Exploration of graphs with excluded minors

Júlia Baligács, Yann Disser, Irene Heinrich, Pascal Schweitzer

TU Darmstadt, Germany

ESA 2023



The online graph exploration problem

Example: visit all cities in an unknown area

The online graph exploration problem

Example: visit all cities in an unknown area

starting position:



roads:





The online graph exploration problem

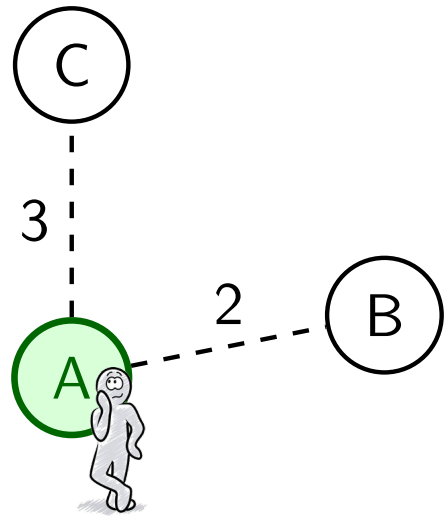
Example: visit all cities in an unknown area

starting position:



roads:





The online graph exploration problem

Example: visit all cities in an unknown area

current position:

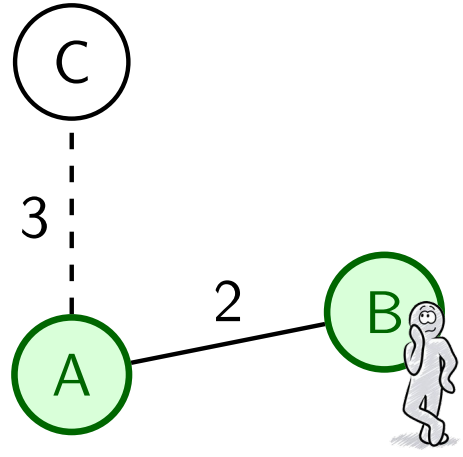
Birkendorf

roads:

Avenfeld 2km

Cederberg 4km

Drachenmoor 5km



The online graph exploration problem

Example: visit all cities in an unknown area

current position:

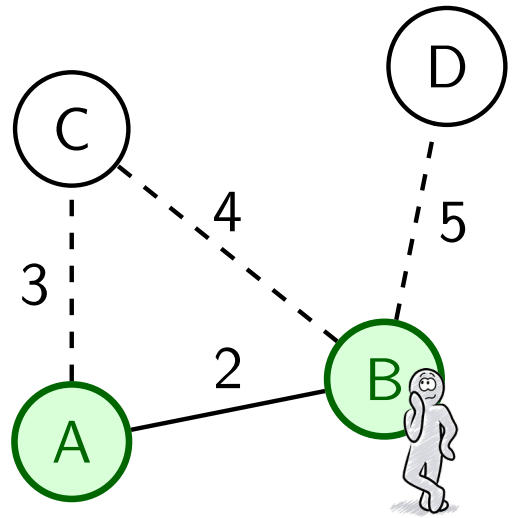
Birkendorf

roads:

Avenfeld 2km

Cederberg 4km

Drachenmoor 5km



The online graph exploration problem

Example: visit all cities in an unknown area

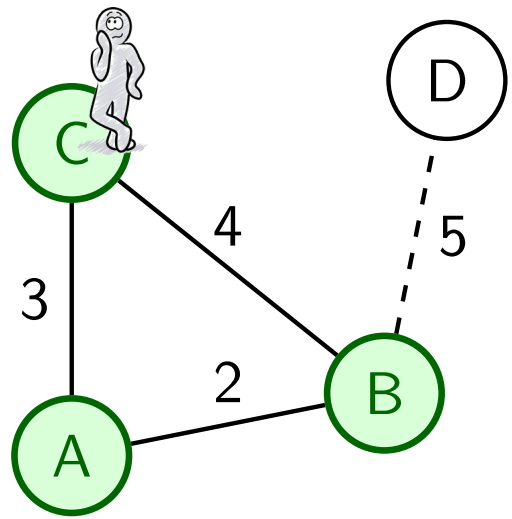
current position:

