Block elimination distance

Giannos Stamoulis

Équipe AlGCo, LIRMM, Université de Montpellier, France.

Joint work with

Archontia C. Giannopoulou[1], **Dimitrios M. Thilikos**[2], and **Öznur Yaşar Diner**[3].

[1] Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, Athens, Greece.

[2] LIRMM, Université de Montpellier, CNRS, Montpellier, France.

[3] Computer Engineering Department, Kadir Has University, Istanbul, Turkey.

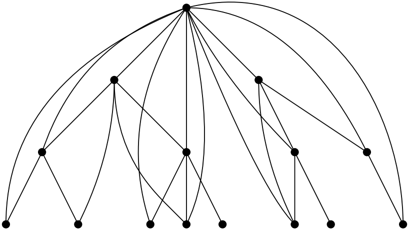
JGA2020, 16-18/11/2020 (en distanciel)

Treedepth

$\mathcal{C}(G) :=$ the set of connected components of G.

Definition

$$\mathrm{td}(G) = \begin{cases} \max\{\mathrm{td}(H) \mid H \in \mathcal{C}(G)\} & \text{if } |\mathcal{C}(G)| > 1 \text{ and } E(G) \neq \emptyset, \\ 1 + \min_{v \in V(G)}\{\mathrm{td}(G \setminus v)\} & \text{if } G \text{ is connected and } E(G) \neq \emptyset, \\ 1 & G \text{ is edgeless} \end{cases}$$

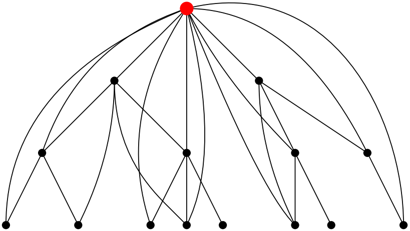


Treedepth

$\mathcal{C}(G) :=$ the set of connected components of G.

Definition

$$\mathrm{td}(G) = \begin{cases} \max\{\mathrm{td}(H) \mid H \in \mathcal{C}(G)\} & \text{if } |\mathcal{C}(G)| > 1 \text{ and } E(G) \neq \emptyset, \\ 1 + \min_{v \in V(G)}\{\mathrm{td}(G \setminus v)\} & \text{if } G \text{ is connected and } E(G) \neq \emptyset, \\ 1 & G \text{ is edgeless} \end{cases}$$

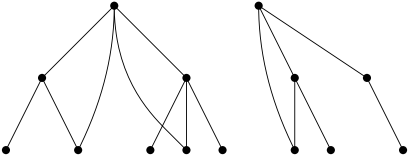


Treedepth

$\mathcal{C}(G) :=$ the set of connected components of G.

Definition

$$\mathrm{td}(G) = \begin{cases} \max\{\mathrm{td}(H) \mid H \in \mathcal{C}(G)\} & \text{if } |\mathcal{C}(G)| > 1 \text{ and } E(G) \neq \emptyset, \\ 1 + \min_{v \in V(G)}\{\mathrm{td}(G \setminus v)\} & \text{if } G \text{ is connected and } E(G) \neq \emptyset, \\ 1 & G \text{ is edgeless} \end{cases}$$

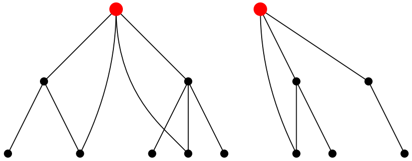


Treedepth

$\mathcal{C}(G) :=$ the set of connected components of G.

Definition

$$\mathrm{td}(G) = \begin{cases} \max\{\mathrm{td}(H) \mid H \in \mathcal{C}(G)\} & \text{if } |\mathcal{C}(G)| > 1 \text{ and } E(G) \neq \emptyset, \\ 1 + \min_{v \in V(G)}\{\mathrm{td}(G \setminus v)\} & \text{if } G \text{ is connected and } E(G) \neq \emptyset, \\ 1 & G \text{ is edgeless} \end{cases}$$

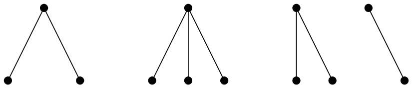


Treedepth

$\mathcal{C}(G) :=$ the set of connected components of G.

Definition

$$\mathrm{td}(G) = \begin{cases} \max\{\mathrm{td}(H) \mid H \in \mathcal{C}(G)\} & \text{if } |\mathcal{C}(G)| > 1 \text{ and } E(G) \neq \emptyset, \\ 1 + \min_{v \in V(G)}\{\mathrm{td}(G \setminus v)\} & \text{if } G \text{ is connected and } E(G) \neq \emptyset, \\ 1 & G \text{ is edgeless} \end{cases}$$

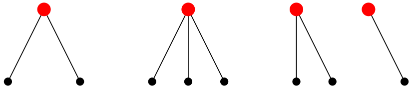


Treedepth

$\mathcal{C}(G) :=$ the set of connected components of G.

Definition

$$\mathrm{td}(G) = \begin{cases} \max\{\mathrm{td}(H) \mid H \in \mathcal{C}(G)\} & \text{if } |\mathcal{C}(G)| > 1 \text{ and } E(G) \neq \emptyset, \\ 1 + \min_{v \in V(G)}\{\mathrm{td}(G \setminus v)\} & \text{if } G \text{ is connected and } E(G) \neq \emptyset, \\ 1 & G \text{ is edgeless} \end{cases}$$



Treedepth

$\mathcal{C}(G) :=$ the set of connected components of G.

