Monitoring edge-geodetic sets on oriented graphs

Tapas Das\textsuperscript{3}, Florent Foucaud\textsuperscript{1}, Pierre-Marie Marcille\textsuperscript{2}, Pavan PD\textsuperscript{3}, Sagnik Sen\textsuperscript{3}

LIMOS, Université Clermont Auvergne
LaBRI, Université de Bordeaux
Indian Institute of Technology Dharwad, India

Journées Graphes et Algorithmes 2023
Monitoring edge-geodetics

**Definition**

A monitoring edge-geodetic set, or MEG-set, of a graph $G$ is a vertex subset $M \subseteq V(G)$ such that given any edge $e$ of $G$, $e$ lies on every shortest $u$-$v$ path of $G$, for some $u, v \in M$. For a graph $G$, we denote $\text{meg}(G)$ the size of a smallest MEG-set of $G$. 

![Diagram of monitoring edge-geodetics](image)
Monitoring edge-geodetics

**Definition**
A monitoring edge-geodetic set, or MEG-set, of a graph $G$ is a vertex subset $M \subseteq V(G)$ such that given any edge $e$ of $G$, $e$ lies on every shortest $u$-$v$ path of $G$, for some $u, v \in M$. For a graph $G$, we denote $\text{meg}(G)$ the size of a smallest MEG-set of $G$.

**Link with networks**
This object simulates probes monitoring a network: if the value of the ping between two probes increases, then one can know where the failure happened.
Monitoring edge-geodetics

Definition
A monitoring edge-geodetic set, or MEG-set, of a graph $G$ is a vertex subset $M \subseteq V(G)$ such that given any edge $e$ of $G$, $e$ lies on every shortest $u$-$v$ path of $G$, for some $u, v \in M$. For a graph $G$, we denote $\text{meg}(G)$ the size of a smallest MEG-set of $G$.

Link with networks
This object simulates probes monitoring a network: if the value of the ping between two probes increases, then one can know where the failure happened.
Introduction

Monitoring edge-geodetics

**Definition**

A monitoring edge-geodetic set, or MEG-set, of a graph $G$ is a vertex subset $M \subseteq V(G)$ such that given any edge $e$ of $G$, $e$ lies on every shortest $u$-$v$ path of $G$, for some $u, v \in M$. For a graph $G$, we denote $\text{meg}(G)$ the size of a smallest MEG-set of $G$.

**Link with networks**

This object simulates probes monitoring a network: if the value of the ping between two probes increases, then one can know where the failure happened.
Monitoring edge-geodetics

**Definition**

A monitoring edge-geodetic set, or MEG-set, of a graph $G$ is a vertex subset $M \subseteq V(G)$ such that given any edge $e$ of $G$, $e$ lies on every shortest $u$-$v$ path of $G$, for some $u, v \in M$. For a graph $G$, we denote $\text{meg}(G)$ the size of a smallest MEG-set of $G$.

**Link with networks**

This object simulates probes monitoring a network: if the value of the ping between two probes increases, then one can know where the failure happened.
Introduction

Monitoring edge-geodetics

Definition

A monitoring edge-geodetic set, or MEG-set, of a graph $G$ is a vertex subset $M \subseteq V(G)$ such that given any edge $e$ of $G$, $e$ lies on every shortest $u$-$v$ path of $G$, for some $u, v \in M$. For a graph $G$, we denote $\text{meg}(G)$ the size of a smallest MEG-set of $G$.

Link with networks

This object simulates probes monitoring a network: if the value of the ping between two probes increases, then one can know where the failure happened.
**Definition**

A monitoring edge-geodetic set, or MEG-set, of a graph $G$ is a vertex subset $M \subseteq V(G)$ such that given any edge $e$ of $G$, $e$ lies on every shortest $u$-$v$ path of $G$, for some $u, v \in M$. For a graph $G$, we denote $\text{meg}(G)$ the size of a smallest MEG-set of $G$.

**Link with networks**

This object simulates probes monitoring a network: if the value of the ping between two probes increases, then one can know where the failure happened.
Definition

A monitoring edge-geodetic set, or MEG-set, of a graph $G$ is a vertex subset $M \subseteq V(G)$ such that given any edge $e$ of $G$, $e$ lies on every shortest $u$-$v$ path of $G$, for some $u, v \in M$. For a graph $G$, we denote $\text{meg}(G)$ the size of a smallest MEG-set of $G$.

