Generalized feedback vertex set problems on bounded-treewidth graphs: chordality is the key to single-exponential parameterized algorithms

Édouard Bonnet, Nick Brettell, O-joung Kwon, Dániel Marx

Middlesex University, London

September 6, 2017, IPEC, Vienna

Single-exponential algorithm parameterized by treewidth w for connectivity problems

Before 2011, the $2^{O(w \log w)} n^{O(1)}$-time algorithm for FEEDBACK VERTEX SET was believed to be optimal, but...

- Cut&Count [Cygan et al. '11]
  $\rightarrow$ randomized $3^w n^{O(1)}$
- Rank and **representative sets** [Bodlaender et al. '15]
  $\rightarrow$ deterministic $2^{O(w)} n^{O(1)}$
- Extension to matroids [Fomin et al. '14]

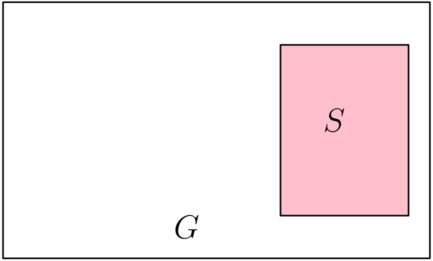
Single-exponential algorithm parameterized by treewidth w for connectivity problems

Before 2011, the $2^{O(w \log w)} n^{O(1)}$-time algorithm for FEEDBACK VERTEX SET was believed to be optimal, but...

- Cut&Count [Cygan et al. '11]
  $\rightarrow$ randomized $3^w n^{O(1)}$
- Rank and **representative sets** [Bodlaender et al. '15]
  $\rightarrow$ deterministic $2^{O(w)} n^{O(1)}$
- Extension to matroids [Fomin et al. '14]

w-boundaried graphs

- A w-**boundaried graph** is a pair (G, S) of a graph G and a subset $S \subseteq V(G)$ of size at most w.
- $S := \partial(G)$ is the **boundary** of G, each vertex in S is called a **boundary vertex**.



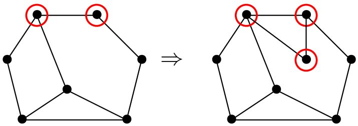
Characterisation of treewidth via w-boundaried graphs

(1) Introduce a vertex v with boundary $\{v\}$.

Characterisation of treewidth via w-boundaried graphs

(1) Introduce a vertex v with boundary $\{v\}$.

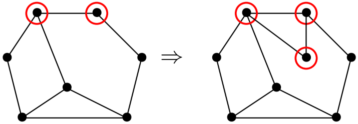
(2) Add a vertex v to (G, S) such that v has only neighbors on S and $S \cup \{v\}$ is a new boundary.



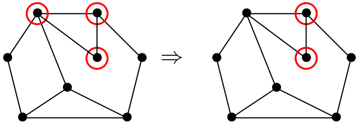
Characterisation of treewidth via w-boundaried graphs

(1) Introduce a vertex v with boundary $\{v\}$.

(2) Add a vertex v to (G, S) such that v has only neighbors on S and $S \cup \{v\}$ is a new boundary.



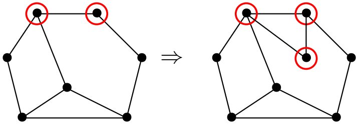
(3) Replace a boundary vertex with a non-boundary vertex



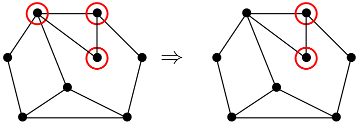
Characterisation of treewidth via w-boundaried graphs

(1) Introduce a vertex v with boundary $\{v\}$.

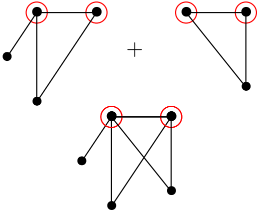
(2) Add a vertex v to (G, S) such that v has only neighbors on S and $S \cup \{v\}$ is a new boundary.