Cederberg

roads:

Avenfeld 3km

Birkendorf 4km



The online graph exploration problem

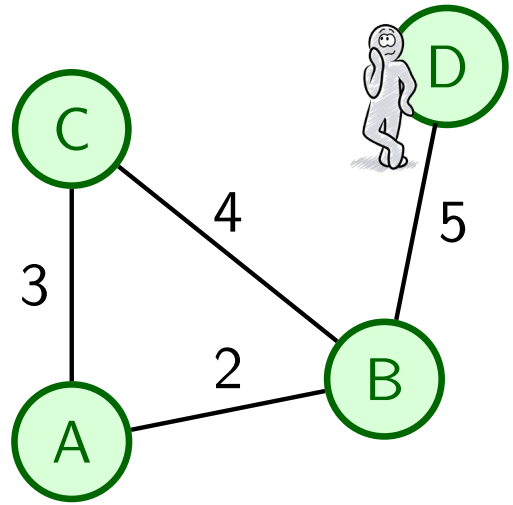
Example: visit all cities in an unknown area

current position:

Drachenmoor

roads:

Birkendorf 5km



The online graph exploration problem

Example: visit all cities in an unknown area

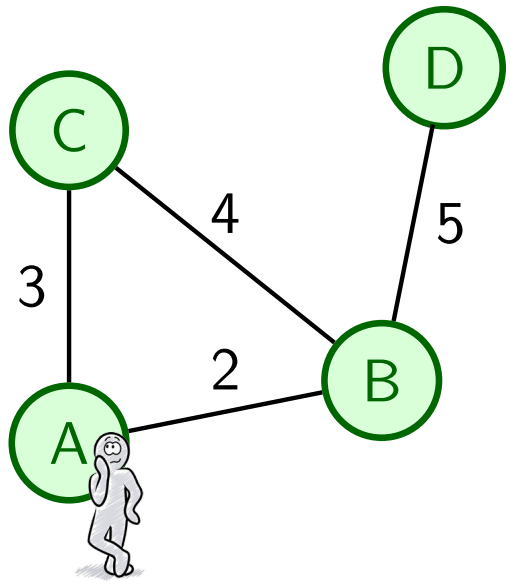
current position:

Avenfeld

roads:

Birkendorf 2km

Cederberg 3km



The online graph exploration problem

Example: visit all cities in an unknown area

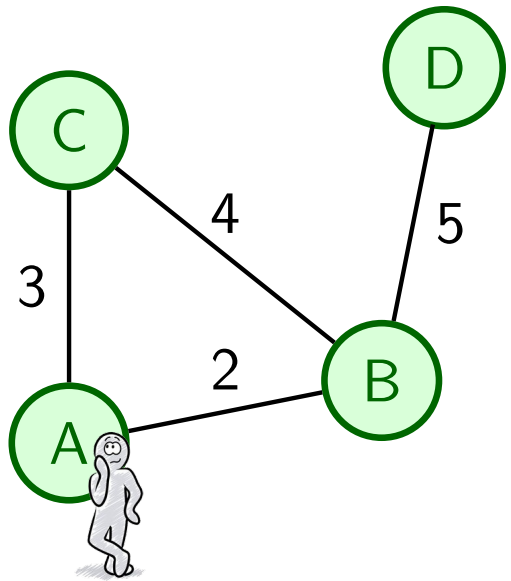
current position:

Avenfeld

roads:

Birkendorf 2km

Cederberg 3km



- tour: $A \xrightarrow{2} B \xrightarrow{4} C \xrightarrow{9} D \xrightarrow{7} A$, total cost: 22

