A Cubic Kernel for Feedback Vertex Set

Hans L. Bodlaender

Department of Information and Computing Sciences
Utrecht University
The Netherlands

Overview

Kernels and FPT

Feedback Vertex Set
- Problem definition
- Known results

The kernelization algorithm
- Start with an approximate FVS
- Trivial rules
- Adding double edges
- Removing vertices
- Abdication

Counting arguments

Conclusions

Kernels and FPT
Feedback Vertex Set
Problem definition
Known results
The kernelization algorithm
Start with an approximate FVS
Trivial rules
Adding double edges
Removing vertices
Abdication
Counting arguments
Conclusions

Preprocessing and Exact Algorithms

If you want to solve an (NP-)hard problem exactly, in practice, one often would

- First apply preprocessing / simplification: gives equivalent, hopefully smaller instance
- Use a 'slow' algorithm (e.g., ILP, branch-and-bound, ...) to solve reduced instance
- Transform solution for reduced instance to solution for original instance

Kernels and FPT

Feedback Vertex Set
- Problem definition
- Known results

The kernelization algorithm
- Start with an approximate FVS
- Trivial rules
- Adding double edges
- Removing vertices
- Abdication

Counting arguments

Conclusions

Kernels: simplification quality guarantees

- Consider a *parameterized* problem Q
  - Inputs are of form (I, k), $k \in \mathbf{N}$.
  - Decision problem
- Q is said to have a *kernel of size* $f(k)$, if there is an algorithm A, that transforms an input (I, k) of Q to another input (I', k') of Q
  - A runs in time, polynomial in $|I|$ and k
  - $Q(I, k) \Leftrightarrow Q(I', k')$
  - $k' \leq k$
  - $|I'| \leq f(k)$
- I.e., there is a polynomial time preprocessing algorithm that obtains equivalent instances of size at most $f(k)$

Kernels and FPT

Feedback Vertex Set
- Problem definition
- Known results

The kernelization algorithm
- Start with an approximate FVS
- Trivial rules
- Adding double edges
- Removing vertices
- Abdication

Counting arguments

Conclusions

Fixed Parameter Tractability

- Downey and Fellows introduced classes of parameterized problems
  - Problems in FPT (*Fixed Parameter Tractable*) have $O(f(k)n^c)$ time algorithm
  - Problems hard for $W[1]$ (or $W[2], \ldots$) are expected to require $O(n^{f(k)})$ time
- All problems in FPT have a kernel, possibly of exponential size
- Interesting question: which problems have small (polynomial size) kernels?
- Small kernels give us quality guarantees for preprocessing/simplification

Kernels and FPT
Feedback Vertex Set
Problem definition
Known results
The kernelization algorithm
Start with an approximate FVS
Trivial rules
Adding double edges
Removing vertices
Abdication
Counting arguments
Conclusions

Preventing illegal street races

- A gang is having illegal street races
- We post policemen on street corners, such that on each circuit of streets, there is a corner with a policeman

- How many policemen do we need?

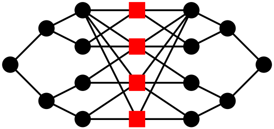
Kernels and FPT
Feedback Vertex Set
Problem definition
Known results
The kernelization algorithm
Start with an approximate FVS
Trivial rules
Adding double edges
Removing vertices
Abdication
Counting arguments
Conclusions

Problem definition

Feedback Vertex Set

- Given: Undirected graph $G = (V, E)$, integer k
- Question: is there a set of vertices $W \subseteq V$, with $G - W$ a forest, and $|W| \leq k$?

Applications in genome assembly, VLSI design, constraint satisfaction.



Related is the Loop Cut Set problem, used for Pearl's algorithm for inference on probabilistic networks

Kernels and FPT
Feedback Vertex Set
Problem definition
Known results
The kernelization algorithm
Start with an approximate FVS
Trivial rules
Adding double edges
Removing vertices
Abdication
Counting arguments
Conclusions

Known results (1)

- NP-complete
- 2-approximation algorithms (Bafna et al., 1999, Becker and Geiger, 1996)
- $O(1.8899^n)$ algorithm, Razgan 2006
- $O(2^{9.264\sqrt{n}})$ time on planar graphs (Dorn et al., 2005)
- FPT: many results

Kernels and FPT
Feedback Vertex Set
Problem definition
Known results
The kernelization algorithm
Start with an approximate FVS
Trivial rules
Adding double edges
Removing vertices
Abdication
Counting arguments
Conclusions

