Parameterized Algorithms for Cycle Hitting Problems

M. Tech. Thesis Presentation

Akshay Kumar

Department of Computer Science & Engineering
IIT Kanpur

July 22, 2015

Table of Contents

Definition

Decision problem

- Problem size, n
- Find a k–sized solution

Running time $= n^{\mathcal{O}(1)} f(k)$

Cycle hitting problems

Delete vertices to kill all cycles with 'certain' properties.

Delete minimum number of vertices such that:

FEEDBACK VERTEX SET — All cycles are killed (Forest)

ODD CYCLE TRANSVERSAL — All odd cycles are killed (Bipartite)

EVEN CYCLE TRANSVERSAL — All even cycles are killed

Decision Problem

Given a graph G, find a set of k vertices whose removal kills all the cycles/odd cycles/even cycles in the graph?

Results

Problem	$f(k)$ Best known	$f(k)$ Thesis	Remarks
FVS	$\mathcal{O}^*(3.619^k)$	$\mathcal{O}^*(5^k)$	Deterministic
	$\mathcal{O}^*(3^k)$	–	Randomized
OCT	$\mathcal{O}^*(2.31^k)$	–	Randomized
ECT	$\mathcal{O}^*(50^k)$	$\mathcal{O}^*(17^k)$	Deterministic

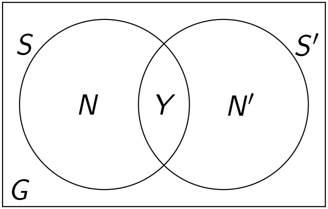
- Simplifies the analysis of FVS algorithm presented by Chen et. al.
- Improves the running time of ECT.

Idea

Iterative Compressive-FVS(IC-FVS)

Compress the size of *fvs* by one

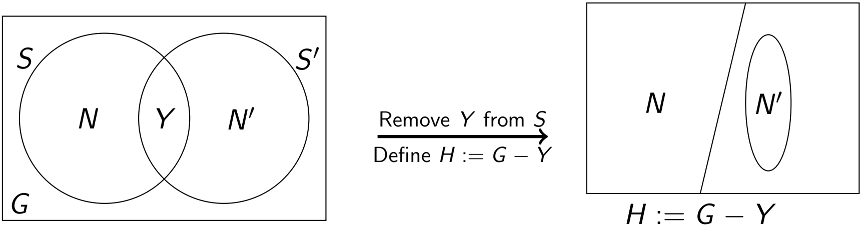
- Solves FEEDBACK VERTEX SET .
  - Base case: trivial *fvs* of graph induced over $k+1$ vertices.
  - Induction step: Contract *fvs* and add a vertex.



- $|S| = k+1$
- $|S'| = k$
- Both N and N' are forests.

- **Idea:** Consider all partitions of S into (N, Y) and build N'.

Idea
Disjoint-FVS



- $H[N]$ and $H - N$ are forests.
- N' is $k - |Y|$ sized *fvs* of H.
- Disjoint-FVS: $\mathcal{O}^*(c^k) \implies$ IC-FVS: $\mathcal{O}^*((c+1)^k)$.
  - Summing over all subsets Y of S: $\sum_{Y \subseteq S} c^{k-|Y|} = (c+1)^k$.

Feedback Vertex Set

Multicut on Trees

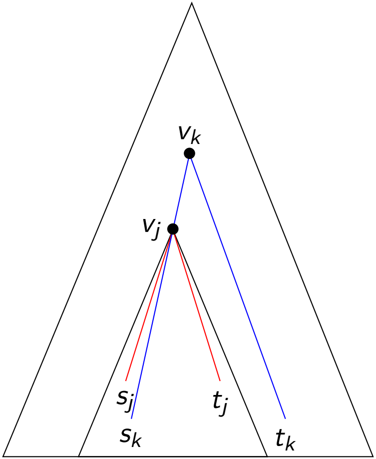
Multicut on Trees

Given: Tree T and $\{(s_i, t_i)\}_{i=1}^{n}$.
Disconnect them by deleting minimum set of vertices.