The online graph exploration problem

Example: visit all cities in an unknown area

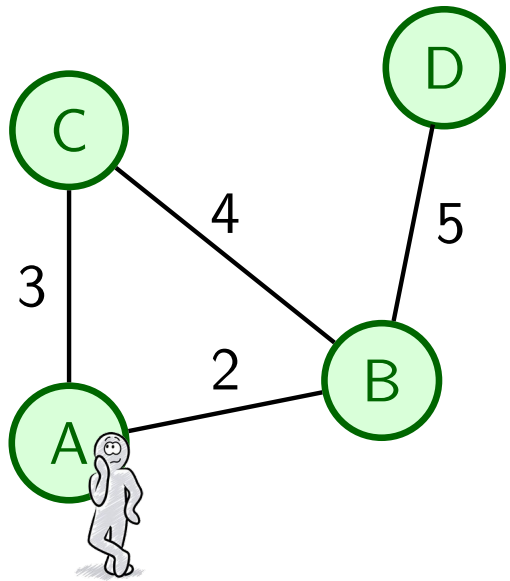
current position:

Avenfeld

roads:

Birkendorf 2km

Cederberg 3km



- tour: $A \xrightarrow{2} B \xrightarrow{4} C \xrightarrow{9} D \xrightarrow{7} A$, total cost: 22
- optimal tour: $A \xrightarrow{2} B \xrightarrow{5} D \xrightarrow{9} C \xrightarrow{3} A$, optimal cost: 19

The online graph exploration problem

Example: visit all cities in an unknown area

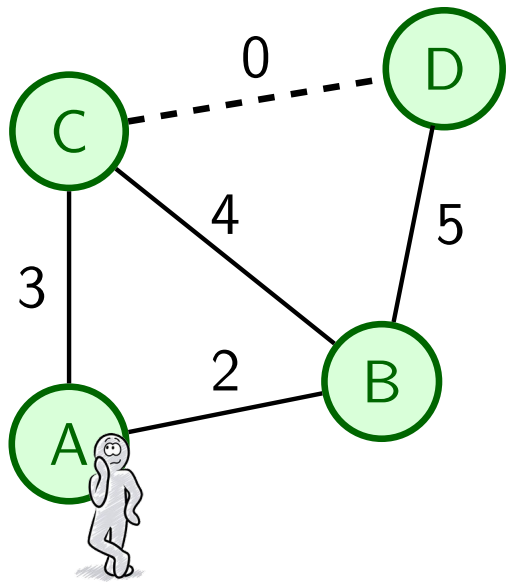
current position:

Avenfeld

roads:

Birkendorf 2km

Cederberg 3km



- tour: $A \xrightarrow{2} B \xrightarrow{4} C \xrightarrow{9} D \xrightarrow{7} A$, total cost: 22
- optimal tour: $A \xrightarrow{2} B \xrightarrow{5} D \xrightarrow{9} C \xrightarrow{3} A$, optimal cost: 19

Problem formalization

The online graph exploration problem

setting:
- single agent has to
  - visit every vertex of a graph
    (weighted, undirected, connected)
  - return to start

objective:

Problem formalization

The online graph exploration problem

setting:

- single agent has to
  - visit every vertex of a graph
    (weighted, undirected, connected)
  - return to start
- upon visiting new vertex, agent learns:
  - adjacent vertices and weights of corresponding edges

objective:

Problem formalization

The online graph exploration problem

setting:

- single agent has to
  - visit every vertex of a graph
    (weighted, undirected, connected)
  - return to start
- upon visiting new vertex, agent learns:
  - adjacent vertices and weights of corresponding edges

objective:

- minimize total walked distance

Competitive analysis

Definition: given exploration algorithm $\textsc{Alg}$,

a) $\textsc{Alg}(G, v)$: total walked distance of $\textsc{Alg}$ on graph G starting in v

Competitive analysis

Definition: given exploration algorithm $\textsc{Alg}$,

a) $\textsc{Alg}(G, v)$: total walked distance of $\textsc{Alg}$ on graph G starting in v

b) $\textsc{Opt}(G)$: length of a shortest TSP tour on G

Competitive analysis

Definition: given exploration algorithm $\textsc{Alg}$,

a) $\textsc{Alg}(G, v)$: total walked distance of $\textsc{Alg}$ on graph G starting in v

b) $\textsc{Opt}(G)$: length of a shortest TSP tour on G

c) $\textsc{Alg}$ is ρ-**competitive** if $\textsc{Alg}(G, v) \leq \rho \cdot \textsc{Opt}(G)$ for all G, v.