(3) Replace a boundary vertex with a non-boundary vertex



(4) Take the disjoint union of two graphs (G_1, S), (G_2, S) where $G_1[S] = G_2[S]$ and identify each vertex of S.



Treewidth

- G has treewidth $\leqslant w$ iff G can be built with those four operations by $(w+1)$-boundaried graphs in a tree-like way.

- Trees have treewidth $\leqslant 1$.
- Cycles have treewidth $\leqslant 2$.

This is just (a slight reformulation of) tree-decomposition.
As usual, we will do dynamic programming on this tree.

Treewidth

- G has treewidth $\leqslant w$ iff G can be built with those four operations by $(w+1)$-boundaried graphs in a tree-like way.

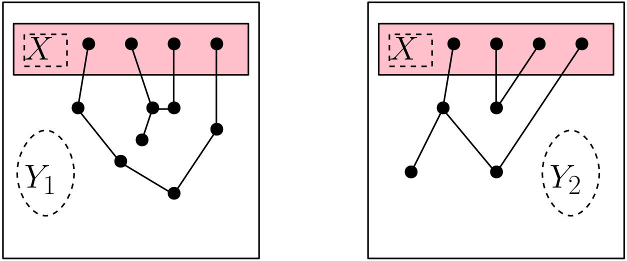
- Trees have treewidth $\leqslant 1$.
- Cycles have treewidth $\leqslant 2$.

This is just (a slight reformulation of) tree-decomposition.
As usual, we will do dynamic programming on this tree.

Feedback Vertex Set

Find a minimum vertex set S such that $G - S$ is a forest

- Partial solution of G_t at a node t:
  - X to delete in $\partial(G_t)$
  - Y to delete in $G_t - \partial(G_t)$
  - and the forest $G_t - (X \cup Y)$.



- The important information is whether or not two remaining vertices on the boundary are connected to each other or not.

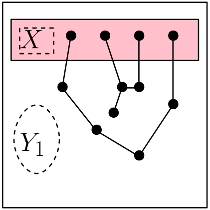


table c: for every $X \subseteq \partial(G_t)$, integer $1 \leqslant \ell \leqslant n$ and partition $\mathcal{P}$ of $\partial(G_t) \backslash X$,

- $c[t, X, \ell, \mathcal{P}] = 1$ if there is a set Y of size ℓ in $G_t - \partial(G_t)$ where $G_t - (X \cup Y)$ is a forest respecting the partition $\mathcal{P}$, and 0 otherwise.

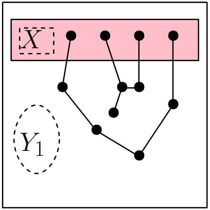


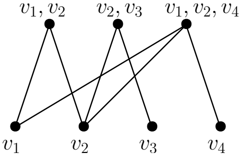
table c: for every $X \subseteq \partial(G_t)$, integer $1 \leqslant \ell \leqslant n$ and partition $\mathcal{P}$ of $\partial(G_t) \backslash X$,

- $c[t, X, \ell, \mathcal{P}] = 1$ if there is a set Y of size ℓ in $G_t - \partial(G_t)$ where $G_t - (X \cup Y)$ is a forest respecting the partition $\mathcal{P}$, and 0 otherwise.

- The number of partitions on w elements? $= 2^{O(w \log w)}$.
- This gives an algorithm running in time $2^{O(w \log w)} n^{O(1)}$.

Acyclicity of hypergraphs

- $Inc(V, E)$: incidence bipartite graph of hypergraph (V, E).
- Example : $V = \{v_1, v_2, v_3, v_4\}$, $E = \{\{v_1, v_2\}, \{v_2, v_3\}, \{v_1, v_2, v_4\}\}$.



- A hypergraph (V, E) is **acyclic** if $Inc(V, E)$ has no cycles.

