A Polynomial Kernel for Funnel Arc Deletion Set

Marcelo Garlet Milani

Technische Universität Berlin

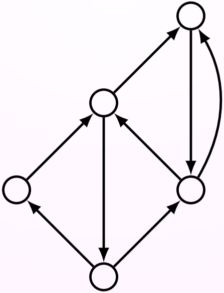
January 24, 2020

Contents

Let D be a digraph and $k \in \mathbb{N}$.

DIRECTED FEEDBACK ARC SET (DFAS)

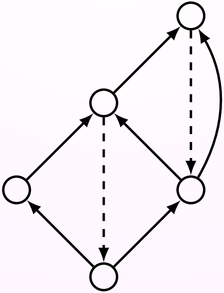
Is there some $A' \subseteq A(D)$ such that $D - A'$ is a directed acyclic graph (DAG) and $|A'| \leq k$?



Let D be a digraph and $k \in \mathbb{N}$.

Directed Feedback Arc Set (DFAS)

Is there some $A' \subseteq A(D)$ such that $D - A'$ is a directed acyclic graph (DAG) and $|A'| \leq k$?



Let D be a digraph, $\mathcal{C}$ a class of (acyclic) digraphs and $k \in \mathbb{N}$.

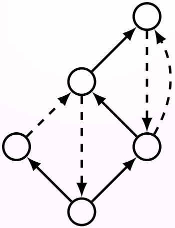
$\mathcal{C}$-Arc Deletion Set ($\mathcal{C}$-ADS)

Is there some $A' \subseteq A(D)$ such that $D - A' \in \mathcal{C}$ and $|A'| \leq k$?

Let D be a digraph, $\mathcal{C}$ a class of (acyclic) digraphs and $k \in \mathbb{N}$.

$\mathcal{C}$-Arc Deletion Set ($\mathcal{C}$-ADS)

Is there some $A' \subseteq A(D)$ such that $D - A' \in \mathcal{C}$ and $|A'| \leq k$?



Example for Out-Forest-ADS.

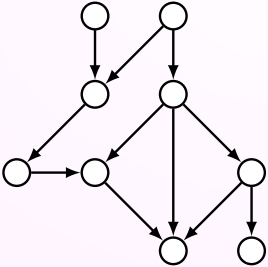
Definition (Funnel)

A DAG D is a funnel if for every source-sink path P there is some arc $a \in A(P)$ such that $Q = P$ for any source-sink path Q containing a.

Definition (Funnel)

A DAG D is a funnel if for every source-sink path P there is some arc $a \in A(P)$ such that $Q = P$ for any source-sink path Q containing a.

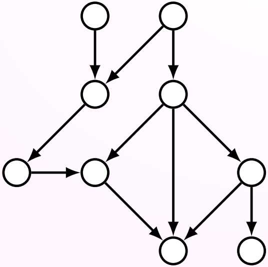
A funnel



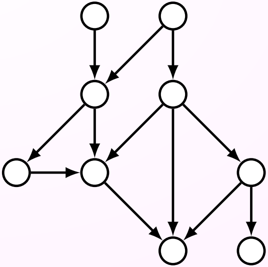
Definition (Funnel)

A DAG D is a funnel if for every source-sink path P there is some arc $a \in A(P)$ such that $Q = P$ for any source-sink path Q containing a.

A funnel

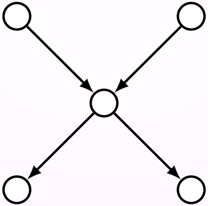


Not a funnel



Forbidden Butterfly-Minor

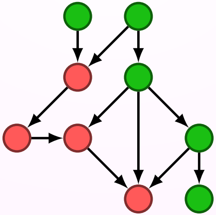
A DAG is a funnel if and only if it does not contain the following as a butterfly minor.