Competitive analysis

Definition: given exploration algorithm $\textsc{Alg}$,

a) $\textsc{Alg}(G, v)$: total walked distance of $\textsc{Alg}$ on graph G starting in v

b) $\textsc{Opt}(G)$: length of a shortest TSP tour on G

c) $\textsc{Alg}$ is ρ-**competitive** if $\textsc{Alg}(G, v) \leq \rho \cdot \textsc{Opt}(G)$ for all G, v.

d) The **competitive ratio** of $\textsc{Alg}$ is $\inf\{\rho : \textsc{Alg}$ is ρ-competitive$\}$.

Competitive analysis

Definition: given exploration algorithm $\textsc{Alg}$,

a) $\textsc{Alg}(G, v)$: total walked distance of $\textsc{Alg}$ on graph G starting in v

b) $\textsc{Opt}(G)$: length of a shortest TSP tour on G

c) $\textsc{Alg}$ is ρ-**competitive** if $\textsc{Alg}(G, v) \leq \rho \cdot \textsc{Opt}(G)$ for all G, v.

d) The **competitive ratio** of $\textsc{Alg}$ is $\inf\{\rho : \textsc{Alg}$ is ρ-competitive$\}$.

Open question: (Kalyanasundaram and Pruhs, 1994)
Is there a constant-competitive algorithm for online graph exploration?

A simple algorithm

- **Nearest Neighbour (NN)**: go to the closest unexplored vertex

A simple algorithm

- **Nearest Neighbour (NN)**: go to the closest unexplored vertex
- Theorem. The competitive ratio of NN is $\Theta(\log n)$.

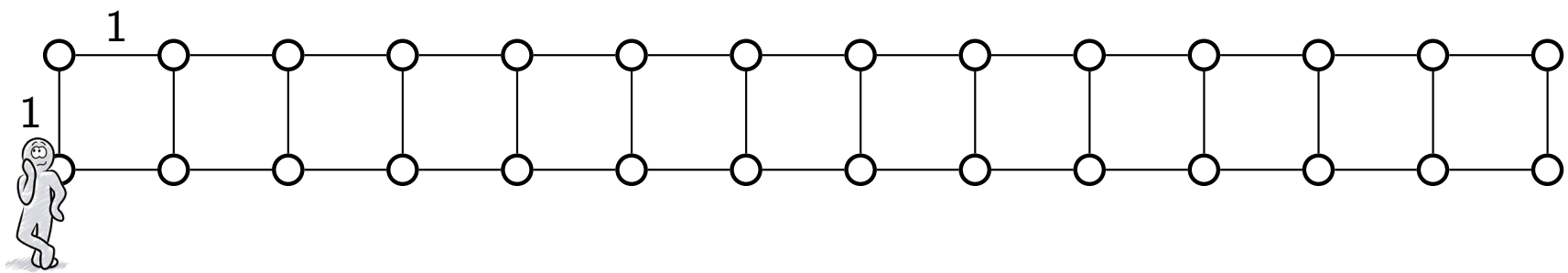
(Rosenkrantz, Stearns, Lewis, 1977)

A simple algorithm

- **Nearest Neighbour (NN)**: go to the closest unexplored vertex
- Theorem. The competitive ratio of NN is $\Theta(\log n)$.

(Rosenkrantz, Stearns, Lewis, 1977)

proof idea for lower bound:

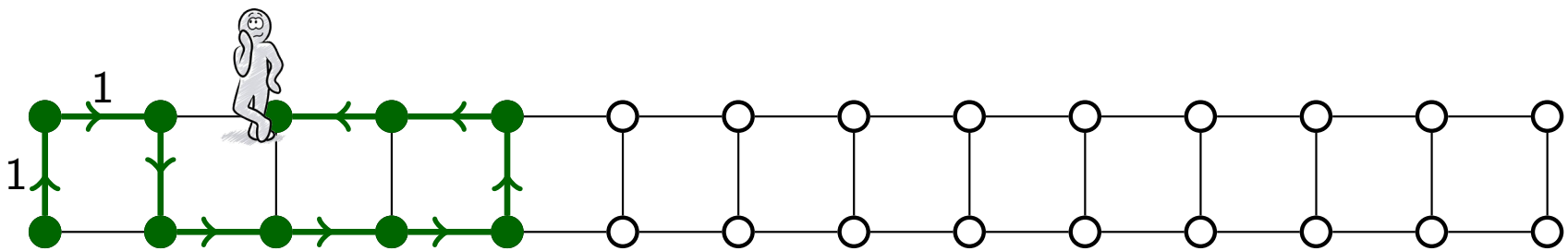


A simple algorithm

- **Nearest Neighbour (NN)**: go to the closest unexplored vertex
- Theorem. The competitive ratio of NN is $\Theta(\log n)$.

(Rosenkrantz, Stearns, Lewis, 1977)

proof idea for lower bound:

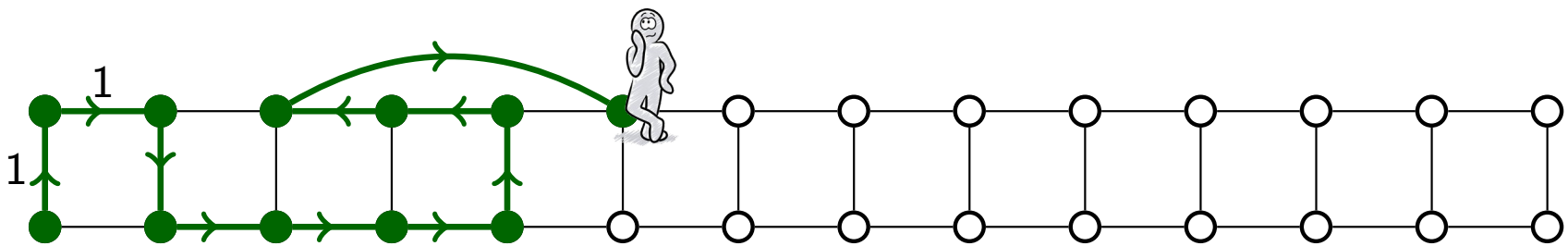


A simple algorithm

- **Nearest Neighbour (NN)**: go to the closest unexplored vertex
- Theorem. The competitive ratio of NN is $\Theta(\log n)$.

(Rosenkrantz, Stearns, Lewis, 1977)

proof idea for lower bound:

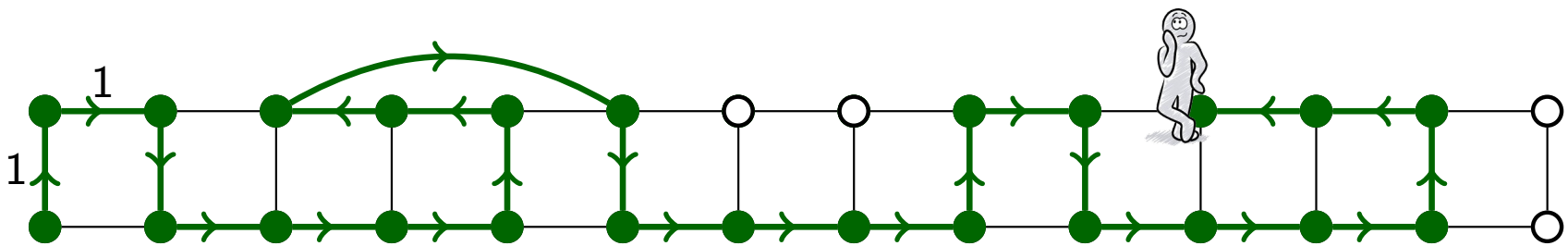


A simple algorithm

- **Nearest Neighbour (NN)**: go to the closest unexplored vertex
- Theorem. The competitive ratio of NN is $\Theta(\log n)$.