- v_i : Least Common Ancestor of (s_i, t_i)
- v_j : Maximum depth v_i over all v_i's

Claim

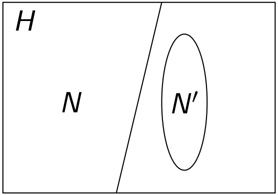
There exists an optimal solution containing v_j.



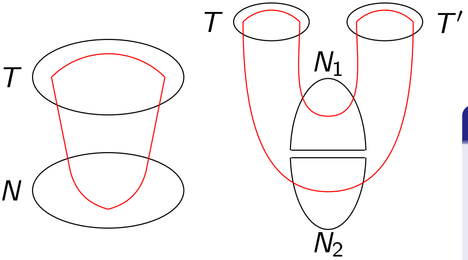
Feedback Vertex Set

DISJOINT-FVS

$H[N]$ and $H - N$ are forests. Find a k–sized *fvs* $N' \subseteq V(H - N)$ of H.



Types of cycles:



T, T' : Components of $H - N$

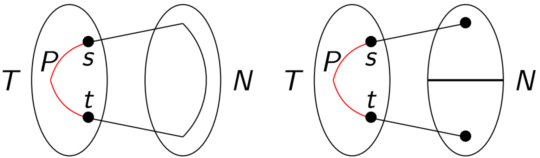
N_1, N_2 : Components of N

Ensure:

- No cycle in $T \cup N$.
- Edges from T incident in same component of N.

Feedback Vertex Set

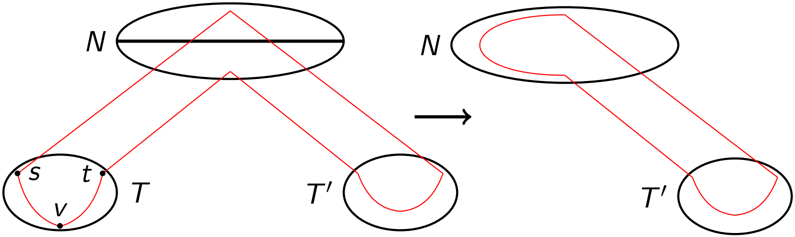
Branching terminal pair and branching path



P : Path joining s to t

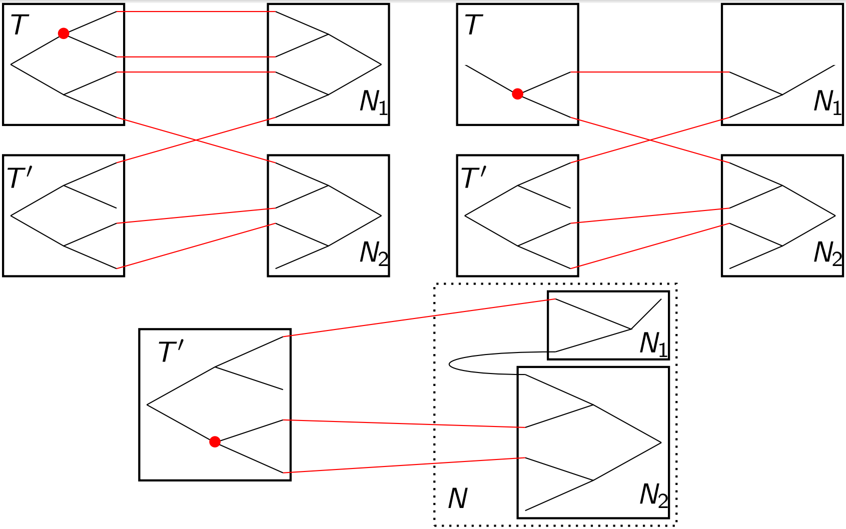
"Branching path"

Aim: Disconnect all branching terminal pairs.



Feedback Vertex Set

Example



Feedback Vertex Set
Time Complexity

Time complexity:

$$\mu = k + \text{ number of connected components of } N$$

- Initially, $\mu \leq 2k + 1$.
- μ decreases by one in every step.
- Branching factor $= 2$.