Link with networks

This object simulates probes monitoring a network: if the value of the ping between two probes increases, then one can know where the failure happened.
Monitoring edge-geodetics

Definition

A monitoring edge-geodetic set, or MEG-set, of a graph $G$ is a vertex subset $M \subseteq V(G)$ such that given any edge $e$ of $G$, $e$ lies on every shortest $u$-$v$ path of $G$, for some $u, v \in M$. For a graph $G$, we denote $\text{meg}(G)$ the size of a smallest MEG-set of $G$.

Link with networks

This object simulates probes monitoring a network: if the value of the ping between two probes increases, then one can know where the failure happened.
A few results on the \textit{meg} parameter

**Theorem [Foucaud et al., 2023]**
- For all $n \geq 2$, $\text{meg}(P_n) = 2$;
A few results on the *meg* parameter

**Theorem [Foucaud et al., 2023]**

- For all $n \geq 2$, $\text{meg}(P_n) = 2$;
- for all $n \neq 4$, $\text{meg}(C_n) = 3$, and $\text{meg}(C_4) = 4$;
A few results on the **meg** parameter

**Theorem [Foucaud et al., 2023]**

- For all \( n \geq 2 \), \( \text{meg}(P_n) = 2 \);
- For all \( n \neq 4 \), \( \text{meg}(C_n) = 3 \), and \( \text{meg}(C_4) = 4 \);
- For all \( n \in \mathbb{N}^* \), \( \text{meg}(K_n) = n \);
A few results on the \textit{meg} parameter

\textbf{Theorem [Foucaud et al., 2023]}

- For all $n \geq 2$, $\text{meg}(P_n) = 2$;
- for all $n \neq 4$, $\text{meg}(C_n) = 3$, and $\text{meg}(C_4) = 4$;
- for all $n \in \mathbb{N}^*$, $\text{meg}(K_n) = n$;
- if $G$ is a tree, then $\text{meg}(G) = |\{u \in V(G), d(u) = 1\}|$. 
A few results on the *meg* parameter

**Theorem [Foucaud et al., 2023]**

- For all $n \geq 2$, $\text{meg}(P_n) = 2$;
- for all $n \neq 4$, $\text{meg}(C_n) = 3$, and $\text{meg}(C_4) = 4$;
- for all $n \in \mathbb{N}^*$, $\text{meg}(K_n) = n$;
- if $G$ is a tree, then $\text{meg}(G) = |\{u \in V(G), d(u) = 1\}|$.

**Theorem [Haslegrave, 2023]**

The decision problem of determining for a graph $G$ and a natural number $k$ whether $\text{meg}(G) \leq k$ is NP-complete.
We consider orientations of simple graphs, without digones.

**Definition**

In an oriented graph $\overrightarrow{G}$, two vertices $x$ and $y$ are said to **monitor** an arc $\overrightarrow{a}$ if $\overrightarrow{a}$ belongs to all oriented shortest paths from $x$ to $y$ or from $y$ to $x$.

**Definition**

A **monitoring arc-geodetic set**, or **MAG-set**, of an oriented graph $\overrightarrow{G}$ is a vertex subset $M \subseteq V(\overrightarrow{G})$ such that given any arc $\overrightarrow{a}$ of $A(\overrightarrow{G})$, $\overrightarrow{a}$ is monitored by $x, y$, for some $x, y \in M$. For an oriented graph $\overrightarrow{G}$, we denote $mag(\overrightarrow{G})$ the size of a smallest MAG-set of $\overrightarrow{G}$.
First results

First note that for an oriented graph $\vec{G}$, the relation between $\text{mag}(\vec{G})$ and $\text{meg}(G)$ is not clear:

\[ \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \bullet \]
First note that for an oriented graph $\vec{G}$, the relation between $mag(\vec{G})$ and $meg(G)$ is not clear:
First note that for an oriented graph $\vec{G}$, the relation between $\text{mag}(\vec{G})$ and $\text{meg}(G)$ is not clear:

![Diagram of oriented graph](image-url)
First note that for an oriented graph $\vec{G}$, the relation between $\text{mag}(\vec{G})$ and $\text{meg}(G)$ is not clear:
First results

First note that for an oriented graph $\vec{G}$, the relation between $\text{mag}(\vec{G})$ and $\text{meg}(G)$ is not clear:
First note that for an oriented graph $\overrightarrow{G}$, the relation between $\text{mag}(\overrightarrow{G})$ and $\text{meg}(G)$ is not clear:

![Diagram](image)

**Remark [Das et al., 2023+]**

Let $\overrightarrow{G}$ be an oriented graph, and $x \in V(\overrightarrow{G})$. If $x$ is either a source or a sink, then $x$ is in all MAG-set of $\overrightarrow{G}$. 
First results

First note that for an oriented graph $\vec{G}$, the relation between $mag(\vec{G})$ and $meg(G)$ is not clear:

![Graph diagram](image)

**Remark [Das et al., 2023+]**

Let $\vec{G}$ be an oriented graph, and $x \in V(\vec{G})$. If $x$ is either a source or a sink, then $x$ is in all MAG-set of $\vec{G}$.