Definition

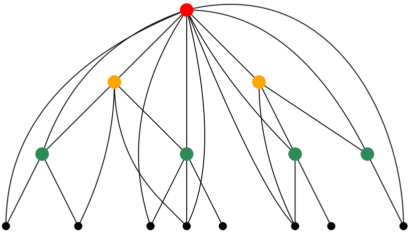
$$\mathrm{td}(G) = \begin{cases} \max\{\mathrm{td}(H) \mid H \in \mathcal{C}(G)\} & \text{if } |\mathcal{C}(G)| > 1 \text{ and } E(G) \neq \emptyset, \\ 1 + \min_{v \in V(G)}\{\mathrm{td}(G \setminus v)\} & \text{if } G \text{ is connected and } E(G) \neq \emptyset, \\ 1 & G \text{ is edgeless} \end{cases}$$

Treedepth

$\mathcal{C}(G) :=$ the set of connected components of G.

Definition

$$\mathrm{td}(G) = \begin{cases} \max\{\mathrm{td}(H) \mid H \in \mathcal{C}(G)\} & \text{if } |\mathcal{C}(G)| > 1 \text{ and } E(G) \neq \emptyset, \\ 1 + \min_{v \in V(G)}\{\mathrm{td}(G \setminus v)\} & \text{if } G \text{ is connected and } E(G) \neq \emptyset, \\ 1 & G \text{ is edgeless} \end{cases}$$



$\mathrm{td}(G) = 4$

Elimination distance to a graph class

Definition

$$\mathrm{ed}_{\mathcal{G}}(G) = \begin{cases} \max\{\mathrm{ed}_{\mathcal{G}}(H) \mid H \in \mathcal{C}(G)\} & \text{if } |\mathcal{C}(G)| > 1 \text{ and } G \notin \mathcal{G}, \\ 1 + \min_{v \in V(G)}\{\mathrm{ed}_{\mathcal{G}}(G \setminus v)\} & \text{if } G \text{ is connected and } G \notin \mathcal{G}, \\ 1 & G \in \mathcal{G} \end{cases}$$

Elimination distance to a graph class

Definition

$$\mathrm{ed}_{\mathcal{G}}(G) = \begin{cases} \max\{\mathrm{ed}_{\mathcal{G}}(H) \mid H \in \mathcal{C}(G)\} & \text{if } |\mathcal{C}(G)| > 1 \text{ and } G \notin \mathcal{G}, \\ 1 + \min_{v \in V(G)}\{\mathrm{ed}_{\mathcal{G}}(G \setminus v)\} & \text{if } G \text{ is connected and } G \notin \mathcal{G}, \\ 1 & G \in \mathcal{G} \end{cases}$$

$\mathrm{td}(G) = \mathrm{ed}_{\mathcal{E}}(G),$ where $\mathcal{E}$ is the class of edgeless graphs.

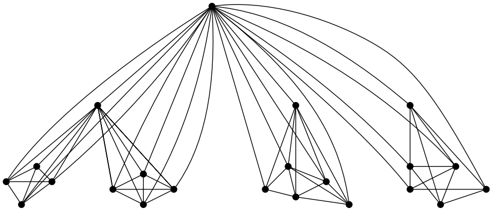
Elimination distance to a graph class

Definition

$$\mathrm{ed}_{\mathcal{G}}(G) = \begin{cases} \max\{\mathrm{ed}_{\mathcal{G}}(H) \mid H \in \mathcal{C}(G)\} & \text{if } |\mathcal{C}(G)| > 1 \text{ and } G \notin \mathcal{G}, \\ 1 + \min_{v \in V(G)}\{\mathrm{ed}_{\mathcal{G}}(G \setminus v)\} & \text{if } G \text{ is connected and } G \notin \mathcal{G}, \\ 1 & G \in \mathcal{G} \end{cases}$$

$$\mathrm{td}(G) = \mathrm{ed}_{\mathcal{E}}(G), \text{ where } \mathcal{E} \text{ is the class of edgeless graphs.}$$

Example for the class $\mathcal{P}$ of planar graphs:



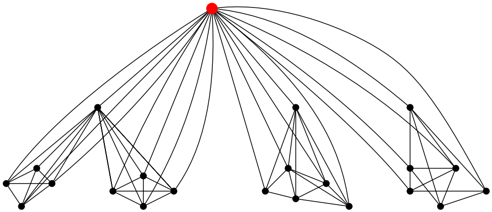
Elimination distance to a graph class

Definition

$$\mathrm{ed}_{\mathcal{G}}(G) = \begin{cases} \max\{\mathrm{ed}_{\mathcal{G}}(H) \mid H \in \mathcal{C}(G)\} & \text{if } |\mathcal{C}(G)| > 1 \text{ and } G \notin \mathcal{G}, \\ 1 + \min_{v \in V(G)}\{\mathrm{ed}_{\mathcal{G}}(G \setminus v)\} & \text{if } G \text{ is connected and } G \notin \mathcal{G}, \\ 1 & G \in \mathcal{G} \end{cases}$$

$$\mathrm{td}(G) = \mathrm{ed}_{\mathcal{E}}(G), \text{ where } \mathcal{E} \text{ is the class of edgeless graphs.}$$

Example for the class $\mathcal{P}$ of planar graphs:



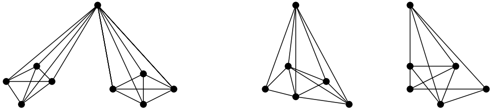
Elimination distance to a graph class

Definition

$$\mathrm{ed}_{\mathcal{G}}(G) = \begin{cases} \max\{\mathrm{ed}_{\mathcal{G}}(H) \mid H \in \mathcal{C}(G)\} & \text{if } |\mathcal{C}(G)| > 1 \text{ and } G \notin \mathcal{G}, \\ 1 + \min_{v \in V(G)}\{\mathrm{ed}_{\mathcal{G}}(G \setminus v)\} & \text{if } G \text{ is connected and } G \notin \mathcal{G}, \\ 1 & G \in \mathcal{G} \end{cases}$$

$$\mathrm{td}(G) = \mathrm{ed}_{\mathcal{E}}(G), \text{ where } \mathcal{E} \text{ is the class of edgeless graphs.}$$