(Rosenkrantz, Stearns, Lewis, 1977)

proof idea for lower bound:

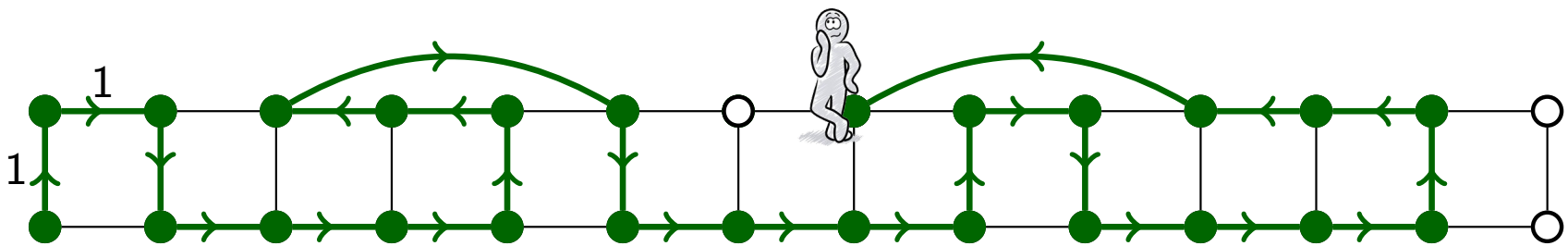


A simple algorithm

- **Nearest Neighbour (NN)**: go to the closest unexplored vertex
- Theorem. The competitive ratio of NN is $\Theta(\log n)$.

(Rosenkrantz, Stearns, Lewis, 1977)

proof idea for lower bound:

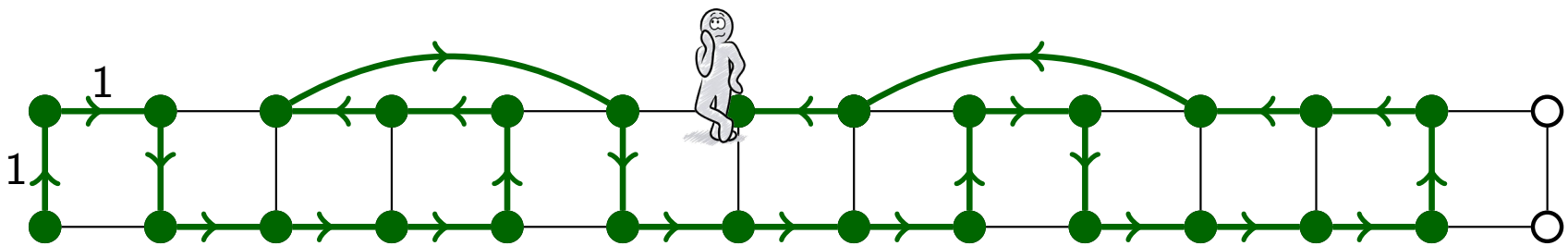


A simple algorithm

- **Nearest Neighbour (NN)**: go to the closest unexplored vertex
- Theorem. The competitive ratio of NN is $\Theta(\log n)$.

(Rosenkrantz, Stearns, Lewis, 1977)

proof idea for lower bound:

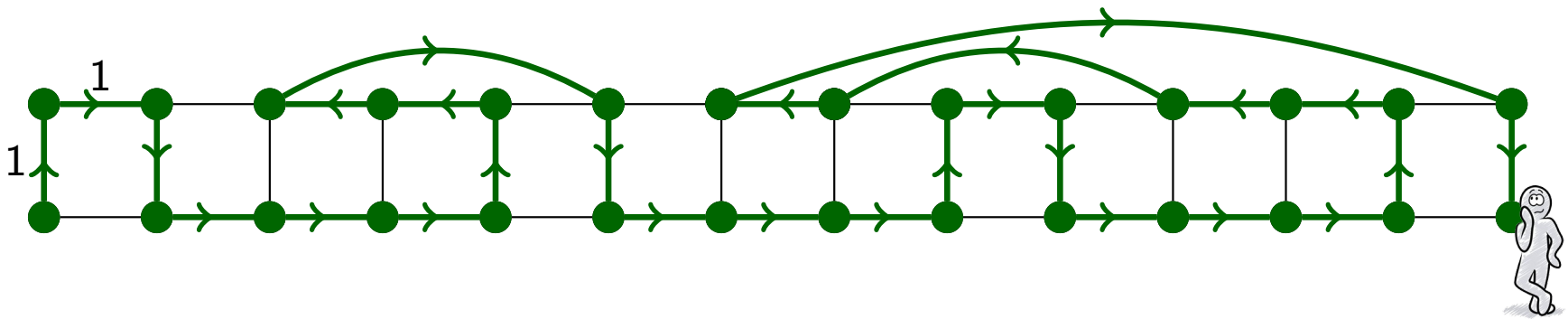


A simple algorithm

- **Nearest Neighbour (NN)**: go to the closest unexplored vertex
- Theorem. The competitive ratio of NN is $\Theta(\log n)$.

