Kernels for FEEDBACK ARC SET IN TOURNAMENTS

Anthony Perez

Joint work with S. Bessy, F. V. Fomin, S. Gaspers, C. Paul, S. Saurabh, S. Thomassé

Université Montpellier II - LIRMM

JGA'09 - Montpellier

Plan

1. Exact resolution - parameterized algorithms
2. Kernels for k-FAST
   - Definitions and structural results
   - Reduction rules and size
3. Conclusion

Plan

1. Exact resolution - parameterized algorithms
2. Kernels for *k*-FAST
   - Definitions and structural results
   - Reduction rules and size
3. Conclusion

Parameterized algorithm

A problem parameterized by $k \in \mathbb{N}$ is said to be fixed-parameter tractable (in *FPT*) if it can be solved in time $f(k).n^{O(1)}$.

Remarks

The function f considered can be anything and depends only on the parameter k. Thus, the function $f(k) = 2^{2^{2^k}}$ is good.

Parameterized algorithm

A problem parameterized by $k \in \mathbb{N}$ is said to be fixed-parameter tractable (in *FPT*) if it can be solved in time $f(k).n^{O(1)}$.

Remarks

The function f considered can be anything and depends only on the parameter k. Thus, the function $f(k) = 2^{2^{2^k}}$ is good.

Kernelization

Given a parameterized problem Π and $(x, k) \in \Pi$, a **kernelization** is a polynomial-time algorithm *(set of reduction rules)* that takes as input $(x, k) \in \Pi$ and outputs $(x', k') \in \Pi$ s.t. :

- x is a YES-instance $\Leftrightarrow$ x' is a YES-instance
- $|x'| \leq h(k)$
- $k' \leq k$

Theorem

$\Pi \in FPT \Leftrightarrow \Pi$ has a kernel (size : exponential).

Kernelization

Given a parameterized problem Π and $(x, k) \in \Pi$, a **kernelization** is a polynomial-time algorithm *(set of reduction rules)* that takes as input $(x, k) \in \Pi$ and outputs $(x', k') \in \Pi$ s.t. :

- x is a YES-instance $\Leftrightarrow$ x' is a YES-instance
- $|x'| \leq h(k)$
- $k' \leq k$

Theorem

$\Pi \in FPT \Leftrightarrow \Pi$ has a kernel (size : exponential).

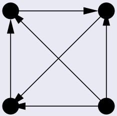
Consequences

- *Pre-processing*
- $\Rightarrow$ Reducing the size of a given input
- $\Rightarrow$ Resolution on kernels
- $\Rightarrow$ Additive complexity $(O(g(k) + poly(n)))$.

Plan

Tournament

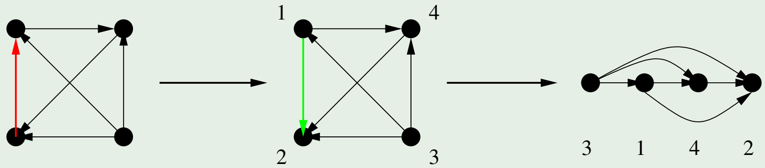
- A *tournament* is a graph obtained by orienting every edge of a complete graph.
- A *tournament* is acyclic iff it has a transitive ordering of its vertices.
- A *feedback arc set* for a tournament $T = (V, A)$ is a (minimum) set of arcs whose removal makes T acyclic.



Lemma (Raman, Saurabh, TCS 2006)

Let $D = (V, A)$ be a digraph and F a feedback arc set for D. The graph D' obtained by reversing all arcs of F in D is acyclic.

FEEDBACK ARC SET IN TOURNAMENTS



From now on, we order the vertices of the tournament under some σ.

Certificate

Let $T_\sigma = (V, A)$ be an ordered tournament, and $f = vu$ any of its backward arc. A *certificate* is a directed path P from u to v such that :

(i) P belongs to the span of f,

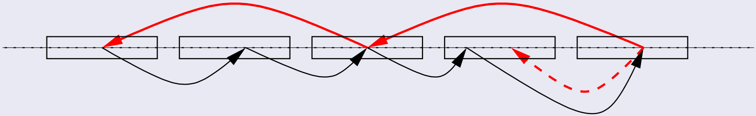
(ii) P uses only forward arcs.

Let $T_\sigma = (V, A)$ be an ordered tournament, and F a set of backward arcs. We say that we can *certify* F if it is possible to find a family of $|F|$ arc-disjoint certificates for the arcs in F.

From now on, we order the vertices of the tournament under some σ.

Safe partition

Let $T_\sigma = (V, A)$ be an ordered tournament, and $\mathcal{P} := \{V_1, \ldots, V_p\}$ a partition of its vertex set into intervals. $\mathcal{P}$ is a *safe partition* if all backward arcs between the V_is can be certified using only arcs between the V_is.