Hence, time complexity $= \mathcal{O}^*(2^{2k}) = \mathcal{O}^*(4^k)$.

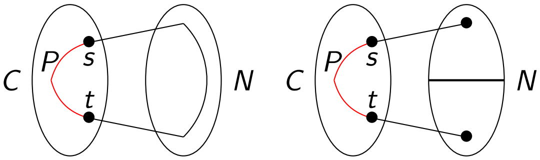
Time complexity of FVS $= \mathcal{O}^*(5^k)$.

Even Cycle Transversal

- Builds on Feedback Vertex Set solution.
- Branching path in FVS $\rightarrow$ Branching path in ECT.

Branching path

A path P in C (a component of $H - N$) is a branching path if $P \cup N$ has an even cycle in which P is a subpath or P has adjacencies to multiple connected components of N.



P : Path joining s to t

"Branching path"

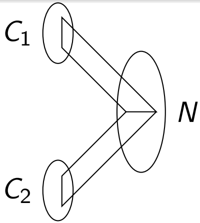
Aim: Disconnect all branching terminal pairs.

No branching path ensures:

- No even cycle in $C \cup N$.
- Edges from C incident in same component of N.

Even Cycle Transversal

Remaining even cycles



Even cycles still remaining.

- Cycle in $C_1 \cup N$ and $C_2 \cup N$: Odd
- Cycle in $C_1 \cup C_2 \cup N$: Even

Fact: At most two edges from each connected component.

Theorem

There exists an optimal solution which contains only one vertex (which has a neighbor in N) from each component C of $H - N$.

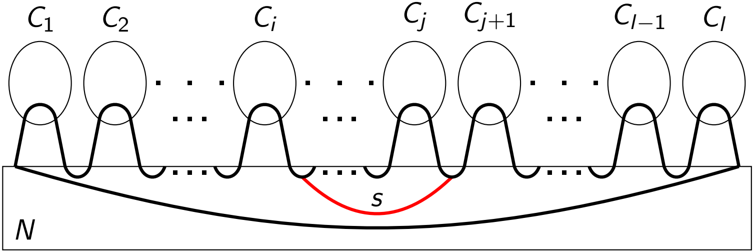
Claim: Only look at even cycles containing two components.

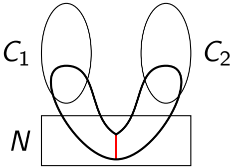
- Kills all the even cycles.
- Two way branching algorithm.

Even Cycle Transversal

Only look at even cycles containing two components of $H - N$

- Ensures all even cycles are killed.





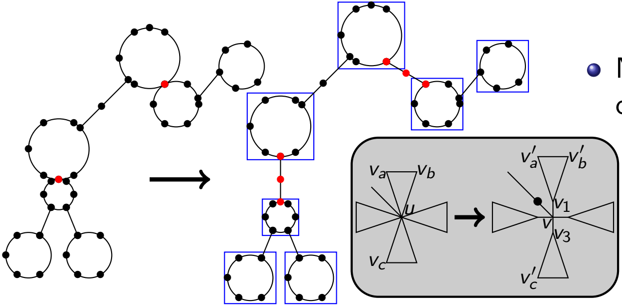
- Cycle containing ≥ 3 C_i's $\implies$ Cycle containing 2 C_i's.
- Cycle containing 2 C_i's $\implies$ Even cycle containing 2 C_i's.

Even Cycle Transversal
Odd Cactus

Cactus graph: Two cycles have at most one vertex in common.

Theorem

If S is an ect of G, then $G - S$ is an odd cactus.



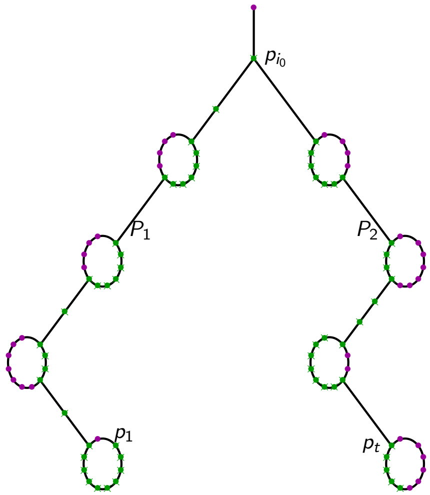
- No red nodes — simple cactus.