Example for the class $\mathcal{P}$ of planar graphs:



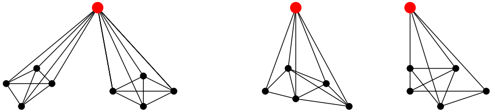
Elimination distance to a graph class

Definition

$$\mathrm{ed}_{\mathcal{G}}(G) = \begin{cases} \max\{\mathrm{ed}_{\mathcal{G}}(H) \mid H \in \mathcal{C}(G)\} & \text{if } |\mathcal{C}(G)| > 1 \text{ and } G \notin \mathcal{G}, \\ 1 + \min_{v \in V(G)}\{\mathrm{ed}_{\mathcal{G}}(G \setminus v)\} & \text{if } G \text{ is connected and } G \notin \mathcal{G}, \\ 1 & G \in \mathcal{G} \end{cases}$$

$$\mathrm{td}(G) = \mathrm{ed}_{\mathcal{E}}(G), \text{ where } \mathcal{E} \text{ is the class of edgeless graphs.}$$

Example for the class $\mathcal{P}$ of planar graphs:



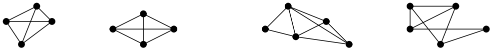
Elimination distance to a graph class

Definition

$$\mathrm{ed}_{\mathcal{G}}(G) = \begin{cases} \max\{\mathrm{ed}_{\mathcal{G}}(H) \mid H \in \mathcal{C}(G)\} & \text{if } |\mathcal{C}(G)| > 1 \text{ and } G \notin \mathcal{G}, \\ 1 + \min_{v \in V(G)}\{\mathrm{ed}_{\mathcal{G}}(G \setminus v)\} & \text{if } G \text{ is connected and } G \notin \mathcal{G}, \\ 1 & G \in \mathcal{G} \end{cases}$$

$$\mathrm{td}(G) = \mathrm{ed}_{\mathcal{E}}(G), \text{ where } \mathcal{E} \text{ is the class of edgeless graphs.}$$

Example for the class $\mathcal{P}$ of planar graphs:



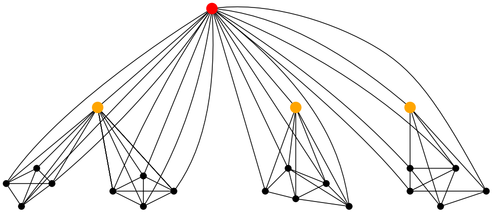
Elimination distance to a graph class

Definition

$$\mathrm{ed}_{\mathcal{G}}(G) = \begin{cases} \max\{\mathrm{ed}_{\mathcal{G}}(H) \mid H \in \mathcal{C}(G)\} & \text{if } |\mathcal{C}(G)| > 1 \text{ and } G \notin \mathcal{G}, \\ 1 + \min_{v \in V(G)}\{\mathrm{ed}_{\mathcal{G}}(G \setminus v)\} & \text{if } G \text{ is connected and } G \notin \mathcal{G}, \\ 1 & G \in \mathcal{G} \end{cases}$$

$$\mathrm{td}(G) = \mathrm{ed}_{\mathcal{E}}(G), \text{ where } \mathcal{E} \text{ is the class of edgeless graphs.}$$

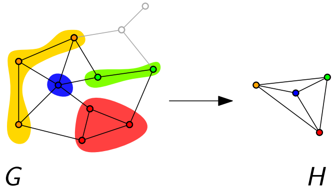
Example for the class $\mathcal{P}$ of planar graphs:



$\mathrm{ed}_{\mathcal{P}}(G) = 3$

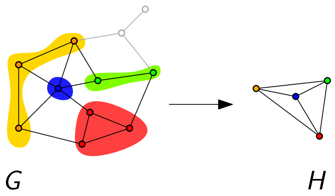
Minor-Closed Graph Classes

Minor-Closed Graph Classes



- H is a minor of G $\iff$
  H is obtained from a subgraph of G by contracting edges.

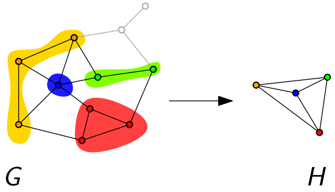
Minor-Closed Graph Classes



- H is a minor of G $\iff$
  H is obtained from a subgraph of G by contracting edges.

- $\mathcal{G}$ is *minor-closed* if $\forall G \in \mathcal{G}$ every minor H of G is in $\mathcal{G}$.

Minor-Closed Graph Classes

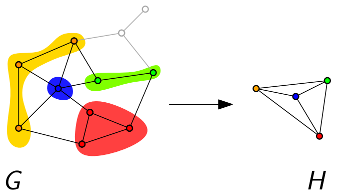


- H is a minor of G $\iff$
  H is obtained from a subgraph of G by contracting edges.

- $\mathcal{G}$ is *minor-closed* if $\forall G \in \mathcal{G}$ every minor H of G is in $\mathcal{G}$.

- obs($\mathcal{G}$):= the set of all *minor-minimal* graphs not in $\mathcal{G}$.

Minor-Closed Graph Classes



- H is a minor of $G \iff$

  H is obtained from a subgraph of G by contracting edges.

- $\mathcal{G}$ is *minor-closed* if $\forall G \in \mathcal{G}$ every minor H of G is in $\mathcal{G}$.

- obs($\mathcal{G}$):= the set of all *minor-minimal* graphs not in $\mathcal{G}$.

Robertson & Seymour Theorem

If $\mathcal{G}$ is minor-closed, then obs($\mathcal{G}$) is a finite set.

Distance from triviality

Distance from triviality

Vertex Deletion:

$\mathcal{A}_k(\mathcal{G})$:= the set of graphs that are k *vertices away* from $\mathcal{G}$.