Partitioning

A DAG D is a funnel if and only if $V(D) = \mathrm{F} \uplus \mathrm{M}$ such that

1. $D[\mathrm{F}]$ is an out-forest,
2. $D[\mathrm{M}]$ is an in-forest, and
3. $\mathrm{M} \times \mathrm{F} \cap A(D) = \emptyset$.



- There is a $\mathcal{O}(4^k k^4 nm)$ time algorithm for DFAS/DFVS.[1]
- DFVS admits a polynomial kernel when parameterized by k and the size of a treewidth-η modulator (for constant η).[2]

[1] J. Chen, Y. Liu, S. Lu, B. O'Sullivan., I. Razgon. *A fixed-parameter algorithm for the directed feedback vertex set problem*. 2008.

[2] D. Lokshtanov, M. S. Ramanujan, S. Saurabh, R. Sharma, M. Zehavi. *Wannabe bounded treewidth graphs admit a polynomial kernel for DFVS*. 2019.

- Out-Forest-ADS and Pumpkin-ADS can be solved in polynomial time.[3]
- Out-Forest-VDS and Pumpkin-VDS are NP-hard and admit polynomial kernels wrt. solution size.[3]

[3] M. Mnich and E. J. van Leeuwen. *Polynomial kernels for deletion to classes of acyclic digraphs*. 2017.

[4] F. V. Fomin, P. A. Golovach, D. M. Thilikos. *On the Parameterized Complexity of Graph Modification to First-Order Logic Properties*. 2020

- OUT-FOREST-ADS and PUMPKIN-ADS can be solved in polynomial time.[3]
- OUT-FOREST-VDS and PUMPKIN-VDS are NP-hard and admit polynomial kernels wrt. solution size.[3]
- $\mathcal{C}$-ADS admits a polynomial kernel if $\mathcal{C}$ "can be described by a simple first-order logic formula".[4]

[3] M. Mnich and E. J. van Leeuwen. *Polynomial kernels for deletion to classes of acyclic digraphs*. 2017.

[4] F. V. Fomin, P. A. Golovach, D. M. Thilikos. *On the Parameterized Complexity of Graph Modification to First-Order Logic Properties*. 2020

- Funnel-ADS is NP-hard even if the input is a DAG.[5]
- Funnel-ADS is in FPT wrt k.[5]

[5] M. G. M., H. Molter, R. Niedermeier, and M. Sorge. *Efficient algorithms for measuring the funnel-likeness of DAGs*. 2018.

FUNNEL-ADS

- FUNNEL-ADS is NP-hard even if the input is a DAG.[5]
- FUNNEL-ADS is in FPT wrt k.[5]
- Compare: OUT-FOREST-ADS and PUMPKIN-ADS can be solved in polynomial time.

[5] M. G. M., H. Molter, R. Niedermeier, and M. Sorge. *Efficient algorithms for measuring the funnel-likeness of DAGs*. 2018.

- FUNNEL-ADS is NP-hard even if the input is a DAG.[5]
- FUNNEL-ADS is in FPT wrt k.[5]
- Compare: OUT-FOREST-ADS and PUMPKIN-ADS can be solved in polynomial time.

Theorem

FUNNEL-ADS admits a kernel with $\mathcal{O}(k^6)$ many vertices and $\mathcal{O}(k^7)$ many arcs, computable in $\mathcal{O}(n+m)$ time.

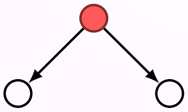
[5] M. G. M., H. Molter, R. Niedermeier, and M. Sorge. *Efficient algorithms for measuring the funnel-likeness of DAGs*. 2018.

Contents

Let D be a digraph, $\ell : V(D) \to \{\mathrm{F}, \mathrm{M}\}$ a (partial) labeling of the vertices and $k \in \mathbb{N}$.

FUNNEL ARC DELETION LABELING (FADL)

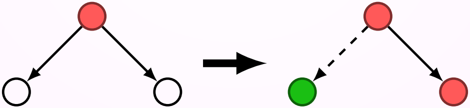
Is there some $A' \subseteq A(D)$ and some $\hat{\ell} \supseteq \ell$ such that $\hat{\ell}$ is a *funnel labeling* for $D - A'$ and $|A'| \leq k$?



