S, G' \setminus S)$
$S \leftarrow S \cup F'$
if $|S| < |F|$ **then**
$F \leftarrow S$
end if
end while
return F

A local search runs the first two stages on a subgraph of G.

Subgraphs are obtained by removing a specific subset uniformly random from the best DFVS found.

Light Reductions

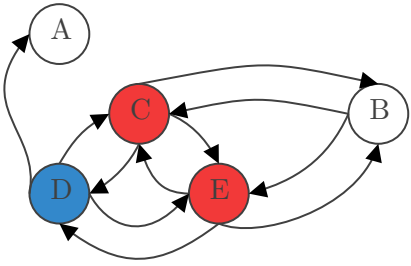
3 High-Level Description



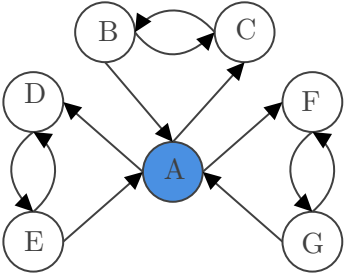
Concept of the Second Reduction

Heavy Reductions

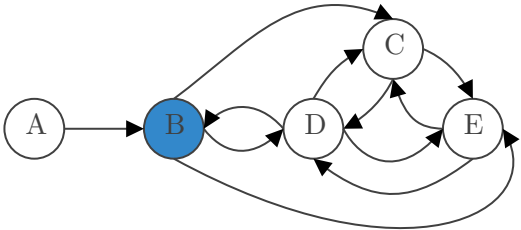
3 High-Level Description



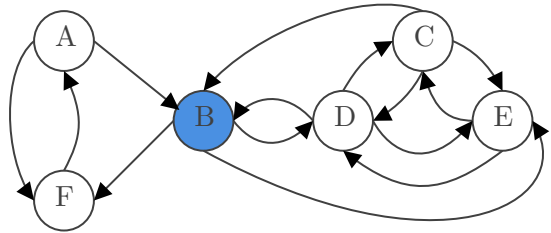
Concept of the Third Reduction



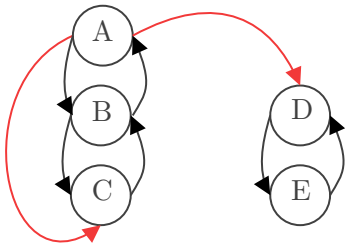
Concept of the Eighth Reduction



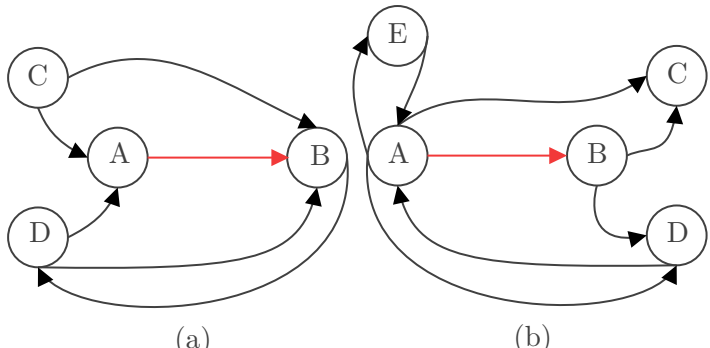
Concept of the Sixth Reduction



Concept of the Seventh Reduction



Concept of the Fourth Reduction



Concept of the Fifth Reduction

Table of Contents

Specifications

4 Implementation Details

- Source code is written in $C++$
- Adjacency lists are implemented as a vector of hash sets
- Heuristic selection of a vertex is made in $\mathcal{O}(log(|V|))$ using a priority queue
- Vertices that can be reduced by an operation of type one or two are identified in $\mathcal{O}(1)$ using a stack specifically for each reduction type
- Due to the low stack memory limit, all functions are implemented iteratively

Complexity Analysis

4 Implementation Details

- Reductions one and two can be applied in $\mathcal{O}(1)$.
- The third reduction involves sorting the nodes by the number of neighbors in $N_G^{\pm}$. It is implemented in $\mathcal{O}((|V| + |E|) \cdot log(|V|))$.
- For the others, checks and changes are made only for nodes with degrees bounded by a small constant, and I assume that the complexity is $\mathcal{O}(|V| + |E|)$
- As an instance can be reduced in $\mathcal{O}(|V|)$ to one with $|V| < |E|$, the final complexity is $\mathcal{O}(T \cdot (|V| + |E| \cdot \log |E|))$, based on Master's Theorem.
  T is the number of local searches, a parameter bounded by the time limit.

Table of Contents

5 Conclusions

Top of the Scoreboard After Two Months

5 Conclusions

#	USER	LANGUAGE	TIME [S]	SCORE	1	2	3	4	5	6	7	8
1	_UAIC_AndreiArhire_	C++	57,451.95	**460.14**	46.00	38.00	235.00	**77.00**	33.00	**74.00**	**169.00**	**71.00**
2	florian	Static binary	55,404.43	**459.94**	46.00	38.00	235.00	80.00	33.00	76.00	171.00	74.00
3	Nanored	TGZ	48,661.08	**459.39**	46.00	38.00	**235.00**	85.00	33.00	76.00	175.00	73.00
4	xuxu9110	C++	28,122.86	**458.39**	48.00	38.00	236.00	83.00	35.00	78.00	177.00	75.00
5	dcastro	CMake	59,370.08	**458.11**	46.00	38.00	**235.00**	80.00	33.00	**74.00**	174.00	**71.00**
6	ths_h	Static binary	50,692.55	**454.91**	53.00	43.00	238.00	92.00	39.00	85.00	206.00	89.00
7	GBathie	TGZ	59,603.90	**453.04**	50.00	42.00	247.00	87.00	36.00	83.00	191.00	79.00
8	mt-doom	Static binary	1,400.01	**452.81**	57.00	44.00	247.00	101.00	39.00	88.00	232.00	80.00
9	Timeroot	TGZ	34,772.98	**447.81**	**46.00**	38.00	313.00	83.00	**33.00**	75.00	179.00	73.00
10	PACE2022	C++	6,606.34	**440.57**	63.00	**38.00**	345.00	107.00	37.00	81.00	230.00	79.00
11	GraPA-JAVA	Jar	30,522.70	**438.91**	57.00	41.00	246.00	91.00	38.00	88.00	219.00	83.00
12	BabyShark	Jar	5,015.25	**438.86**	56.00	43.00	244.00	97.00	39.00	87.00	224.00	78.00
13	GRAPA-Rust	Static binary	36,650.77	**433.24**	162.00	126.00	352.00	301.00	101.00	160.00	441.00	217.00

Future Work

- Following this process, I developed an efficient and fast heuristic solver for the DFVS problem, and I submitted the solution description at a conference.
- Using a genetic algorithm and the local search can generate higher quality solutions in a longer time.
- Improving heuristic selection is challenging, as is finding a more appropriate criterion. A solution idea can start by driving a neural network depending on the number of edges of a vertex, their type, graph density, and other values.

Q&A

Thank you!

References

Chen Jianer; Liu Yang; Lu Songjian; O'sullivan Barry; Razgon Igor, *A fixed-parameter algorithm for the directed feedback vertex set problem*, Journal of the ACM **55** (2008), 1–19.

Bergougnoux Benjamin; Eiben Eduard; Ganian Robert; Ordyniak Sebastian; Ramanujan M. S., *Towards a polynomial kernel for directed feedback vertex set*, Algorithmica (2020).