**Theorem [Das et al., 2023+]**

Let $\vec{G}$ be an oriented tree. There is a unique minimal MAG-set to $\vec{G}$, and it is exactly the set of sources and sinks of $\vec{G}$. 
**Theorem [Das et al., 2023+]**

Let \( \vec{G} \) be an orientation of \( K_n \) for some \( n \in \mathbb{N}^* \). Then \( mag(\vec{G}) \in \{ n - 1, n \} \).

Since one can check in polynomial type if a set of vertices of \( \vec{G} \) is an MAG-set, we can now easily characterize all tournaments for this parameter.
Complexity of computing the MAG-set size

We consider the following decision problem:

MAG-SET problem

Instance: An oriented graph $\vec{G}$, an integer $k$.

Question: Does there exist an MAG-set of $\vec{G}$ of size $k$?
We consider the following decision problem:

**MAG-SET problem**

**Instance:** An oriented graph $\vec{G}$, an integer $k$.

**Question:** Does there exist an MAG-set of $\vec{G}$ of size $k$?

**Theorem [Das et al., 2023+]**

The MAG-SET problem is NP-complete.
The SETCOVER problem

We proceed with a reduction from the SETCOVER problem.

```
\[ S_0 \quad S_1 \quad S_2 \quad S_3 \quad S_4 \]
\[ X_0 \quad X_1 \quad X_2 \quad X_3 \quad X_4 \]
```

**SETCOVER Problem:**

**Instance:** A set \( \{ X_0, X_2, \cdots, X_n \} \), sets \( \{ S_0, S_1, \cdots, S_m \} \) such that \( \bigcup_{i=0}^{m} S_i = \{ X_0, X_2, \cdots, X_n \} \) and an integer \( k \).

**Question:** Does there exist a subcollection of at most \( k \) sets \( S_i \)'s such that their union is \( \{ X_0, X_2, \cdots, X_n \} \).
Our gadget for the reduction

From any instance of SETCOVER $I$, we can compute $G(I)$ an instance of MAG-SET. Assume we have $M$ an MAG-set of $G(I)$.

We now need to study the properties of any MAG-set of $G(I)$. 
Our gadget for the reduction

From any instance of SETCOVER $I$, we can compute $G(I)$ an instance of MAG-SET. Assume we have $M$ an MAG-set of $G(I)$.

We now need to study the properties of any MAG-set of $G(I)$.
Our gadget for the reduction

From any instance of SETCOVER $I$, we can compute $G(I)$ an instance of MAG-SET. Assume we have $M$ an MAG-set of $G(I)$.

We now need to study the properties of any MAG-set of $G(I)$.
Our gadget for the reduction

From any instance of SETCOVER $I$, we can compute $G(I)$ an instance of MAG-SET. Assume we have $M$ an MAG-set of $G(I)$.

We now need to study the properties of any MAG-set of $G(I)$.
Our gadget for the reduction

From any instance of SETCOVER $I$, we can compute $G(I)$ an instance of MAG-SET. Assume we have $M$ an MAG-set of $G(I)$.

We now need to study the properties of any MAG-set of $G(I)$. 
Our gadget for the reduction

From any instance of SETCOVER $I$, we can compute $G(I)$ an instance of MAG-SET. Assume we have $M$ an MAG-set of $G(I)$.

We now need to study the properties of any MAG-set of $G(I)$. 
Our gadget for the reduction

From any instance of SETCOVER $I$, we can compute $\overrightarrow{G(I)}$ an instance of MAG-SET. Assume we have $M$ an MAG-set of $\overrightarrow{G(I)}$.

We now need to study the properties of any MAG-set of $\overrightarrow{G(I)}$. 
Our gadget for the reduction

From any instance of SETCOVER $I$, we can compute $G(I)$ an instance of MAG-SET. Assume we have $M$ an MAG-set of $G(I)$.

