ITERATIVE COMPRESSION

Feedback Vertex Set (FVS)

input: graph G, $k \in \mathbb{N}$
question: is there $S \subseteq V(G)$ $|S| \leq k$ such that $G \setminus S$ is a forest?

We define the following problem:

Compression FVS:

input: graph G, $k \in \mathbb{N}$, $S \subseteq V(G)$ a FVS of size $k+1$
question: is there a FVS $S' \subseteq V(G)$ of size k in G?

Theorem: if Compression-FVS can be solved in time $f(k)\, n^c$, then FVS can be solved in time $f(k)\, n^{c+1}$

Proof: Consider the following algorithm, assuming we have an algorithm for COMPRESSION-FVS:

Algo-FVS (G, k): let $V(G) = \{v_1, \dots, v_n\}$

- $S \leftarrow \{v_1, \dots, v_k\}$
- for i from $k+1$ to n:
  - $S \leftarrow S \cup \{v_i\}$
  - if compression-FVS(G, k, S) return no $\Rightarrow$ return no
  - otherwise, $S \leftarrow$ solution returned by compression-FVS(G, k, S)

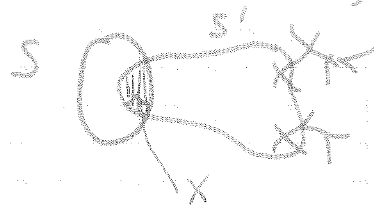
exercise: ~~the~~ check this algorithm does the job.

Now: Solving Compression-FVS:

let G, k, $S \subseteq V(G)$ FVS of size $k+1$.

idea: if there is a solution S' of compression-FVS, we would like to guess $S' \cap S$

$\Rightarrow$ we enumerate all subsets X of S (at most 2^{k+1} such sets),



"Sanity check": if $S \setminus X$ is not a FVS of $G \setminus X$: reject it

once X is guessed: we delete X from the graph. What remains to do is to find a set $Y \subseteq V(G) \setminus S$ of size at most $k - |X|$ which is a FVS of $G \setminus X$. We define:

Disjoint-FVS

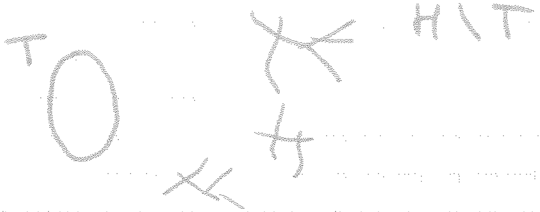
input: graph H, $q \in \mathbb{N}$, $T \subseteq V(H)$ of size $\leq q+1$ such that T is a FVS of H
question is there $Z \subseteq V(H)$: $|Z| \leq q$, $Z \cap T = \emptyset$ and Z is a FVS of H

We run an algorithm for Disjoint FVS with input $G \setminus X$, $S \setminus X$, $k - |X|$

theorem: if Disjoint-FVS can be solved in $f(q)|V(H)|^{O(1)}$, then FVS can be solved in $f'(k)\ |V(G)|^{O(1)}$ for some function f' depending on f. (in particular, if $f(q) = \alpha^q$, then $f'(k) = (\alpha+1)^k$) for some $\alpha \geq 1$

<u>Now</u>: a $4^q\ |V(H)|^{O(1)}$ algorithm for Disjoint-FVS.

let H, q, T be an instance



<u>RR1</u>: if there is a vertex of degree 1 in H, remove it

<u>RR2</u>: if there is $v \in V(H) \setminus T$ with two neighbors in a same connected component of T, take it in the solution (ie: remove v from H, decrease q by one).

<u>RR3</u>: if there is $v \in V(H) \setminus T$ of degree 2 in H and at least one neighbor in T, remove v and connect its two neighbors if they weren't connected.

<u>exercise</u>: prove that RR1, RR2 and RR3 are safe.

<u>Now</u>: let v be a <u>leaf</u> in $H \setminus T$.
we branch on taking v in the solution or not:

<u>Branch 1</u>: Recurse on $(H \setminus v,\ q-1,\ T)$

<u>Branch 2</u>: Recurse on $(H,\ q,\ T \cup \{v\})$

<u>Analysis</u>: in the first branch, q decreases by one.
in the second branch, q remains the same, but $T \cup \{v\}$ contains strictly less connected components than T.