$v_a u v_b \rightarrow v'_a v_1 v'_b$

$v_a u v_c \rightarrow v'_a v_1 v v_3 v'_c$

Branching path in the tree

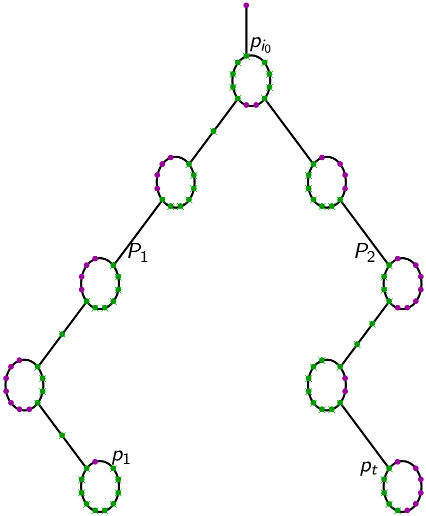
Upper branching path - I



- $P = p_1 p_2 \dots p_t$
- Decompose P
  - $P \equiv p_1 P_1 p_{i_0} P_2 p_t$

Branching path in the tree

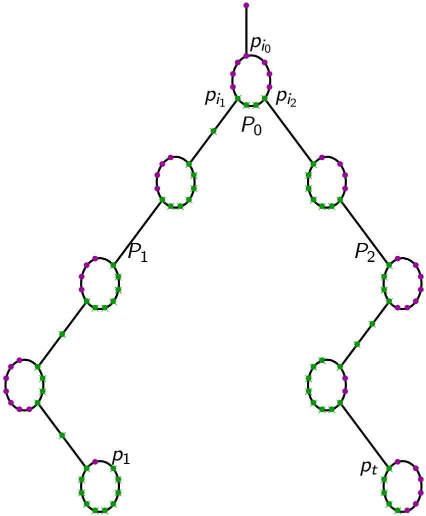
Upper branching path - II



- $P = p_1 p_2 \dots p_t$
- Decompose P
  - $P \equiv p_1 P_1 p_{i_0} P_2 p_t$

Branching path in the tree

Lower branching path

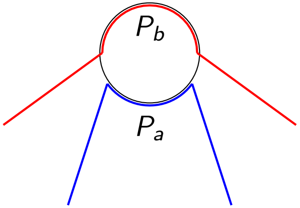


- $P = p_1 p_2 \dots p_t$
- Decompose P
  - $P \equiv p_1 P_1 p_{i_1} P_0 p_{i_2} P_2 p_t$

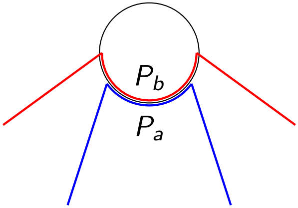
Partial order on branching paths

Two branching paths: P_a and P_b

- Top most node of P_a lies strictly in the subtree rooted at top most node of P_b $\implies$ $P_b \lhd P_a$.
- Both have the same top most node:



$P_b \lhd P_a$

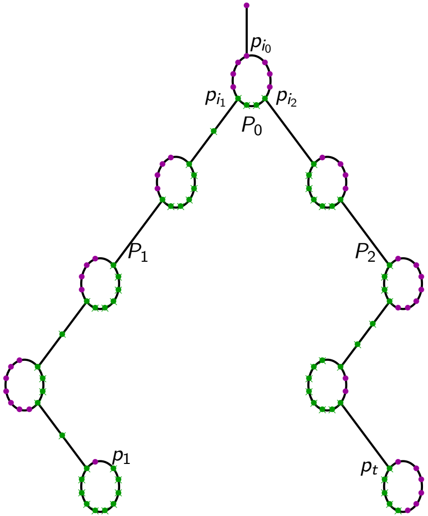


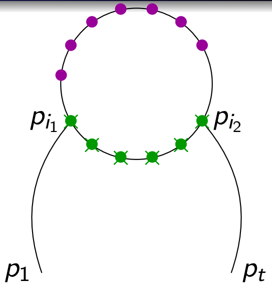
$P_b \lhd P_a$

- **Idea:** The further lower down a branching path is, the "higher" it is.