Distance from triviality

Vertex Deletion:

$\mathcal{A}_k(\mathcal{G})$:= the set of graphs that are k *vertices away* from $\mathcal{G}$.

Theorem [Adler, Grohe, & Kreutzer, SODA 2008]

For every minor-closed $\mathcal{G}$, there is a computable function f that maps obs($\mathcal{G}$) to obs($\mathcal{A}_k(\mathcal{G})$).

Distance from triviality

Vertex Deletion:

$\mathcal{A}_k(\mathcal{G})$:= the set of graphs that are k *vertices away* from $\mathcal{G}$.

Theorem [Adler, Grohe, & Kreutzer, SODA 2008]

For every minor-closed $\mathcal{G}$, there is a computable function f that maps obs($\mathcal{G}$) to obs($\mathcal{A}_k(\mathcal{G})$).

Explicit bound on f [Sau, Stamoulis, & Thilikos, ICALP 2020].

Distance from triviality

Vertex Deletion:

$\mathcal{A}_k(\mathcal{G})$:= the set of graphs that are k *vertices away* from $\mathcal{G}$.

Theorem [Adler, Grohe, & Kreutzer, SODA 2008]

For every minor-closed $\mathcal{G}$, there is a computable function f that maps $\mathrm{obs}(\mathcal{G})$ to $\mathrm{obs}(\mathcal{A}_k(\mathcal{G}))$.

Explicit bound on f [Sau, Stamoulis, & Thilikos, ICALP 2020].

Elimination Distance:

$\mathcal{Z}_k(\mathcal{G}) := \{G \mid \mathrm{ed}_{\mathcal{G}}(G) \leq k\}$.

Distance from triviality

Vertex Deletion:

$\mathcal{A}_k(\mathcal{G})$:= the set of graphs that are k *vertices away* from $\mathcal{G}$.

Theorem [Adler, Grohe, & Kreutzer, SODA 2008]

For every minor-closed $\mathcal{G}$, there is a computable function f that maps $\mathrm{obs}(\mathcal{G})$ to $\mathrm{obs}(\mathcal{A}_k(\mathcal{G}))$.

Explicit bound on f [Sau, Stamoulis, & Thilikos, ICALP 2020].

Elimination Distance:

$\mathcal{Z}_k(\mathcal{G}) := \{G \mid \mathrm{ed}_{\mathcal{G}}(G) \leq k\}$.

Theorem [Bulian & Dawar, Algorithmica 2017]

For every minor-closed $\mathcal{G}$, there is a computable function f that maps $\mathrm{obs}(\mathcal{G})$ to $\mathrm{obs}(\mathcal{Z}_k(\mathcal{G}))$.

Distance from triviality

Vertex Deletion:

$\mathcal{A}_k(\mathcal{G})$:= the set of graphs that are k *vertices away* from $\mathcal{G}$.

Theorem [Adler, Grohe, & Kreutzer, SODA 2008]

For every minor-closed $\mathcal{G}$, there is a computable function f that maps obs($\mathcal{G}$) to obs($\mathcal{A}_k(\mathcal{G})$).

Explicit bound on f [Sau, Stamoulis, & Thilikos, ICALP 2020].

Elimination Distance:

$\mathcal{Z}_k(\mathcal{G}) := \{G \mid \mathrm{ed}_{\mathcal{G}}(G) \leq k\}$.

Theorem [Bulian & Dawar, Algorithmica 2017]

For every minor-closed $\mathcal{G}$, there is a computable function f that maps obs($\mathcal{G}$) to obs($\mathcal{Z}_k(\mathcal{G})$).

No explicit bound on f.

Block elimination distance

$\mathcal{B}(G)$:= the set of biconnected components of G.

Block elimination distance

$\mathcal{B}(G)$:= the set of biconnected components of G.

Definition

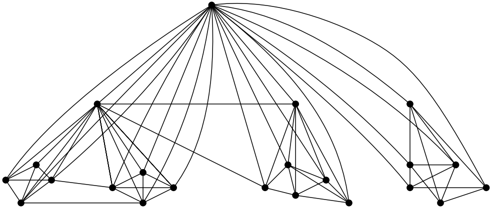
$$\mathrm{bed}_{\mathcal{G}}(G) = \begin{cases} \max\{\mathrm{bed}_{\mathcal{G}}(H) \mid H \in \mathcal{B}(G)\} & \text{if } |\mathcal{B}(G)| > 1 \text{ and } G \notin \mathcal{G}, \\ 1 + \min_{v \in V(G)}\{\mathrm{bed}_{\mathcal{G}}(G \setminus v)\} & \text{if } G \text{ is biconnected and } G \notin \mathcal{G}, \\ 1 & G \in \mathcal{G} \end{cases}$$

Block elimination distance

$\mathcal{B}(G)$:= the set of biconnected components of G.

Definition

$$\mathrm{bed}_{\mathcal{G}}(G) = \begin{cases} \max\{\mathrm{bed}_{\mathcal{G}}(H) \mid H \in \mathcal{B}(G)\} & \text{if } |\mathcal{B}(G)| > 1 \text{ and } G \notin \mathcal{G}, \\ 1 + \min_{v \in V(G)}\{\mathrm{bed}_{\mathcal{G}}(G \setminus v)\} & \text{if } G \text{ is biconnected and } G \notin \mathcal{G}, \\ 1 & G \in \mathcal{G} \end{cases}$$



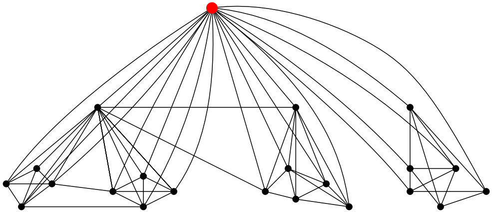
$\mathrm{ed}_{\mathcal{P}}(G) =$

Block elimination distance

$\mathcal{B}(G)$:= the set of biconnected components of G.