Representative sets

For $\mathcal{A}$ and $\mathcal{B}$ two families of partitions over U,
a subset $\mathcal{A}' \subseteq \mathcal{A}$ is a **representative set** with respect to $\mathcal{B}$ if

- For every $\mathcal{P} \in \mathcal{A}$ and $\mathcal{Q} \in \mathcal{B}$ with $(U, \mathcal{P} \cup \mathcal{Q})$ is acyclic, there exists $\mathcal{P}' \in \mathcal{A}'$ where $(U, \mathcal{P}' \cup \mathcal{Q})$ is acyclic.

Corollary of Bodlaender, Cygan, Kratsch, and Nederlof '15

Given families $\mathcal{A}$, $\mathcal{B}$ of partitions of a set U, one can output a representative set of $\mathcal{A}$ of size at most $|U| \cdot 2^{|U|-1}$ in time $\mathcal{A}^{O(1)} 2^{O(|U|)}$.

Shrink the number of partitions in the table c from $2^{O(w \log w)}$ down to $2^{O(w)}$.

Representative sets

For $\mathcal{A}$ and $\mathcal{B}$ two families of partitions over U,
a subset $\mathcal{A}' \subseteq \mathcal{A}$ is a **representative set** with respect to $\mathcal{B}$ if

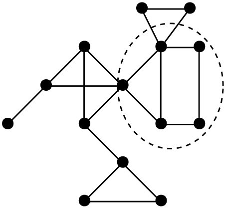
- For every $\mathcal{P} \in \mathcal{A}$ and $\mathcal{Q} \in \mathcal{B}$ with $(U, \mathcal{P} \cup \mathcal{Q})$ is acyclic, there exists $\mathcal{P}' \in \mathcal{A}'$ where $(U, \mathcal{P}' \cup \mathcal{Q})$ is acyclic.

Corollary of Bodlaender, Cygan, Kratsch, and Nederlof '15

Given families $\mathcal{A}$, $\mathcal{B}$ of partitions of a set U, one can output a representative set of $\mathcal{A}$ of size at most $|U| \cdot 2^{|U|-1}$ in time $\mathcal{A}^{O(1)} 2^{O(|U|)}$.

Shrink the number of partitions in the table c from $2^{O(w \log w)}$ down to $2^{O(w)}$.

- A **block** B is a maximal induced subgraph with no cut vertex.



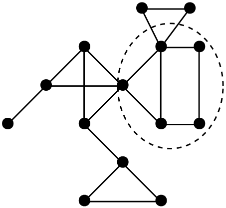
- Forests are graphs whose blocks have $\leqslant 2$ vertices.

Generalized Feedback Vertex Set

Find a vertex set S of size at most k such that each block of $G - S$ has $\leqslant d$ vertices and belongs to a fixed class $\mathcal{P}$.

Feedback Vertex Set can be obtained by setting d to 2.

- A **block** B is a maximal induced subgraph with no cut vertex.



- Forests are graphs whose blocks have $\leqslant 2$ vertices.

Generalized Feedback Vertex Set

Find a vertex set S of size at most k such that each block of $G - S$ has $\leqslant d$ vertices and belongs to a fixed class $\mathcal{P}$.

Feedback Vertex Set can be obtained by setting d to 2.

B., Brettell, Kwon, Marx '16

(1) GFVS can be solved in time $2^{O(k \log d)} poly(n)$.

(2) Under ETH, GFVS cannot be solved in time $2^{o(k \log d)} poly(n)$.

(3) GFVS is W[1]-hard parameterized by only k or d.

B., Brettell, Kwon, Marx '16

(1) GFVS can be solved in time $2^{O(k \log d)} poly(n)$.

(2) Under ETH, GFVS cannot be solved in time $2^{o(k \log d)} poly(n)$.

(3) GFVS is W[1]-hard parameterized by only k or d.