Let D be a digraph, $\ell : V(D) \to \{\mathrm{F}, \mathrm{M}\}$ a (partial) labeling of the vertices and $k \in \mathbb{N}$.

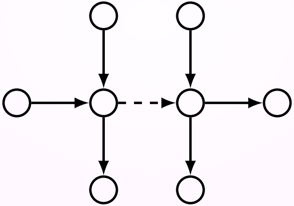
Funnel Arc Deletion Labeling (FADL)

Is there some $A' \subseteq A(D)$ and some $\hat{\ell} \supseteq \ell$ such that $\hat{\ell}$ is a *funnel labeling* for $D - A'$ and $|A'| \leq k$?



Reduction Rule

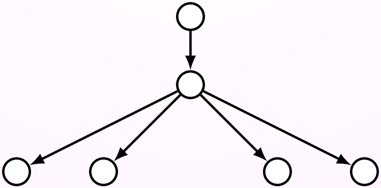
If there are more than $2k$ vertices with both in- and outdegree greater than 1, reject the input.



Basic Reduction Rules

Reduction Rule

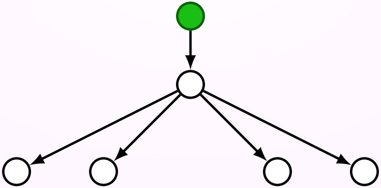
1. Label with F every source vertex.
2. Label with M every sink vertex.



Basic Reduction Rules

Reduction Rule

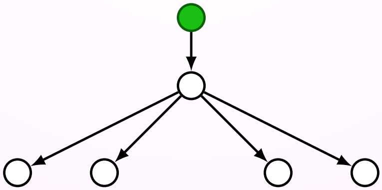
1. Label with F every source vertex.
2. Label with M every sink vertex.



Basic Reduction Rules

Reduction Rule

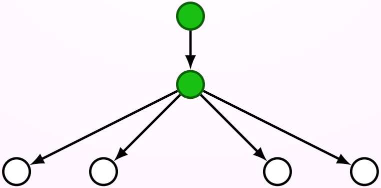
1. Label with F every source vertex.
2. Label with M every sink vertex.
3. Label with F every vertex with outdegree $> k+1$.
4. Label with M every vertex with indegree $> k+1$.