Definition

$$\mathrm{bed}_{\mathcal{G}}(G) = \begin{cases} \max\{\mathrm{bed}_{\mathcal{G}}(H) \mid H \in \mathcal{B}(G)\} & \text{if } |\mathcal{B}(G)| > 1 \text{ and } G \notin \mathcal{G}, \\ 1 + \min_{v \in V(G)}\{\mathrm{bed}_{\mathcal{G}}(G \setminus v)\} & \text{if } G \text{ is biconnected and } G \notin \mathcal{G}, \\ 1 & G \in \mathcal{G} \end{cases}$$



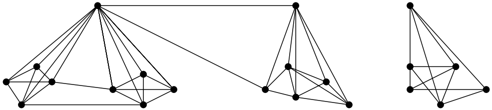
$\mathrm{ed}_{\mathcal{P}}(G) =$

Block elimination distance

$\mathcal{B}(G)$:= the set of biconnected components of G.

Definition

$$\mathrm{bed}_{\mathcal{G}}(G) = \begin{cases} \max\{\mathrm{bed}_{\mathcal{G}}(H) \mid H \in \mathcal{B}(G)\} & \text{if } |\mathcal{B}(G)| > 1 \text{ and } G \notin \mathcal{G}, \\ 1 + \min_{v \in V(G)}\{\mathrm{bed}_{\mathcal{G}}(G \setminus v)\} & \text{if } G \text{ is biconnected and } G \notin \mathcal{G}, \\ 1 & G \in \mathcal{G} \end{cases}$$



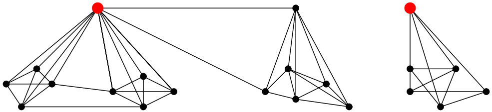
$\mathrm{ed}_{\mathcal{P}}(G) =$

Block elimination distance

$\mathcal{B}(G)$:= the set of biconnected components of G.

Definition

$$\mathrm{bed}_{\mathcal{G}}(G) = \begin{cases} \max\{\mathrm{bed}_{\mathcal{G}}(H) \mid H \in \mathcal{B}(G)\} & \text{if } |\mathcal{B}(G)| > 1 \text{ and } G \notin \mathcal{G}, \\ 1 + \min_{v \in V(G)}\{\mathrm{bed}_{\mathcal{G}}(G \setminus v)\} & \text{if } G \text{ is biconnected and } G \notin \mathcal{G}, \\ 1 & G \in \mathcal{G} \end{cases}$$



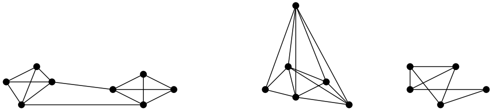
$\mathrm{ed}_{\mathcal{P}}(G) =$

Block elimination distance

$\mathcal{B}(G)$:= the set of biconnected components of G.

Definition

$$\mathrm{bed}_{\mathcal{G}}(G) = \begin{cases} \max\{\mathrm{bed}_{\mathcal{G}}(H) \mid H \in \mathcal{B}(G)\} & \text{if } |\mathcal{B}(G)| > 1 \text{ and } G \notin \mathcal{G}, \\ 1 + \min_{v \in V(G)}\{\mathrm{bed}_{\mathcal{G}}(G \setminus v)\} & \text{if } G \text{ is biconnected and } G \notin \mathcal{G}, \\ 1 & G \in \mathcal{G} \end{cases}$$



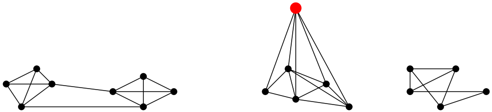
$\mathrm{ed}_{\mathcal{P}}(G) =$

Block elimination distance

$\mathcal{B}(G)$:= the set of biconnected components of G.

Definition

$$\mathrm{bed}_{\mathcal{G}}(G) = \begin{cases} \max\{\mathrm{bed}_{\mathcal{G}}(H) \mid H \in \mathcal{B}(G)\} & \text{if } |\mathcal{B}(G)| > 1 \text{ and } G \notin \mathcal{G}, \\ 1 + \min_{v \in V(G)}\{\mathrm{bed}_{\mathcal{G}}(G \setminus v)\} & \text{if } G \text{ is biconnected and } G \notin \mathcal{G}, \\ 1 & G \in \mathcal{G} \end{cases}$$



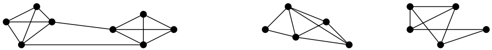
$\mathrm{ed}_{\mathcal{P}}(G) =$

Block elimination distance

$\mathcal{B}(G)$:= the set of biconnected components of G.

Definition

$$\mathrm{bed}_{\mathcal{G}}(G) = \begin{cases} \max\{\mathrm{bed}_{\mathcal{G}}(H) \mid H \in \mathcal{B}(G)\} & \text{if } |\mathcal{B}(G)| > 1 \text{ and } G \notin \mathcal{G}, \\ 1 + \min_{v \in V(G)}\{\mathrm{bed}_{\mathcal{G}}(G \setminus v)\} & \text{if } G \text{ is biconnected and } G \notin \mathcal{G}, \\ 1 & G \in \mathcal{G} \end{cases}$$



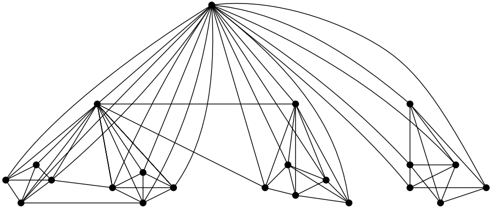
$\mathrm{ed}_{\mathcal{P}}(G) = 4$

Block elimination distance

$\mathcal{B}(G)$:= the set of biconnected components of G.