- Open question : can GFVS be solved in time $2^{O(k+d)} poly(n)$?

- YES-instances have treewidth $\leqslant k + d$.
  Can we solve GFVS in time $2^{O(w)} n^{O(1)}$?

Chordality of $\mathcal{P}$ is key

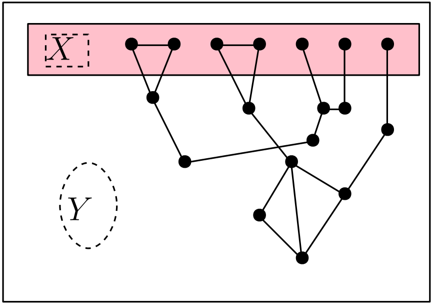
A **chordal** graph is a graph having no induced cycle of length at least 4.

Assume $\mathcal{P}$ is **block-hereditary** (i.e., closed by biconnected induced subgraph) and recognizable in polynomial time.

Our main result

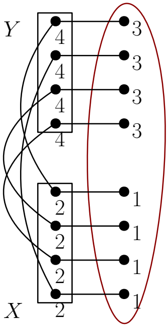
- If $\mathcal{P}$ consists only of chordal graphs, then the problem can be solved in time $2^{O(wd^2)}n^{O(1)}$.
- If $\mathcal{P}$ contains a graph with an induced cycle of length $\ell \geqslant 4$, then the problem is not solvable in time $2^{o(w \log w)}n^{O(1)}$ even for fixed $d = \ell$, unless the ETH fails.

Partial solutions



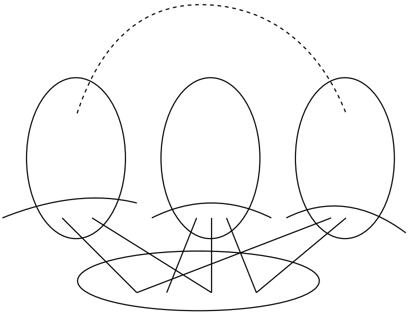
- If two vertices $v, w \in \partial(G_t)\backslash X$ are contained in the same block of $G_t - (X \cup Y)$, then they will be in the same block of $G - S$.
- For each block of $G_t[\partial(G_t)\backslash X]$, we guess the final shape of it ($2^{O(d^2)}$ guesses).
- What about the vertices of $\partial(G_t)\backslash X$ which are not yet in the same block but will eventually be?

Problem?



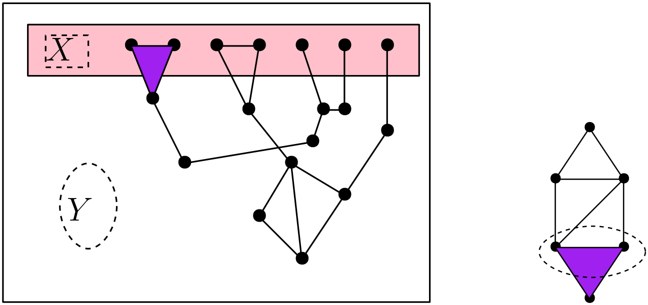
- An example showing $2^{\Theta(w \log w)}$ possible 4-labeled graphs.

Local separator in chordal graphs



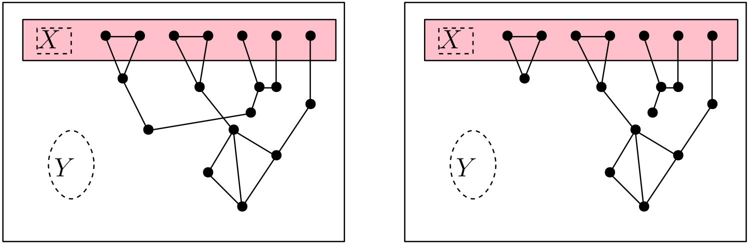
- In chordal graphs, there is a 1-to-1 mapping between components of $G - S$ and components of $G[N(S)]$.
- Map U of $G[N(S)]$ to the component of $G - S$ containing U.