Plan

Lemma ("well-known")

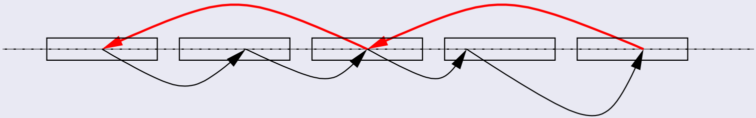
A tournament is acyclic iff it does not contain any (directed) triangle.

Reduction rule 1

Let v be a vertex that does not belong to any triangle. Remove v from T.

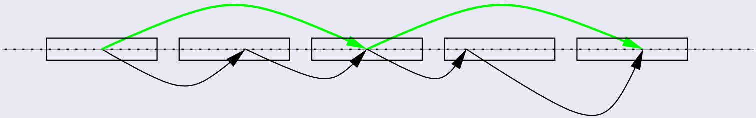
Reduction rule 2

Let $T = (V, A)$ be an ordered tournament, $\mathcal{P} := \{V_1, \ldots, V_l\}$ be a safe partition of V into intervals and F be the set of backward arcs between the intervals. Then reverse all arcs of F and decrease k by $|F|$.

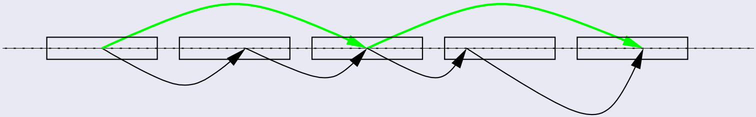


Reduction rule 2

Let $T = (V, A)$ be an ordered tournament, $\mathcal{P} := \{V_1, \ldots, V_l\}$ be a safe partition of V into intervals and F be the set of backward arcs between the intervals. Then reverse all arcs of F and decrease k by $|F|$.



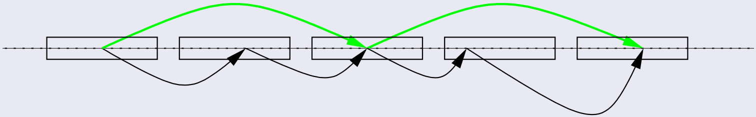
Reduction rule 2



Proof

- As $\mathcal{P}$ is a safe partition, all backward arcs between the parts of $\mathcal{P}$ can be certified (meaning that every such arc belongs to a directed cycle) without using any arc inside the parts.
- By definition, the others directed cycles are entirely contained in the parts of $\mathcal{P}$. Thus the edition of the directed cycles between parts is independent from any other edition.

Reduction rule 2



Proof

- As $\mathcal{P}$ is a safe partition, all backward arcs between the parts of $\mathcal{P}$ can be certified (meaning that every such arc belongs to a directed cycle) without using any arc inside the parts.
- By definition, the others directed cycles are entirely contained in the parts of $\mathcal{P}$. Thus the edition of the directed cycles between parts is independent from any other edition.

Lemma

Let $T = (V, A)$ be an ordered tournament with $|V| \geqslant 2p + 1$ and at most p backward arcs, where $p \geqslant 1$. Then we can find a safe partition (in polynomial time) having at least one backward arc between its parts.

Theorem

Let $T = (V, A)$ be any tournament. For every $\epsilon > 0$, there is a polynomial time algorithm that reduces T to an equivalent instance T' having at most $(2 + \epsilon)k$ vertices.

Proof

- Using a known PTAS for k-FAST, start with a feedback arc set F of at most $(1 + \frac{\epsilon}{2})$ arcs.
- Order the vertices according to the transitive ordering obtained by reversing F. Observe that such an ordering has at most $(1 + \frac{\epsilon}{2})$ backward arcs.
- Using the Lemma, we reduce the graph (finding a safe partition) as long as it has more than $(2 + \epsilon)k$ vertices.

Theorem

Let $T = (V, A)$ be any tournament. For every $\epsilon > 0$, there is a polynomial time algorithm that reduces T to an equivalent instance T' having at most $(2 + \epsilon)k$ vertices.

Proof

- Using a known PTAS for k-FAST, start with a feedback arc set F of at most $(1 + \frac{\epsilon}{2})$ arcs.
- Order the vertices according to the transitive ordering obtained by reversing F. Observe that such an ordering has at most $(1 + \frac{\epsilon}{2})$ backward arcs.
- Using the Lemma, we reduce the graph (finding a safe partition) as long as it has more than $(2 + \epsilon)k$ vertices.

Plan

Perspectives and open problems

- Similar results for FEEDBACK VERTEX SET IN TOURNAMENTS ?
- What about FEEDBACK ARC SET in general graphs ?
- Use approximation to obtain kernels for other problems ?