Definition

$$\mathrm{bed}_{\mathcal{G}}(G) = \begin{cases} \max\{\mathrm{bed}_{\mathcal{G}}(H) \mid H \in \mathcal{B}(G)\} & \text{if } |\mathcal{B}(G)| > 1 \text{ and } G \notin \mathcal{G}, \\ 1 + \min_{v \in V(G)}\{\mathrm{bed}_{\mathcal{G}}(G \setminus v)\} & \text{if } G \text{ is biconnected and } G \notin \mathcal{G}, \\ 1 & G \in \mathcal{G} \end{cases}$$



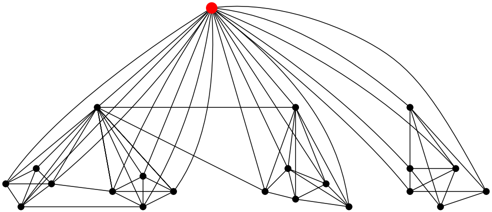
$\mathrm{ed}_{\mathcal{P}}(G) = 4$
$\mathrm{bed}_{\mathcal{P}}(G) =$

Block elimination distance

$\mathcal{B}(G)$:= the set of biconnected components of G.

Definition

$$\mathrm{bed}_{\mathcal{G}}(G) = \begin{cases} \max\{\mathrm{bed}_{\mathcal{G}}(H) \mid H \in \mathcal{B}(G)\} & \text{if } |\mathcal{B}(G)| > 1 \text{ and } G \notin \mathcal{G}, \\ 1 + \min_{v \in V(G)}\{\mathrm{bed}_{\mathcal{G}}(G \setminus v)\} & \text{if } G \text{ is biconnected and } G \notin \mathcal{G}, \\ 1 & G \in \mathcal{G} \end{cases}$$



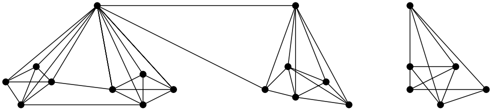
$\mathrm{ed}_{\mathcal{P}}(G) = 4$
$\mathrm{bed}_{\mathcal{P}}(G) =$

Block elimination distance

$\mathcal{B}(G)$:= the set of biconnected components of G.

Definition

$$\mathrm{bed}_{\mathcal{G}}(G) = \begin{cases} \max\{\mathrm{bed}_{\mathcal{G}}(H) \mid H \in \mathcal{B}(G)\} & \text{if } |\mathcal{B}(G)| > 1 \text{ and } G \notin \mathcal{G}, \\ 1 + \min_{v \in V(G)}\{\mathrm{bed}_{\mathcal{G}}(G \setminus v)\} & \text{if } G \text{ is biconnected and } G \notin \mathcal{G}, \\ 1 & G \in \mathcal{G} \end{cases}$$



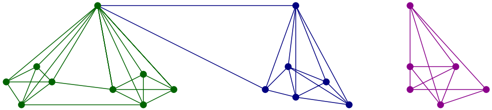
$\mathrm{ed}_{\mathcal{P}}(G) = 4$
$\mathrm{bed}_{\mathcal{P}}(G) =$

Block elimination distance

$\mathcal{B}(G)$:= the set of biconnected components of G.

Definition

$$\mathrm{bed}_{\mathcal{G}}(G) = \begin{cases} \max\{\mathrm{bed}_{\mathcal{G}}(H) \mid H \in \mathcal{B}(G)\} & \text{if } |\mathcal{B}(G)| > 1 \text{ and } G \notin \mathcal{G}, \\ 1 + \min_{v \in V(G)}\{\mathrm{bed}_{\mathcal{G}}(G \setminus v)\} & \text{if } G \text{ is biconnected and } G \notin \mathcal{G}, \\ 1 & G \in \mathcal{G} \end{cases}$$



$\mathrm{ed}_{\mathcal{P}}(G) = 4$

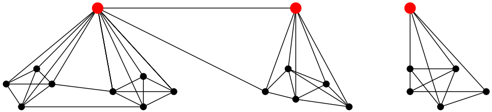
$\mathrm{bed}_{\mathcal{P}}(G) =$

Block elimination distance

$\mathcal{B}(G)$:= the set of biconnected components of G.

Definition

$$\mathrm{bed}_{\mathcal{G}}(G) = \begin{cases} \max\{\mathrm{bed}_{\mathcal{G}}(H) \mid H \in \mathcal{B}(G)\} & \text{if } |\mathcal{B}(G)| > 1 \text{ and } G \notin \mathcal{G}, \\ 1 + \min_{v \in V(G)}\{\mathrm{bed}_{\mathcal{G}}(G \setminus v)\} & \text{if } G \text{ is biconnected and } G \notin \mathcal{G}, \\ 1 & G \in \mathcal{G} \end{cases}$$



$\mathrm{ed}_{\mathcal{P}}(G) = 4$

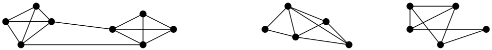
$\mathrm{bed}_{\mathcal{P}}(G) =$

Block elimination distance

$\mathcal{B}(G)$:= the set of biconnected components of G.

Definition

$$\mathrm{bed}_{\mathcal{G}}(G) = \begin{cases} \max\{\mathrm{bed}_{\mathcal{G}}(H) \mid H \in \mathcal{B}(G)\} & \text{if } |\mathcal{B}(G)| > 1 \text{ and } G \notin \mathcal{G}, \\ 1 + \min_{v \in V(G)}\{\mathrm{bed}_{\mathcal{G}}(G \setminus v)\} & \text{if } G \text{ is biconnected and } G \notin \mathcal{G}, \\ 1 & G \in \mathcal{G} \end{cases}$$



$\mathrm{ed}_{\mathcal{P}}(G) = 4$

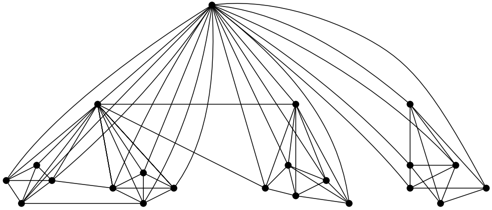
$\mathrm{bed}_{\mathcal{P}}(G) = 3$

Block elimination distance

$\mathcal{B}(G)$:= the set of biconnected components of G.