FPT results for FVS

- $O(f(k)n)$ time (Bodlaender, 1994)
- (Downey and Fellows, 1999)
- $O(4^k n)$ expected, probabilistic algorithm (Becker and Geiger, 2000)
- (Raman et al, 2002); (Kanj et al, 2004); (Raman et al., 2005), (Guo et al., 2005)
- $O(10.567^k p(n))$ time, (Dehne et al. 2006)
- A kernel of size $O(k^{11})$ (Burrage et al., 2006)
- A kernel of size $O(k^3)$ (this paper)

Kernels and FPT
Feedback Vertex Set
Problem definition
Known results
The kernelization algorithm
Start with an approximate FVS
Trivial rules
Adding double edges
Removing vertices
Abdication
Counting arguments
Conclusions

Main result

Theorem

There is a polynomial time algorithm, that given an undirected graph $G = (V, E)$, (possibly with parallel edges) and an integer k, gives a graph G' and an integer k', such that $k' \leq k$, G' has $O(k^3)$ vertices and edges, and G has a feedback vertex set of size at most k, if and only if G' has a feedback vertex set of size at most k'.

- $\exists$ constructive version: a FVS of G' can be translated to a FVS of G.

Kernels and FPT

Feedback Vertex Set
- Problem definition
- Known results

The kernelization algorithm
- Start with an approximate FVS
- Trivial rules
- Adding double edges
- Removing vertices
- Abdication

Counting arguments

Conclusions

Universiteit Utrecht

Structure of the algorithm

- Input: Undirected graph $G = (V, E)$, integer k.
- For an easier algorithm, we allow the graph to have *parallel* edges.
- Run 2-approximation algorithm for FVS
  - If it gives FVS of size $> 2k$: stop
  - Otherwise: we have FVS A of size at most $2k$
- Repeat until no longer possible
  - Apply a reduction or improvement rule
- A counting argument shows that resulting graph has $O(k^3)$ vertices and edges

Kernels and FPT
Feedback Vertex Set
Problem definition
Known results
The kernelization algorithm
Start with an approximate FVS
Trivial rules
Adding double edges
Removing vertices
Abdication
Counting arguments
Conclusions

Invariants

- A is a FVS of size at most $2k$. If we decrease k, then restart the initialization step on the current graph
- We maintain a set $B \subseteq V - A$: B is the set of vertices with a parallel edge to a vertex in A

Kernels and FPT

Feedback Vertex Set

Problem definition
Known results

The kernelization algorithm

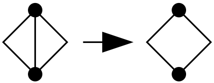
Start with an approximate FVS
Trivial rules
Adding double edges
Removing vertices
Abdication

Counting arguments

Conclusions

Parallel edges

- We allow the graph to have parallel edges.
- If there are two or more edges $\{v, w\}$, then each FVS contains v or w.



Rule (Parallel edges rule)

If there are three or more parallel edges $\{v, w\}$, then remove all but two of these.

Kernels and FPT
Feedback Vertex Set
Problem definition
Known results
The kernelization algorithm
Start with an approximate FVS
Trivial rules
Adding double edges
Removing vertices
Abdication
Counting arguments
Conclusions

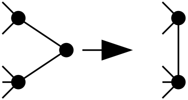
Removing low degree vertices

Rule (Degree 0 and 1 rule)

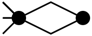
If v has degree at most one, remove v

Rule (Degree two rule)

If v has degree two, and has neighbors w, x, $w \neq x$, then remove w and add the edge $\{w, x\}$.



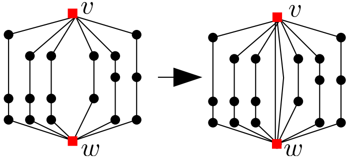
If v has degree two, with two edges to w, then remove v and w, and decrease k by one.



Kernels and FPT
Feedback Vertex Set
Problem definition
Known results
The kernelization algorithm
Start with an approximate FVS
Trivial rules
Adding double edges
Removing vertices
Abdication
Counting arguments
Conclusions

Many paths

- If there are at least $k+2$ vertex disjoint paths from v to w, then each FVS of size at most k contains v or w.



Rule (Many paths rule)

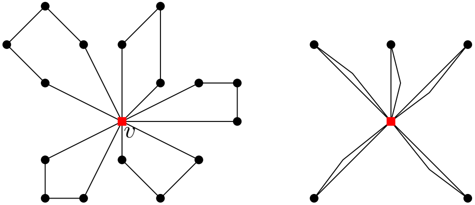
If there are at least $k+2$ vertex disjoint paths from v to w, then add a double edge $\{v, w\}$.