$\Rightarrow$ For an instance $I = (H, q, T)$, we define the measure $\mu(I) = q + \text{number of c.c.}(T)$
in each branch, $\mu(I)$ decreases by 1, and $\mu(I) \leq 2q+1$.

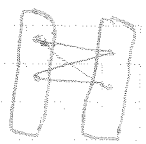
$\Rightarrow$ branching tree with width 2 and height $\leq 2q+1$ $\Rightarrow$ the algorithm runs in time $O(4^q . |V(H)|^{O(1)})$.

Odd Cycle Transversal OCT

input: G graph, $k \in \mathbb{N}$

question: is there a set $S \subseteq V(G)$

$|S| \leq k$ such that $G \setminus S$ is bipartite.



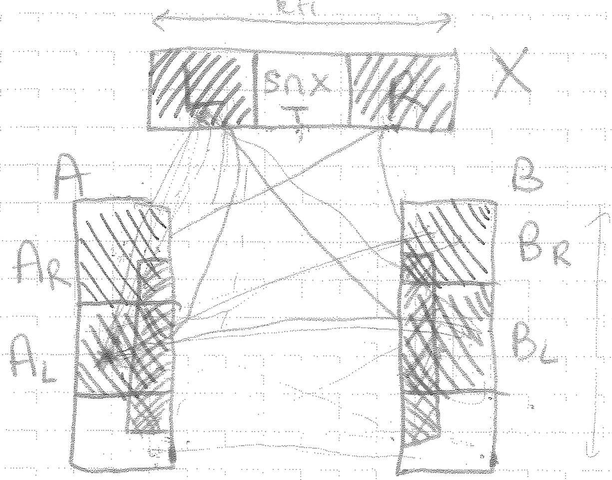
bipartition $A \uplus B$

Compression OCT

input: graph G, $X \subseteq V(G)$ an OCT of size $\leq k+1$

question: is there an OCT S of size $\leq k$?

bipartition extending R and L



1st step: "guess" the shape of the solution in X. $\Rightarrow$ enumerate all tripartitions of $X = L \uplus T \uplus R$

"sanity checks": • no edge in L, no edge in R

• if there is $v \in A \cup B$ adjacent to a vertex of L and a vertex of R $\Rightarrow$ remove it, decrease k by one.

$\rightarrow$ Define the candidates for extending L and R

A_R: neighbors of R in A

A_L: ——— L ———

B_R: neighbors of R in B

B_L: ——— L in B

bipartition L^+ and R^+

$L \subseteq L^+$ and $R \subseteq R^+$

solution $|Y| \leq k$

Goal: find $Y \subseteq A \cup B$ such that $T \cup Y$ is an OCT ~~such that~~ extending L and R

Lemma: Y is a solution iff it is a cut in between $(A_R \cup B_L)$ and $(A_L \cup B_R)$ in $G[A \cup B]$

set of vertices s.t. after removing them there is no path between ...

Proof:

$\Rightarrow$) Suppose it is not a cut:

• A_R and A_L: $\Rightarrow$ it must be of even lenght in $G[A \cup B]$ but then there would be a path between R^+ and L^+ of even lenght $\Rightarrow$ impossible.

if there is a path $A_R - B_R$ it must be of odd length, but then we have a $R^+ - R^+$ path of odd lenght

$\Rightarrow$ impossible

$\Leftarrow$: suppose Y is a cut between (...) in $A \cup B$. Suppose it is not a solution,

$\Rightarrow$ there is an odd cycle C in $G \setminus (Y \cup T)$.

- C cannot lie entirely in $A \cup B$ (because $G[A \cup B]$ is bipartite).
- $C \cap X \neq \emptyset$. let $v_1, v_2, \dots, v_q$ be the vertices of $C \cap X$ in the order of C L L R L R R R L R L R

observation:

- L-L: even lenght
- R-R: even lenght
- R-L: odd lenght
- L-R: odd lenght

$\left.\begin{matrix} \text{L-L: even lenght} \\ \text{R-R: even lenght} \end{matrix}\right] \text{even}$, $\left.\begin{matrix} \text{R-L: odd lenght} \\ \text{L-R: odd lenght} \end{matrix}\right] \text{even}$; even $\Rightarrow$ impossible.

[B. Reed, Smith, A. Vetta, 2004]

"Simpler parameterized Algorithm for OCT" by Lokshtanov, Saurabh, Sikdar. 2009

FVS Cygan et al. Book Parameterized Algorithms

length

Clement

Zoe Mathis