Definition

$$\mathrm{bed}_{\mathcal{G}}(G) = \begin{cases} \max\{\mathrm{bed}_{\mathcal{G}}(H) \mid H \in \mathcal{B}(G)\} & \text{if } |\mathcal{B}(G)| > 1 \text{ and } G \notin \mathcal{G}, \\ 1 + \min_{v \in V(G)}\{\mathrm{bed}_{\mathcal{G}}(G \setminus v)\} & \text{if } G \text{ is biconnected and } G \notin \mathcal{G}, \\ 1 & G \in \mathcal{G} \end{cases}$$



$\mathrm{ed}_{\mathcal{P}}(G) = 4$
$\mathrm{bed}_{\mathcal{P}}(G) = 3$

Our first result

$$\mathcal{G}^{(k)} := \{G \mid \mathrm{bed}_{\mathcal{G}}(G) \leq k\}.$$

Our first result

$\mathcal{G}^{(k)} := \{G \mid \mathrm{bed}_{\mathcal{G}}(G) \leq k\}$.

Theorem 1

For every minor-closed $\mathcal{G}$, there is a computable function that maps $\mathrm{obs}(\mathcal{G})$ to $\mathrm{obs}(\mathcal{G}^{(k)})$.

Our first result

$\mathcal{G}^{(k)} := \{G \mid \mathrm{bed}_{\mathcal{G}}(G) \leq k\}$.

Theorem 1

For every minor-closed $\mathcal{G}$, there is a computable function that maps $\mathrm{obs}(\mathcal{G})$ to $\mathrm{obs}(\mathcal{G}^{(k)})$.

Proof Idea:

Our first result

$\mathcal{G}^{(k)} := \{G \mid \mathrm{bed}_{\mathcal{G}}(G) \leq k\}.$

Theorem 1

For every minor-closed $\mathcal{G}$, there is a computable function that maps $\mathrm{obs}(\mathcal{G})$ to $\mathrm{obs}(\mathcal{G}^{(k)})$.

Proof Idea:

Biconnected closure of $\mathcal{G}$, $\mathrm{bcl}(\mathcal{G}) := \{G \mid \mathcal{B}(G) \subseteq \mathcal{G}\}$.

Our first result

$\mathcal{G}^{(k)} := \{G \mid \mathrm{bed}_{\mathcal{G}}(G) \leq k\}$.

Theorem 1

For every minor-closed $\mathcal{G}$, there is a computable function that maps $\mathrm{obs}(\mathcal{G})$ to $\mathrm{obs}(\mathcal{G}^{(k)})$.

Proof Idea:

Biconnected closure of $\mathcal{G}$, $\mathrm{bcl}(\mathcal{G}) := \{G \mid \mathcal{B}(G) \subseteq \mathcal{G}\}$.

We prove the following intermediate result:

Lemma

There is an algorithm that, given $\mathrm{obs}(\mathcal{G})$, outputs $\mathrm{obs}(\mathrm{bcl}(\mathcal{G}))$.

Our first result

$\mathcal{G}^{(k)} := \{G \mid \mathrm{bed}_{\mathcal{G}}(G) \leq k\}$.

Theorem 1

For every minor-closed $\mathcal{G}$, there is a computable function that maps $\mathrm{obs}(\mathcal{G})$ to $\mathrm{obs}(\mathcal{G}^{(k)})$.

Proof Idea:

Biconnected closure of $\mathcal{G}$, $\mathrm{bcl}(\mathcal{G}) := \{G \mid \mathcal{B}(G) \subseteq \mathcal{G}\}$.

We prove the following intermediate result:

Lemma

There is an algorithm that, given $\mathrm{obs}(\mathcal{G})$, outputs $\mathrm{obs}(\mathrm{bcl}(\mathcal{G}))$.

Lemma $+$ algorithm of Adler et al. $\rightarrow$ **Theorem 1**

Block elimination distance to edgeless graphs

"Biconnected analogue" of treedepth.

Block elimination distance to edgeless graphs

"Biconnected analogue" of treedepth.

Definition

$$\mathrm{btd}(G) = \begin{cases} \max\{\mathrm{btd}(H) \mid H \in \mathcal{B}(G)\} & \text{if } |\mathcal{B}(G)| > 1 \text{ and } E(G) \neq \emptyset, \\ 1 + \min_{v \in V(G)}\{\mathrm{btd}(G \setminus v)\} & \text{if } G \text{ is biconnected and } E(G) \neq \emptyset, \\ 1 & G \text{ is edgeless.} \end{cases}$$

$\mathrm{btd}(G) = \mathrm{bed}_{\mathcal{E}}(G)$, where $\mathcal{E}$ is the class of edgeless graphs.

Block elimination distance to edgeless graphs

"Biconnected analogue" of treedepth.

Definition

$$\mathrm{btd}(G) = \begin{cases} \max\{\mathrm{btd}(H) \mid H \in \mathcal{B}(G)\} & \text{if } |\mathcal{B}(G)| > 1 \text{ and } E(G) \neq \emptyset, \\ 1 + \min_{v \in V(G)}\{\mathrm{btd}(G \setminus v)\} & \text{if } G \text{ is biconnected and } E(G) \neq \emptyset, \\ 1 & G \text{ is edgeless.} \end{cases}$$

$$\mathrm{btd}(G) = \mathrm{bed}_{\mathcal{E}}(G), \text{ where } \mathcal{E} \text{ is the class of edgeless graphs.}$$

$$\mathrm{td}(P_n) = \Theta(\log n)$$

Block elimination distance to edgeless graphs

"Biconnected analogue" of treedepth.