We now need to study the properties of any MAG-set of $G(I)$. 
Our gadget for the reduction

From any instance of SETCOVER $I$, we can compute $G(I)$ an instance of MAG-SET. Assume we have $M$ an MAG-set of $G(I)$.

We now need to study the properties of any MAG-set of $G(I)$.
Our gadget for the reduction

From any instance of SETCOVER $I$, we can compute $G(I)$ an instance of MAG-SET. Assume we have $M$ an MAG-set of $G(I)$.

For every $X_i$, either $x_i$ or some $s_j$ with $X_i \in S_j$ is in $M$. 

\begin{center}
\begin{tikzpicture}
  \node (a0) at (0,0) [black] {$a_0$};
  \node (b0) at (1.5,0) [black] {$b_0$};
  \node (s0) at (0,-0.5) [black] {$s_0$};
  \node (s1) at (1,-0.5) [black] {$s_1$};
  \node (s2) at (2,-0.5) [black] {$s_2$};
  \node (s3) at (3,-0.5) [black] {$s_3$};
  \node (s4) at (4,-0.5) [black] {$s_4$};
  \node (x0) at (0,-1) [black] {$x_0$};
  \node (x1) at (1,-1) [black] {$x_1$};
  \node (x2) at (2,-1) [black] {$x_2$};
  \node (x3) at (3,-1) [black] {$x_3$};
  \node (x4) at (4,-1) [black] {$x_4$};
  \node (e0) at (0,-2) [black] {$e_0$};
  \node (e1) at (1,-2) [black] {$e_1$};
  \node (e2) at (2,-2) [black] {$e_2$};
  \node (e3) at (3,-2) [black] {$e_3$};
  \node (e4) at (4,-2) [black] {$e_4$};
  \node (f0) at (0,-2.5) [black] {$f_0$};
  \node (f1) at (1,-2.5) [black] {$f_1$};
  \node (f2) at (2,-2.5) [black] {$f_2$};
  \node (f3) at (3,-2.5) [black] {$f_3$};
  \node (f4) at (4,-2.5) [black] {$f_4$};

  \draw[->] (a0) to (s0);
  \draw[->] (b0) to (s0);
  \draw[->] (s0) to (x0);
  \draw[->] (s0) to (x1);
  \draw[->] (s0) to (x2);
  \draw[->] (s0) to (x3);
  \draw[->] (s0) to (x4);
  \draw[->] (s0) to (e0);
  \draw[->] (s0) to (f0);
  \draw[->] (s0) to (e1);
  \draw[->] (s0) to (f1);
  \draw[->] (s0) to (e2);
  \draw[->] (s0) to (f2);
  \draw[->] (s0) to (e3);
  \draw[->] (s0) to (f3);
  \draw[->] (s0) to (e4);
  \draw[->] (s0) to (f4);

  \draw[->] (s1) to (s2);
  \draw[->] (s1) to (s3);
  \draw[->] (s1) to (s4);
  \draw[->] (s1) to (x0);
  \draw[->] (s1) to (x1);
  \draw[->] (s1) to (x2);
  \draw[->] (s1) to (x3);
  \draw[->] (s1) to (x4);
  \draw[->] (s1) to (e0);
  \draw[->] (s1) to (f0);
  \draw[->] (s1) to (e1);
  \draw[->] (s1) to (f1);
  \draw[->] (s1) to (e2);
  \draw[->] (s1) to (f2);
  \draw[->] (s1) to (e3);
  \draw[->] (s1) to (f3);
  \draw[->] (s1) to (e4);
  \draw[->] (s1) to (f4);

  \draw[->] (s2) to (s3);
  \draw[->] (s2) to (s4);
  \draw[->] (s2) to (x0);
  \draw[->] (s2) to (x1);
  \draw[->] (s2) to (x2);
  \draw[->] (s2) to (x3);
  \draw[->] (s2) to (x4);
  \draw[->] (s2) to (e0);
  \draw[->] (s2) to (f0);
  \draw[->] (s2) to (e1);
  \draw[->] (s2) to (f1);
  \draw[->] (s2) to (e2);
  \draw[->] (s2) to (f2);
  \draw[->] (s2) to (e3);
  \draw[->] (s2) to (f3);
  \draw[->] (s2) to (e4);
  \draw[->] (s2) to (f4);