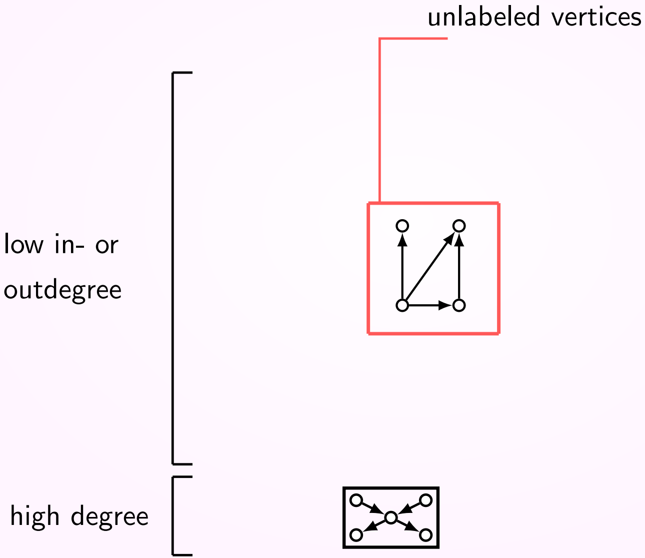
Basic Reduction Rules

Reduction Rule

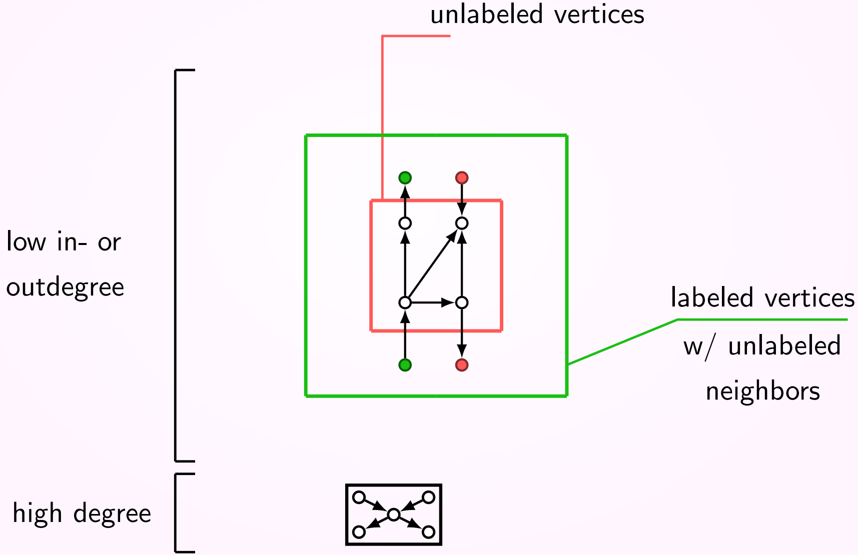
1. Label with F every source vertex.
2. Label with M every sink vertex.
3. Label with F every vertex with outdegree $> k+1$.
4. Label with M every vertex with indegree $> k+1$.



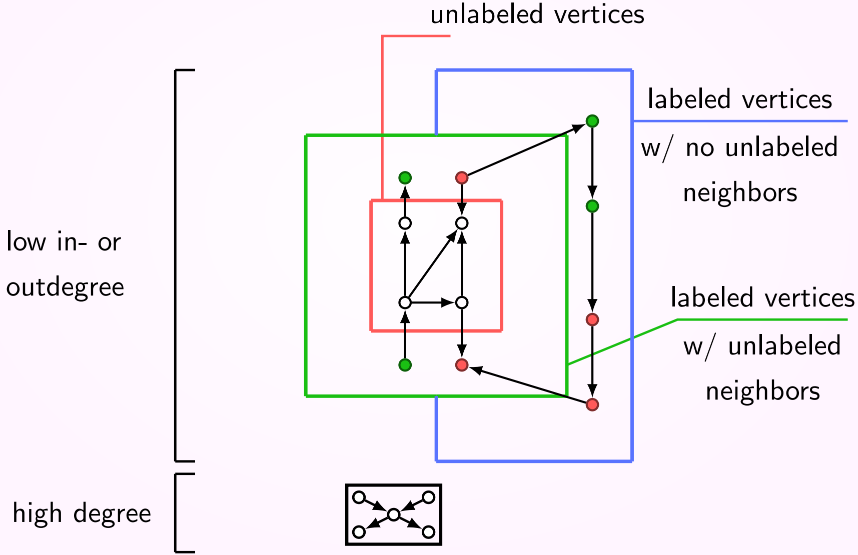
We partition D as follows.



We partition D as follows.

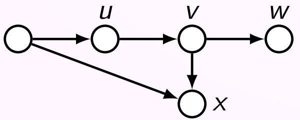


We partition D as follows.



Reduction Rule (shift neighbors)

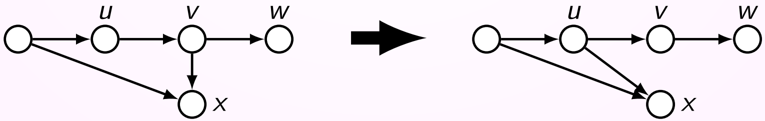
Let u, v, w be a path of unlabeled vertices such that all of them have indegree one.
We can "shift" a neighbor x of v "backwards" if it is not a neighbor of u.