Definition

$$\mathrm{btd}(G) = \begin{cases} \max\{\mathrm{btd}(H) \mid H \in \mathcal{B}(G)\} & \text{if } |\mathcal{B}(G)| > 1 \text{ and } E(G) \neq \emptyset, \\ 1 + \min_{v \in V(G)}\{\mathrm{btd}(G \setminus v)\} & \text{if } G \text{ is biconnected and } E(G) \neq \emptyset, \\ 1 & G \text{ is edgeless.} \end{cases}$$

$$\mathrm{btd}(G) = \mathrm{bed}_{\mathcal{E}}(G), \text{ where } \mathcal{E} \text{ is the class of edgeless graphs.}$$

$$\mathrm{td}(P_n) = \Theta(\log n)$$

Block elimination distance to edgeless graphs

"Biconnected analogue" of treedepth.

Definition

$$\mathrm{btd}(G) = \begin{cases} \max\{\mathrm{btd}(H) \mid H \in \mathcal{B}(G)\} & \text{if } |\mathcal{B}(G)| > 1 \text{ and } E(G) \neq \emptyset, \\ 1 + \min_{v \in V(G)}\{\mathrm{btd}(G \setminus v)\} & \text{if } G \text{ is biconnected and } E(G) \neq \emptyset, \\ 1 & G \text{ is edgeless.} \end{cases}$$

$$\mathrm{btd}(G) = \mathrm{bed}_{\mathcal{E}}(G), \text{ where } \mathcal{E} \text{ is the class of edgeless graphs.}$$

$$\mathrm{td}(P_n) = \Theta(\log n)$$
$$\mathrm{btd}(P_n) = 2$$

Block elimination distance to edgeless graphs

"Biconnected analogue" of treedepth.

Definition

$$\mathrm{btd}(G) = \begin{cases} \max\{\mathrm{btd}(H) \mid H \in \mathcal{B}(G)\} & \text{if } |\mathcal{B}(G)| > 1 \text{ and } E(G) \neq \emptyset, \\ 1 + \min_{v \in V(G)}\{\mathrm{btd}(G \setminus v)\} & \text{if } G \text{ is biconnected and } E(G) \neq \emptyset, \\ 1 & G \text{ is edgeless.} \end{cases}$$

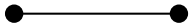
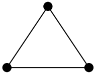
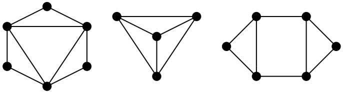
$$\mathrm{btd}(G) = \mathrm{bed}_{\mathcal{E}}(G), \text{ where } \mathcal{E} \text{ is the class of edgeless graphs.}$$

$$\mathrm{td}(P_n) = \Theta(\log n)$$
$$\mathrm{btd}(P_n) = 2$$

Observation: $\{G \mid \mathrm{btd}(G) \leq 2\} = \mathcal{E}^{(2)} = \{\text{forests}\}$.

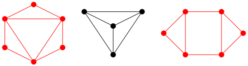
Obstruction sets for bounded btd

We study the set obs$(\mathcal{E}^{(k)})$ for every $k \in \mathbb{N}$.

	obs$(\mathcal{E}^{(k)})$
k $= 1$	
k $= 2$	
k $= 3$	

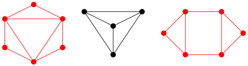
Our second result

We completely identify the outerplanar graphs in obs($\mathcal{E}^{(k)}$).



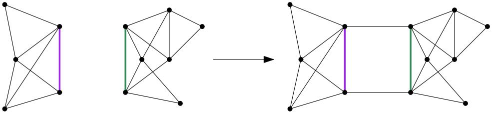
Our second result

We completely identify the outerplanar graphs in obs($\mathcal{E}^{(k)}$).



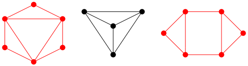
We define the following operations:

- Parallel join:



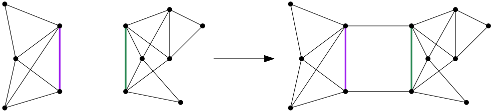
Our second result

We completely identify the outerplanar graphs in obs($\mathcal{E}^{(k)}$).

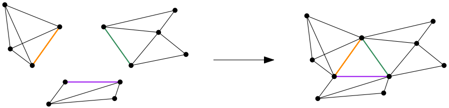


We define the following operations:

- Parallel join:

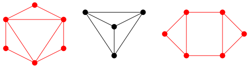


- Triangle gluing:



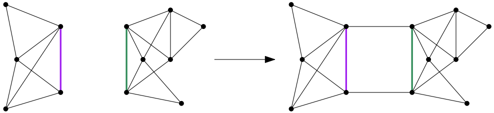
Our second result

We completely identify the outerplanar graphs in obs($\mathcal{E}^{(k)}$).

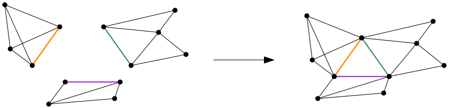


We define the following operations:

- Parallel join:



- Triangle gluing:



Theorem 2

Every outerplanar graph in obs($\mathcal{E}^{(k)}$) can be obtained by parallel join or triangle gluing of outerplanar graphs in obs($\mathcal{E}^{(k-1)}$).

Conclusion

Open Problems:

- Explicit function f for computing $\text{obs}(\mathcal{Z}_k(\mathcal{G})$ and $\text{obs}(\mathcal{G}^{(k)})$?
- Optimize the parametric dependence (on k) of the algorithms.

Thank You!