Maximal lower branching path

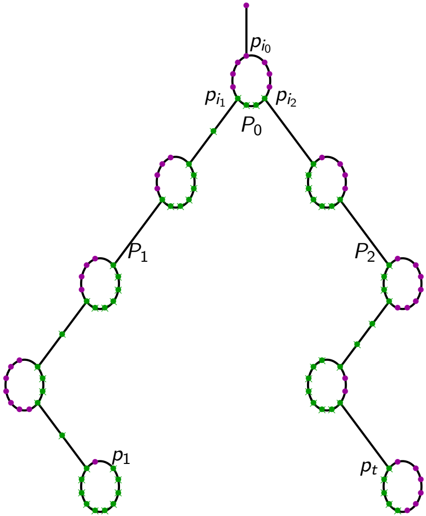
Properties

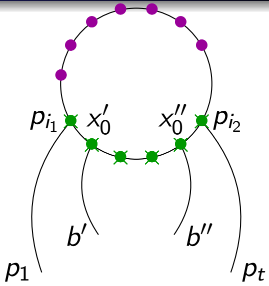




Maximal lower branching path

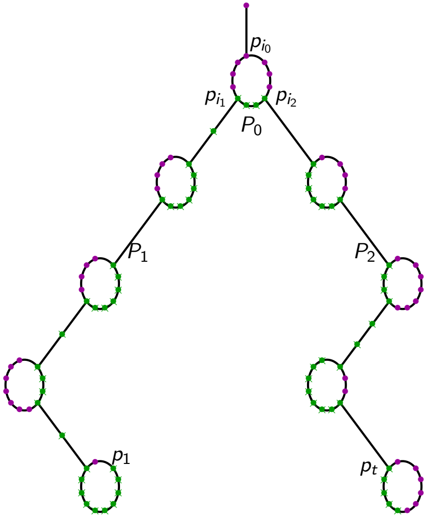
Properties

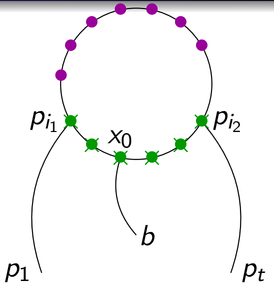




Maximal lower branching path

Properties





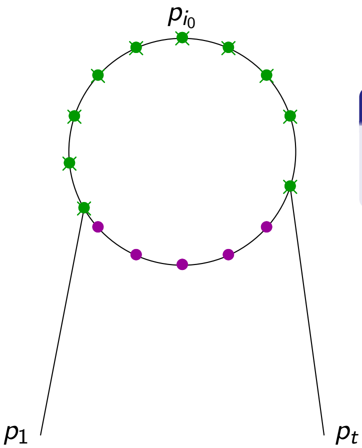
Theorem

There exists an optimal solution S_0 containing only p_{i_1}, p_{i_2} or x_0 from P.

- Move $p_{i_1}/p_{i_2}/x_0$ to S_0.
- Move P to N.

Maximal upper braching path

Properties



Theorem

There exists an optimal solution S_0 containing only p_{i_0} from P.

- Move p_{i_0} to S_0.
- Move P to N.

Even Cycle Transversal
Time Complexity

$$\mu = k + \text{ number of connected component of } N$$

- Initially, $\mu \leq 2k + 1$.
- μ decreases by one in every step.
- Overall branching factor $= 4$.
    - First phase: 4
    - Second phase: 2

Hence, time complexity $= \mathcal{O}^*(4^{2k}) = \mathcal{O}^*(16^k)$.

Time complexity of ECT $= \mathcal{O}^*(17^k)$.

Thank You!

Questions?