MaxMin Separation Problems: FPT Algorithms for st-Separator and Odd Cycle Transversal

Ajinkya Gaikwad

Joint work with H. Kumar, S. Maity, S. Saurabh, R. Sharma

Indian Institute of Science Education and Research, Pune, India

STACS 2025

Minimal s-t Separator and Minimal Odd Cycle Transversal

Definition

Given a connected graph $G = (V, E)$ and two vertices $s, t \in V$, a set $S \subseteq V(G)$ is called an s-t *separator* if there is no path between s and t in $G - S$.

Minimal s-t Separator and Minimal Odd Cycle Transversal

Definition

Given a connected graph $G = (V, E)$ and two vertices $s, t \in V$, a set $S \subseteq V(G)$ is called an s-t *separator* if there is no path between s and t in $G - S$.

Definition

An s-t separator is called a *minimal* s-t *separator* in G if no proper subset of S also separates s and t.

Minimal s-t Separator and Minimal Odd Cycle Transversal

Definition

Given a connected graph $G = (V, E)$ and two vertices $s, t \in V$, a set $S \subseteq V(G)$ is called an s-t *separator* if there is no path between s and t in $G - S$.

Definition

An s-t separator is called a *minimal* s-t *separator* in G if no proper subset of S also separates s and t.

Definition

Given a graph $G = (V, E)$, a set $S \subseteq V(G)$ is called an *Odd Cycle Transversal* if $G - S$ is a bipartite graph.

Minimal s-t Separator and Minimal Odd Cycle Transversal

Definition

Given a connected graph $G = (V, E)$ and two vertices $s, t \in V$, a set $S \subseteq V(G)$ is called an s-t *separator* if there is no path between s and t in $G - S$.

Definition

An s-t separator is called a *minimal* s-t *separator* in G if no proper subset of S also separates s and t.

Definition

Given a graph $G = (V, E)$, a set $S \subseteq V(G)$ is called an *Odd Cycle Transversal* if $G - S$ is a bipartite graph.

Definition

A set $S \subseteq V(G)$ is called a *minimal Odd Cycle Transversal* if:

1. $G - S$ is a bipartite graph,
2. No proper subset of S is an Odd Cycle Transversal.

Computational Problems

MAXIMUM MINIMAL st-SEPARATOR
Input: An undirected graph $G = (V, E)$, two distinct vertices s and t and a positive integer k.
Question: Determine whether there exists a minimal st-separator in G of size at least k.

Computational Problems

MAXIMUM MINIMAL st-SEPARATOR
Input: An undirected graph $G = (V, E)$, two distinct vertices s and t and a positive integer k.
Question: Determine whether there exists a minimal st-separator in G of size at least k.

MAXIMUM MINIMAL ODD CYCLE TRANSVERSAL
Input: An undirected graph $G = (V, E)$, and a positive integer k.
Question: Determine whether there exists a minimal odd cycle transversal in G of size at least k.

Computational Problems

MAXIMUM MINIMAL st-SEPARATOR
Input: An undirected graph $G = (V, E)$, two distinct vertices s and t and a positive integer k.
Question: Determine whether there exists a minimal st-separator in G of size at least k.

MAXIMUM MINIMAL ODD CYCLE TRANSVERSAL
Input: An undirected graph $G = (V, E)$, and a positive integer k.
Question: Determine whether there exists a minimal odd cycle transversal in G of size at least k.

Note: Both the MAXIMUM MINIMAL st-SEPARATOR and the MAXIMUM MINIMAL ODD CYCLE TRANSVERSAL problem are NP-hard.

Our Results and Main Technique

Theorem

MAXIMUM MINIMAL st-SEPARATOR *and* MAXIMUM MINIMAL OCT *parameterized by* k *are FPT.*

Our Results and Main Technique

Theorem

Maximum Minimal st-Separator *and* Maximum Minimal OCT *parameterized by* k *are FPT.*

To prove these results, we rely on the following meta-result of Lokshatonov, Ramanujan, Saurabh, and Zehavi [ICALP 2018].