First step



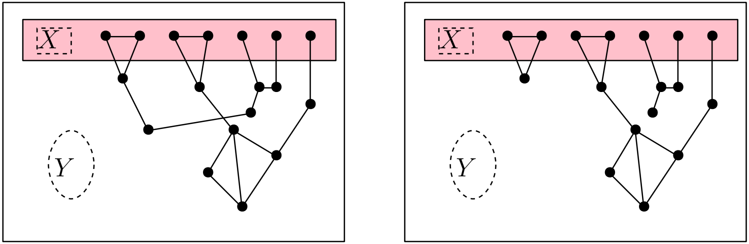
- For each non-trivial block of $\partial(G_t)\backslash X$, guess its final shape, and keep which neighborhoods appear in $G_t - \partial(G_t)$.
- Two partials solutions with the same information on blocks yields the same outcome.

Second step (works as for FVS)



- need to determine chordality.

Second step (works as for FVS)



- need to determine chordality.
- related to the partitions of connected components in $\partial(G_t) \setminus X$.
- Representative sets to avoid the blow-up to $2^{O(w \log w)}$.

Lower bound when $\mathcal{P}$ contains a long induced cycle

Permutation $k \times k$ Independent Set

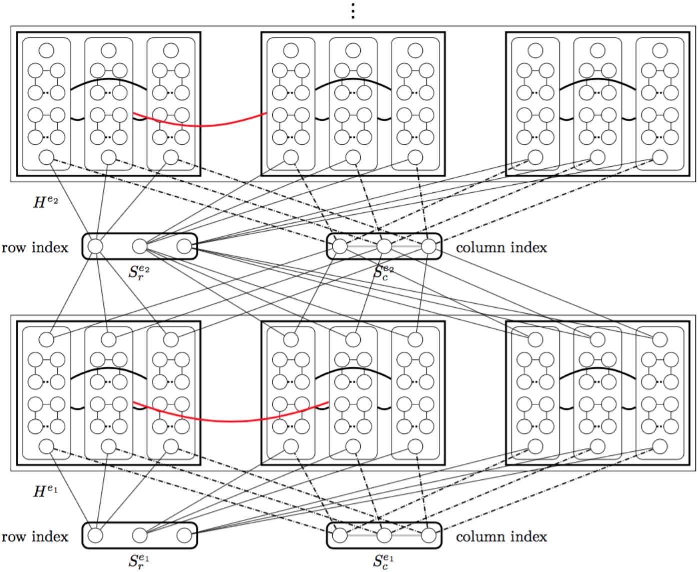
Given a graph $G = (\{1, \ldots, k\} \times \{1, \ldots, k\}, E)$, find an independent set of size k containing exactly one vertex per row and per column.

- Exponential Time Hypothesis (ETH) implies that n-variable 3-SAT cannot be solved in time $2^{o(n)}$ (Impagliazzo, Paturi, and Zane '01)

Lokshtanov, Marx, Saurabh '11

Unless ETH fails, Permutation $k \times k$ Independent Set cannot be solved in time $2^{o(k \log k)}$.

Reduction from Permutation $k \times k$ Independent Set.


⋮
H^{e_2}
row index
$S_r^{e_2}$
$S_c^{e_2}$
column index
H^{e_1}
row index
$S_r^{e_1}$
$S_c^{e_1}$
column index

Conclusion

- Can GFVS be solved in $2^{O(w \log w)} n^{O(1)}$?
- For $\mathcal{P}$ containing only chordal graphs can the dependency on d be improved?

Conclusion

- Can GFVS be solved in $2^{O(w \log w)} n^{O(1)}$?
- For $\mathcal{P}$ containing only chordal graphs can the dependency on d be improved?

Thank you for your attention!