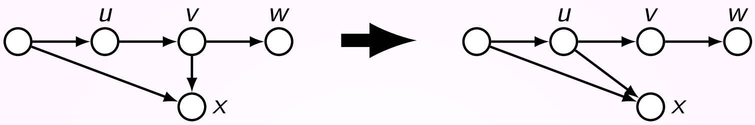
Reduction Rule (shift neighbors)

Let u, v, w be a path of unlabeled vertices such that all of them have indegree one.

We can "shift" a neighbor x of v "backwards" if it is not a neighbor of u.



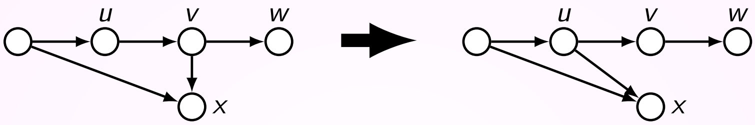
Unlabeled Vertices w/ Low Degree



We get a *win-win* situation: If there is path P where every vertex has indegree one, then

1. we can shift some vertex backwards in the path, or

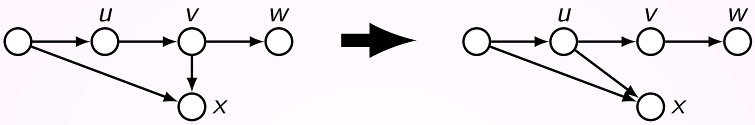
Unlabeled Vertices w/ Low Degree



We get a *win-win* situation: If there is path P where every vertex has indegree one, then

1. we can shift some vertex backwards in the path, or
2. there is a long subpath of P where every vertex also has outdegree one (and we can contract it), or

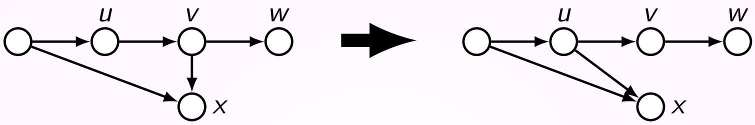
Unlabeled Vertices w/ Low Degree



We get a *win-win* situation: If there is path P where every vertex has indegree one, then

1. we can shift some vertex backwards in the path, or
2. there is a long subpath of P where every vertex also has outdegree one (and we can contract it), or
3. some $u \in V(P)$ has many neighbors (and we can label u), or

Unlabeled Vertices w/ Low Degree



We get a *win-win* situation: If there is path P where every vertex has indegree one, then

1. we can shift some vertex backwards in the path, or
2. there is a long subpath of P where every vertex also has outdegree one (and we can contract it), or
3. some $u \in V(P)$ has many neighbors (and we can label u), or
4. P is short.

Unlabeled Vertices w/ Low Degree

With an additional reduction rule, we can prove the following.

Lemma

Let D be a reduced digraph. Then there are $\mathcal{O}(k^5)$ unlabeled vertices with in- or outdegree one.

With an additional reduction rule, we can prove the following.

Lemma

Let D be a reduced digraph. Then there are $\mathcal{O}(k^5)$ unlabeled vertices with in- or outdegree one.

Corollary

There are $\mathcal{O}(k^6)$ labeled vertices with unlabeled neighbors.

Labeled Vertices w/ Labeled Neighbors

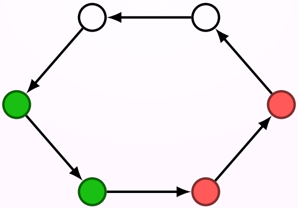
Reduction Rule (remove arcs)

Remove (v, u) if v, u have different labels. Further, set $k := k - 1$ if v is labeled with M and u is labeled with F.

Labeled Vertices w/ Labeled Neighbors

Reduction Rule (remove arcs)

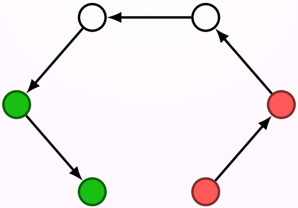
Remove (v, u) if v, u have different labels. Further, set $k := k - 1$ if v is labeled with M and u is labeled with F.