Theorem

Let ψ *be a CMSO formula. For all* $k \in \mathbb{N}$*, there exists* $q \in \mathbb{N}$ *such that if there exists an algorithm that solves CMSO*$[\psi]$ *on* (q, k)*-unbreakable structures in time* $\mathcal{O}(n^d)$ *for some* $d > 4$*, then there exists an algorithm that solves CMSO*$[\psi]$ *on general structures in time* $\mathcal{O}(n^d)$*.*

Our Results and Main Technique

Theorem

MAXIMUM MINIMAL st-SEPARATOR *and* MAXIMUM MINIMAL OCT *parameterized by k are FPT.*

To prove these results, we rely on the following meta-result of Lokshatonov, Ramanujan, Saurabh, and Zehavi [ICALP 2018].

Theorem

Let ψ be a CMSO formula. For all $k \in \mathbb{N}$, there exists $q \in \mathbb{N}$ such that if there exists an algorithm that solves CMSO$[\psi]$ on (q, k)-unbreakable structures in time $\mathcal{O}(n^d)$ for some $d > 4$, then there exists an algorithm that solves CMSO$[\psi]$ on general structures in time $\mathcal{O}(n^d)$.

Application: If the following conditions are satisfied:

1. The problem can be expressed as a counting monadic second-order logic (CMSO) formula of length $f(k)$.
2. The problem is fixed-parameter tractable on (q, k)-unbreakable graphs parameterized by both q and k.

Then, the problem is fixed-parameter tractable (FPT) on general graphs.

Step 1

Lemma

Maximum Minimal st-Separator *is CMSO-definable with a formula of length $f(k)$.*

CMSO Formula:

$$\psi = \exists Z \subseteq V(G)\Bigg(\exists v_1, v_2, \ldots, v_k \in Z\Big(\bigwedge_{1 \le i < j \le k} v_i \neq v_j\Big)$$

$$\wedge \neg \exists U \subseteq V(G) \setminus Z\big((s \in U) \wedge (t \in U) \wedge \mathsf{conn}(U)\big)$$

$$\wedge \bigwedge_{i=1}^{k} \exists U \subseteq V(G) \setminus (Z \setminus \{v_i\})\big((s \in U) \wedge (t \in U) \wedge \mathsf{conn}(U)\big)\Bigg)$$

It is clear that the size of the above formula ψ depends only on k.

Step 2

Theorem

For positive integers $q, k \geq 1$, Maximum Minimal st-Separator *on* (q, k)*-unbreakable graphs on* n *vertices can be solved in time* $(k-1)^{2q} \cdot n^{\mathcal{O}(1)}$.

Step 2

Theorem

For positive integers $q, k \geq 1$, Maximum Minimal st-Separator *on (q, k)-unbreakable graphs on n vertices can be solved in time $(k-1)^{2q} \cdot n^{\mathcal{O}(1)}$.*

Theorem

Maximum Minimal st-Separator *is FPT when parameterized by k.*

FPT Algorithm for Maximum Minimal OCT

Step 1: CMSO formula for Maximum Minimal OCT

$$\varphi \equiv \exists Z \subseteq V(G) \left(\exists v_1, v_2, \ldots, v_k \in Z \left(\bigwedge_{1 \leq i < j \leq k} v_i \neq v_j \right) \right.$$
$$\wedge\, \textbf{bipartite}(V(G) \setminus Z)$$
$$\left. \wedge \left(\bigwedge_{i=1}^{k} \neg\textbf{bipartite}(V(G) \setminus (Z \setminus \{v_i\}) \right) \right)$$

where **bipartite**(W) is a CMSO sentence given below, which checks whether the graph induced by the vertices in W is bipartite.

Step 1: CMSO formula for MAXIMUM MINIMAL OCT

$$\varphi \equiv \exists Z \subseteq V(G) \left(\exists v_1, v_2, \ldots, v_k \in Z \left(\bigwedge_{1 \le i < j \le k} v_i \neq v_j \right) \right.$$
$$\wedge \textbf{bipartite}(V(G) \setminus Z)$$
$$\left. \wedge \left(\bigwedge_{i=1}^{k} \neg \textbf{bipartite}(V(G) \setminus (Z \setminus \{v_i\}) \right) \right)$$

where **bipartite**(W) is a CMSO sentence given below, which checks whether the graph induced by the vertices in W is bipartite.

$$\textbf{bipartite}(W) \equiv \exists X \subseteq W, \exists Y \subseteq W$$
$$\Big((X \cap Y = \emptyset) \wedge (X \cup Y = W)$$
$$\wedge \forall u, v \in W \; (E(u,v) \implies (u \in X \iff v \in Y)) \Big).$$

Step 2: FPT Algorithm on $(q, 2k)$-Unbreakable Graphs

We provide two sufficient conditions for the existence of a minimal OCT of size at least k on $(q, 2k)$-unbreakable graphs.