Kernels and FPT
Feedback Vertex Set
Problem definition
Known results
The kernelization algorithm
Start with an approximate FVS
Trivial rules
Adding double edges
Removing vertices
Abdication
Counting arguments
Conclusions

Flowers

Rule (Flower rule)

If there are $k+1$ cycles that mutually intersect only in v, then remove v from G, and set $k = k-1$.



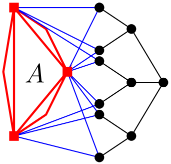
Rule (Large double degree)

If v is incident to at least $k+1$ double edges, then remove v from G, and set $k = k-1$.

Kernels and FPT
Feedback Vertex Set
Problem definition
Known results
The kernelization algorithm
Start with an approximate FVS
Trivial rules
Adding double edges
Removing vertices
Abdication
Counting arguments
Conclusions

Rulers and abdication

- $G - (A \cup B)$ is a forest
- The *border* of a tree T in this forest is the set of vertices in $A \cup B$ adjacent to T
- Two *abdication* rules help to remove forests whose border is a 'double clique'
- $v \in A \cup B$ *rules* tree T, if it has a double edge to each other vertex in the border of T



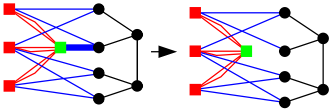
Kernels and FPT
Feedback Vertex Set
Problem definition
Known results
The kernelization algorithm
Start with an approximate FVS
Trivial rules
Adding double edges
Removing vertices
Abdication
Counting arguments
Conclusions

Universiteit Utrecht

First abdication rule

Let T be a tree in the forest $G - (A \cup B)$.
If v rules tree T, and it has one edge to T, say $\{v, w\}$m then a minimum FVS of G is a minimum FVS of $G - \{v, w\}$, and vice versa.

- Either v or all other vertices in the border belong to the FVS, so each cycle that contains edge $\{v, w\}$ has a vertex on the FVS



Rule (First abdication rule)

If v rules tree T, and has one edge to T, then remove this edge.

Kernels and FPT
Feedback Vertex Set
Problem definition
Known results
The kernelization algorithm
Start with an approximate FVS
Trivial rules
Adding double edges
Removing vertices
Abdication
Counting arguments
Conclusions

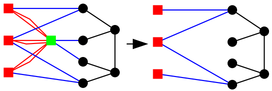
Second abdication rule

Let T be a tree in the forest $G - (A \cup B)$
If v rules tree T, and it has *at least two* edges to T, then v belongs to a minimum size FVS

- If $v \notin S$, S a FVS, then S contains all vertices in the border, and at least one vertex w in T. Now $S - w + v$ is also a minimum FVS.

Rule (Second abdication rule)

If v rules tree T, and has at least two edges to T, then remove v, and decrease k by one.



Kernels and FPT
Feedback Vertex Set
Problem definition
Known results
The kernelization algorithm
Start with an approximate FVS
Trivial rules
Adding double edges
Removing vertices
Abdication
Counting arguments
Conclusions

The algorithm

- For each of the rules, we can find in polynomial time if it can be applied.
  - Flower rule: generalized matching.
  - Many paths rule: network flow.
- The algorithm is:
  - Initialize A and B.
  - Repeatedly try to apply any of the given rules until none is possible.

Theorem

The obtained kernel has $O(k^3)$ vertices and edges.

Too detailed arguments for this talk. Just a bit of flavour ...

Kernels and FPT
Feedback Vertex Set
Problem definition
Known results
The kernelization algorithm
Start with an approximate FVS
Trivial rules
Adding double edges
Removing vertices
Abdication
Counting arguments
Conclusions

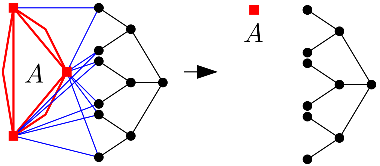
Counting different types of vertices

1. A: vertices in the approximate FVS: $O(k)$.
2. B: vertices not in A, with a double edge to a vertex in A: $O(k^2)$.
   - If A has more than k incident double edges, then Large Double Degree rule applies.
3. C: vertices x in $V - (A \cup B)$ adjacent to a vertex v in $A \cup B$. (Detailed arguments needed.)
4. D: all other vertices: vertices in $V - (A \cup B)$ who only have neighbors in $V - (A \cup B)$. As $G[C \cup D]$ is a forest and vertices in D have degree ≥ 3 in $G[C \cup D]$, $|D| < |C|$.