  \draw[->] (s3) to (s4);
  \draw[->] (s3) to (x0);
  \draw[->] (s3) to (x1);
  \draw[->] (s3) to (x2);
  \draw[->] (s3) to (x3);
  \draw[->] (s3) to (x4);
  \draw[->] (s3) to (e0);
  \draw[->] (s3) to (f0);
  \draw[->] (s3) to (e1);
  \draw[->] (s3) to (f1);
  \draw[->] (s3) to (e2);
  \draw[->] (s3) to (f2);
  \draw[->] (s3) to (e3);
  \draw[->] (s3) to (f3);
  \draw[->] (s3) to (e4);
  \draw[->] (s3) to (f4);

  \draw[->] (x0) to (e0);
  \draw[->] (x0) to (f0);
  \draw[->] (x1) to (e1);
  \draw[->] (x1) to (f1);
  \draw[->] (x2) to (e2);
  \draw[->] (x2) to (f2);
  \draw[->] (x3) to (e3);
  \draw[->] (x3) to (f3);
  \draw[->] (x4) to (e4);
  \draw[->] (x4) to (f4);

  \draw[->] (e0) to (f0);
  \draw[->] (e1) to (f1);
  \draw[->] (e2) to (f2);
  \draw[->] (e3) to (f3);
  \draw[->] (e4) to (f4);

\end{tikzpicture}
\end{center}
Our gadget for the reduction

From any instance of SETCOVER $I$, we can compute $G(I)$ an instance of MAG-SET. Assume we have $M$ an MAG-set of $G(I)$.

If $s_j \in M$ then we are done!
Our gadget for the reduction

From any instance of SETCOVER $I$, we can compute $G(I)$ an instance of MAG-SET. Assume we have $M$ an MAG-set of $G(I)$.

If some $x_i \in M$, then we remove it and add an arbitrary $s_j$ to $M$, with $X_i \in S_j$. 
Conclusion

We have proven the following results on oriented graphs:

<table>
<thead>
<tr>
<th></th>
<th>non-oriented</th>
<th>oriented</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trees</td>
<td>leaves</td>
<td>sources and sinks</td>
</tr>
<tr>
<td>Cycles</td>
<td>$3(4$ for $C_4)$</td>
<td>$2 \leq mag \leq n$</td>
</tr>
<tr>
<td>$K_n$</td>
<td>$n$</td>
<td>either $n - 1$ or $n$</td>
</tr>
<tr>
<td>Decision problem</td>
<td>NP-hard</td>
<td>NP-hard</td>
</tr>
</tbody>
</table>
Conclusion

We have proven the following results on oriented graphs:

<table>
<thead>
<tr>
<th></th>
<th>non-oriented</th>
<th>oriented</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trees</td>
<td>leaves</td>
<td>sources and sinks</td>
</tr>
<tr>
<td>Cycles</td>
<td>$3(4 \text{ for } C_4)$</td>
<td>$2 \leq \text{mag} \leq n$</td>
</tr>
<tr>
<td>$K_n$</td>
<td>$n$</td>
<td>either $n-1$ or $n$</td>
</tr>
<tr>
<td>Decision problem</td>
<td>NP-hard</td>
<td>NP-hard</td>
</tr>
</tbody>
</table>

A few perspectives:

- To follow up on the idea of networks, we can study the interaction of monitoring with local constraints on all subgraphs.
- Some other results have been proven for the non-oriented case and the bounds are not known in the oriented case.
- We proved that MAGSET is hard on DAG. One can also wonder if the MAGSET problem is still hard for simpler graph structures, like planar graphs.
We have proven the following results on oriented graphs:

<table>
<thead>
<tr>
<th></th>
<th>non-oriented</th>
<th>oriented</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trees</td>
<td>leaves</td>
<td>sources and sinks</td>
</tr>
<tr>
<td>Cycles</td>
<td>$3(4 \text{ for } C_4)$</td>
<td>$2 \leq \text{mag} \leq n$</td>
</tr>
<tr>
<td>$K_n$</td>
<td>$n$</td>
<td>either $n - 1$ or $n$</td>
</tr>
<tr>
<td>Decision problem</td>
<td>NP-hard</td>
<td>NP-hard</td>
</tr>
</tbody>
</table>

A few perspectives:

- To follow up on the idea of networks, we can study the interaction of monitoring with local constraints on all subgraphs.
- Some other results have been proven for the non-oriented case and the bounds are not known in the oriented case.
- We proved that MAGSET is hard on DAG. One can also wonder if the MAGSET problem is still hard for simpler graph structures, like planar graphs.

Thank you for your attention!