Step 2: FPT Algorithm on $(q, 2k)$-Unbreakable Graphs

We provide two sufficient conditions for the existence of a minimal OCT of size at least k on $(q, 2k)$-unbreakable graphs.

Condition 1: If there exists an induced odd cycle of length at least $2q + 2$ in G, then G has a minimal oct of size at least k.

Step 2: FPT Algorithm on $(q, 2k)$-Unbreakable Graphs

We provide two sufficient conditions for the existence of a minimal OCT of size at least k on $(q, 2k)$-unbreakable graphs.

Condition 1: If there exists an induced odd cycle of length at least $2q + 2$ in G, then G has a minimal oct of size at least k.

Condition 2: Let d be any positive integer. If there exists a family $\mathcal{F}$ containing distinct induced odd cycles of G of length at most d and $|\mathcal{F}| > d(d!)(k-1)^d$ then G has a minimal oct of size at least k.

Step 2: FPT Algorithm on $(q, 2k)$-Unbreakable Graphs

We provide two sufficient conditions for the existence of a minimal OCT of size at least k on $(q, 2k)$-unbreakable graphs.

Condition 1: If there exists an induced odd cycle of length at least $2q + 2$ in G, then G has a minimal oct of size at least k.

Condition 2: Let d be any positive integer. If there exists a family $\mathcal{F}$ containing distinct induced odd cycles of G of length at most d and $|\mathcal{F}| > d(d!)(k-1)^d$ then G has a minimal oct of size at least k.

Observation: Let (G, k) be an instance of $\textsc{Maximum minimal OCT}$. If a vertex $x \in V(G)$ does not participate in any induced odd cycle, then

$$(G, k) \text{ is equivalent to } (G - x, k).$$

Lemma (Combination Lemma)

Given a graph G, a vertex $x \in V(G)$, and positive integers d, k, there is an algorithm that runs in $(kd)^{\mathcal{O}(d)}.n^{\mathcal{O}(1)}$ time and correctly outputs one of the following:

1. *An induced odd cycle containing x.*
2. *An induced odd cycle of length at least d.*
3. *A family $\mathcal{F}$ of distinct induced odd cycles, each of length at most $d-1$, such that*
$$|\mathcal{F}| \geq d \cdot d! \cdot (k-1)^d.$$
4. *A determination that there is no induced odd cycle containing x in G.*

Lemma (Combination Lemma)

Given a graph G, a vertex $x \in V(G)$, and positive integers d, k, there is an algorithm that runs in $(kd)^{\mathcal{O}(d)}.n^{\mathcal{O}(1)}$ time and correctly outputs one of the following:

1. *An induced odd cycle containing x.*
2. *An induced odd cycle of length at least d.*
3. *A family $\mathcal{F}$ of distinct induced odd cycles, each of length at most $d-1$, such that*
$$|\mathcal{F}| \geq d \cdot d! \cdot (k-1)^d.$$
4. *A determination that there is no induced odd cycle containing x in G.*

- Run the above lemma for each vertex $x \in V(G)$ and set $d = 2q + 2$.

Lemma (Combination Lemma)

Given a graph G, a vertex $x \in V(G)$, and positive integers d, k, there is an algorithm that runs in $(kd)^{\mathcal{O}(d)}.n^{\mathcal{O}(1)}$ time and correctly outputs one of the following:

1. *An induced odd cycle containing x.*
2. *An induced odd cycle of length at least d.*
3. *A family $\mathcal{F}$ of distinct induced odd cycles, each of length at most $d-1$, such that*
$$|\mathcal{F}| \geq d \cdot d! \cdot (k-1)^d.$$
4. *A determination that there is no induced odd cycle containing x in G.*

- Run the above lemma for each vertex $x \in V(G)$ and set $d = 2q + 2$.
- Progress for each possibility:

Lemma (Combination Lemma)

Given a graph G, a vertex $x \in V(G)$, and positive integers d, k, there is an algorithm that runs in $(kd)^{\mathcal{O}(d)}.n^{\mathcal{O}(1)}$ time and correctly outputs one of the following:

1. *An induced odd cycle containing x.*
2. *An induced odd cycle of length at least d.*
3. *A family $\mathcal{F}$ of distinct induced odd cycles, each of length at most $d-1$, such that*
$$|\mathcal{F}| \geq d \cdot d! \cdot (k-1)^d.$$
4. *A determination that there is no induced odd cycle containing x in G.*

- Run the above lemma for each vertex $x \in V(G)$ and set $d = 2q + 2$.
- Progress for each possibility:
  - (1): Add this cycle of length at most $2q + 1$ to $\mathcal{F}$.

Lemma (Combination Lemma)

Given a graph G, a vertex $x \in V(G)$, and positive integers d, k, there is an algorithm that runs in $(kd)^{\mathcal{O}(d)}.n^{\mathcal{O}(1)}$ time and correctly outputs one of the following:

1. *An induced odd cycle containing x.*
2. *An induced odd cycle of length at least d.*
3. *A family $\mathcal{F}$ of distinct induced odd cycles, each of length at most $d-1$, such that*
$$|\mathcal{F}| \geq d \cdot d! \cdot (k-1)^d.$$
4. *A determination that there is no induced odd cycle containing x in G.*

- Run the above lemma for each vertex $x \in V(G)$ and set $d = 2q + 2$.
- Progress for each possibility:
  - (1): Add this cycle of length at most $2q + 1$ to $\mathcal{F}$.
  - (2) and (3): Directly imply a **yes** instance.

Lemma (Combination Lemma)

Given a graph G, a vertex $x \in V(G)$, and positive integers d, k, there is an algorithm that runs in $(kd)^{\mathcal{O}(d)}.n^{\mathcal{O}(1)}$ time and correctly outputs one of the following:

1. *An induced odd cycle containing x.*
2. *An induced odd cycle of length at least d.*
3. *A family $\mathcal{F}$ of distinct induced odd cycles, each of length at most $d-1$, such that*
$$|\mathcal{F}| \geq d \cdot d! \cdot (k-1)^d.$$
4. *A determination that there is no induced odd cycle containing x in G.*

- Run the above lemma for each vertex $x \in V(G)$ and set $d = 2q + 2$.
- Progress for each possibility:
  - (1): Add this cycle of length at most $2q + 1$ to $\mathcal{F}$.
  - (2) and (3): Directly imply a **yes** instance.
  - (4): Reduce the graph by deleting x.

Lemma

If the number of vertices in G is at least $(2q+2)^2(2q+2)!(k-1)^{2q+2}+1$, then G contains a minimal oct of size at least k.

Lemma

If the number of vertices in G is at least $(2q+2)^2(2q+2)!(k-1)^{2q+2}+1$, then G contains a minimal oct of size at least k.

Theorem

For any positive integers $q, k \geq 1$, $\textsc{Maximum Minimal OCT}$ on $(q, 2k)$-unbreakable graphs on n vertices can be solved in time $2^{(qk)^{\mathcal{O}(q)}} \cdot n^{\mathcal{O}(1)}$.

Lemma

If the number of vertices in G is at least $(2q+2)^2(2q+2)!(k-1)^{2q+2}+1$, then G contains a minimal oct of size at least k.

Theorem

For any positive integers $q, k \geq 1$, Maximum Minimal OCT *on $(q, 2k)$-unbreakable graphs on n vertices can be solved in time $2^{(qk)^{\mathcal{O}(q)}} \cdot n^{\mathcal{O}(1)}$.*

Theorem

Maximum Minimal Odd Cycle Transversal *is FPT when parameterized by k.*

Acknowledgment

Conference Travel Support

- **Infosys Travel Grant** – Supported by Infosys Foundation for attending STACS 2025.
- **IARCS Travel Grant** – Provided by the Indian Association for Research in Computing Science (IARCS) to support my travel for STACS 2025.

I sincerely appreciate the support from these organizations, which made my participation in this conference possible.

Thank You!

Questions?

Feel free to ask.