Kernels and FPT
Feedback Vertex Set
Problem definition
Known results
The kernelization algorithm
Start with an approximate FVS
Trivial rules
Adding double edges
Removing vertices
Abdication
Counting arguments
Conclusions

Trees with a double clique border

- Suppose T is a tree is $G - (V \cup A)$, and there is a double edge between each pair of vertices in the border of T.
- By the abdication rules, all edges from T to its border (and possibly some vertices in the border) will be removed.
- After this T itself will be removed by Degree 0 and 1 rule
- For each tree T in $G - (A \cup B)$, there are two vertices in its border that do not have a double edge between them.



Kernels and FPT
Feedback Vertex Set
Problem definition
Known results
The kernelization algorithm
Start with an approximate FVS
Trivial rules
Adding double edges
Removing vertices
Abdication
Counting arguments
Conclusions

Universiteit Utrecht

A bound on the number of trees

Lemma

There are $O(k^5)$ trees in the kernel.

- For each tree T in $G - (A \cup B)$, there are two vertices in its border that do not have a double edge between them.
- Assign T to one such pair.
- If v, w have at least $k + 2$ trees assigned, then Many paths rule applies.
- So, $O((|A| + |B|)^2)$ pairs each having $\leq k + 1$ trees assigned.
- This gives $O(k^5)$ trees in $G - (A \cup B)$.

A more detailed proof gives $O(k^3)$.

Kernels and FPT
Feedback Vertex Set
Problem definition
Known results
The kernelization algorithm
Start with an approximate FVS
Trivial rules
Adding double edges
Removing vertices
Abdication
Counting arguments
Conclusions

A combinatorial result

Theorem

Let $T = (V_T, E_T)$ be a tree, and let $W \subseteq V_T$ be a set of vertices, with $|W| \geq k^2 + 3k + 4$. Then one of the following two cases holds:

1. *There are at least $k+1$ vertex disjoint paths between vertices in W.*
2. *There is a vertex $x \in V_T$, such that there are $k+2$ vertices in W with disjoint paths to x.*

Kernels and FPT
Feedback Vertex Set
Problem definition
Known results
The kernelization algorithm
Start with an approximate FVS
Trivial rules
Adding double edges
Removing vertices
Abdication
Counting arguments
Conclusions

A bound on C

Corollary

Let $v \in A \cup B$, T a tree in $G - (A \cup B)$ and W the set of vertices adjacent to v in T. Then $|W| \leq k^2 + 3k + 3$.

Proof.

If $|W| \geq k^2 + 3k + 3$, use previous result.
If there are $k + 1$ vertex disjoint paths between vertices in W, these paths and edges to v form a flower: Flower rule applies.
If there are $k + 2$ vertices in W with disjoint paths to x, we have $k + 2$ disjoint paths from v to x, and x is placed in B. □

- $|C| = O(k^9)$: for each of the $O(k^2)$ vertices $v \in A \cup B$, and each of the $O(k^5)$ trees T, T contains $O(k^2)$ vertices adjacent to v.

Kernels and FPT
Feedback Vertex Set
Problem definition
Known results
The kernelization algorithm
Start with an approximate FVS
Trivial rules
Adding double edges
Removing vertices
Abdication
Counting arguments
Conclusions

Universiteit Utrecht

Counting all vertices

- $|A| = O(k)$.
- $|B| = O(k^2)$.
- $|C| = O(k^9)$.
- Each vertex in D has at least three neighbors in $C \cup D$.
- The forest induced by $C \cup D$ has at most $|C|$ leaves, hence less than C vertices of degree at least three.
- $|D| < |C| = O(k^9)$.
- $|V| = |A| + |B| + |C| + |D| = O(k^9)$.

As said: $O(k^3)$ vertices is possible with more detailed proofs.

Kernels and FPT
Feedback Vertex Set
Problem definition
Known results
The kernelization algorithm
Start with an approximate FVS
Trivial rules
Adding double edges
Removing vertices
Abdication
Counting arguments
Conclusions

Conclusions and further work

- Kernelization: quality guarantees for preprocessing.
- $O(k^3)$ with the same algorithm, but more careful counting
- Also: $O(k^3)$ kernel for the Loop Cut Set problem used for Pearl's inference algorithm for probabilistic networks (with Thomas van Dijk).
- Recent result: $O(k)$ kernel for FVS on *planar graphs* (with Eelko Penninkx).
- Open: polynomial kernel for Disjoint Cycles
- Open and hard: Directed Feedback Vertex Set and Directed Feedback Arc Set in FPT?

Kernels and FPT
Feedback Vertex Set
Problem definition
Known results
The kernelization algorithm
Start with an approximate FVS
Trivial rules
Adding double edges
Removing vertices
Abdication
Counting arguments
Conclusions