Labeled Vertices w/ Labeled Neighbors

Reduction Rule (remove arcs)

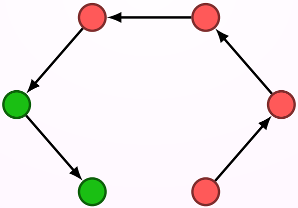
Remove (v, u) if v, u have different labels. Further, set $k := k - 1$ if v is labeled with M and u is labeled with F.



Labeled Vertices w/ Labeled Neighbors

Reduction Rule (remove arcs)

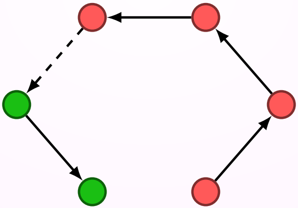
Remove (v, u) if v, u have different labels. Further, set $k := k - 1$ if v is labeled with M and u is labeled with F.



Labeled Vertices w/ Labeled Neighbors

Reduction Rule (remove arcs)

Remove (v, u) if v, u have different labels. Further, set $k := k - 1$ if v is labeled with M and u is labeled with F.



Labeled Vertices w/ Labeled Neighbors

With two additional reduction rules we can show the following.

Lemma

There are $\mathcal{O}(k)$ labeled vertices with in- or outdegree at most one whose entire neighborhood is also labeled.

Labeled Vertices w/ Labeled Neighbors

With two additional reduction rules we can show the following.

Lemma

There are $\mathcal{O}(k)$ labeled vertices with in- or outdegree at most one whose entire neighborhood is also labeled.

Putting together previous statements :

Lemma

FADL admits a problem kernel with $\mathcal{O}(k^6)$ vertices and $\mathcal{O}(k^6)$ arcs.

Lemma

FADL admits a problem kernel with $\mathcal{O}(k^6)$ vertices and $\mathcal{O}(k^6)$ arcs.

And with a simple reduction from FADL to FADS (and with a running-time analysis), we obtain:

Theorem

FADS admits a problem kernel with $\mathcal{O}(k^6)$ vertices and $\mathcal{O}(k^7)$ arcs, computable in $\mathcal{O}(n + m)$ time.

Contents

Future Work

- What about a polynomial kernel for Funnel Vertex Deletion Set?

Future Work

- What about a polynomial kernel for FUNNEL VERTEX DELETION SET?
- What would be sufficient conditions for a digraph class $\mathcal{C}$ that guarentee that $\mathcal{C}$-ADS admits a polynomial kernel?
  - Do characterizations over partitions and forbidden substructures help?

Future Work

- What about a polynomial kernel for Funnel Vertex Deletion Set?
- What would be sufficient conditions for a digraph class $\mathcal{C}$ that guarentee that $\mathcal{C}$-ADS admits a polynomial kernel?
  - Do characterizations over partitions and forbidden substructures help?
- For which pairs of digraph classes $\mathcal{C}$ and $\mathcal{D}$ the problems Directed Feedback Arc Set and $\mathcal{C}$-ADS are the same when the input digraph D is in $\mathcal{D}$?
  - Trivial if $\mathcal{D}$ is closed under arc deletion and $\mathcal{C} = \mathcal{D} \cap \text{DAGs}$.

Future Work

- What about a polynomial kernel for FUNNEL VERTEX DELETION SET?
- What would be sufficient conditions for a digraph class $\mathcal{C}$ that guarentee that $\mathcal{C}$-ADS admits a polynomial kernel?
  - Do characterizations over partitions and forbidden substructures help?
- For which pairs of digraph classes $\mathcal{C}$ and $\mathcal{D}$ the problems DIRECTED FEEDBACK ARC SET and $\mathcal{C}$-ADS are the same when the input digraph D is in $\mathcal{D}$?
  - Trivial if $\mathcal{D}$ is closed under arc deletion and $\mathcal{C} = \mathcal{D} \cap$ DAGs.

Thank you.