(Rosenkrantz, Stearns, Lewis, 1977)

proof idea for lower bound:



State of the art

General graphs

- No algorithm with competitive ratio $o(\log n)$ is known yet.

State of the art

General graphs

- No algorithm with competitive ratio $o(\log n)$ is known yet.
- Every exploration algorithm has competitive ratio $\geq 10/3$.

(Birx, Disser, Hopp, Karousatou, 2021)

State of the art

General graphs

- No algorithm with competitive ratio $o(\log n)$ is known yet.
- Every exploration algorithm has competitive ratio $\geq 10/3$.

(Birx, Disser, Hopp, Karousatou, 2021)

Restricted graph classes

- competitive ratio of 2 on unweighted graphs and 1.366 on cycles

(Miyazaki, Morimoto, Okabe, 2009)

State of the art

General graphs

- No algorithm with competitive ratio $o(\log n)$ is known yet.
- Every exploration algorithm has competitive ratio $\geq 10/3$.

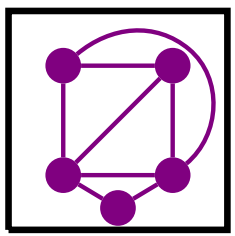
(Birx, Disser, Hopp, Karousatou, 2021)

Restricted graph classes

- competitive ratio of 2 on unweighted graphs and 1.366 on cycles

(Miyazaki, Morimoto, Okabe, 2009)

- There is a constant-competitive algorithm on
  - planar graphs, (Kalyanasundaram, Pruhs, 1994)



State of the art

General graphs

- No algorithm with competitive ratio $o(\log n)$ is known yet.
- Every exploration algorithm has competitive ratio $\geq 10/3$.

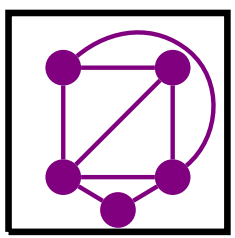
(Birx, Disser, Hopp, Karousatou, 2021)

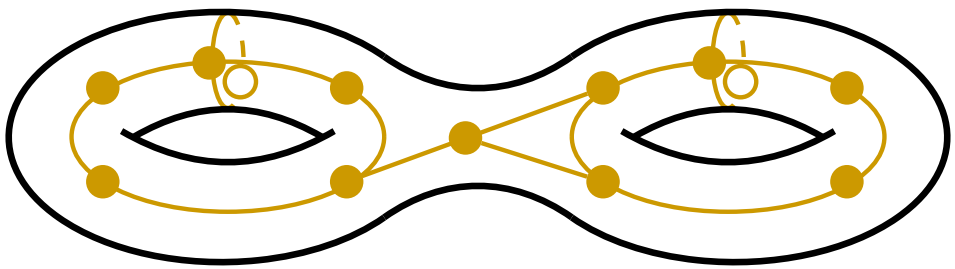
Restricted graph classes

- competitive ratio of 2 on unweighted graphs and 1.366 on cycles

(Miyazaki, Morimoto, Okabe, 2009)

- There is a constant-competitive algorithm on
  - planar graphs, (Kalyanasundaram, Pruhs, 1994)
  - bounded genus graphs. (Megow, Mehlhorn, Schweitzer, 2012)





State of the art

General graphs

- No algorithm with competitive ratio $o(\log n)$ is known yet.
- Every exploration algorithm has competitive ratio $\geq 10/3$.

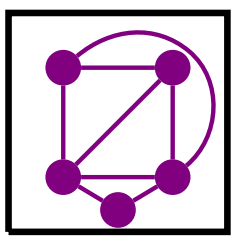
(Birx, Disser, Hopp, Karousatou, 2021)

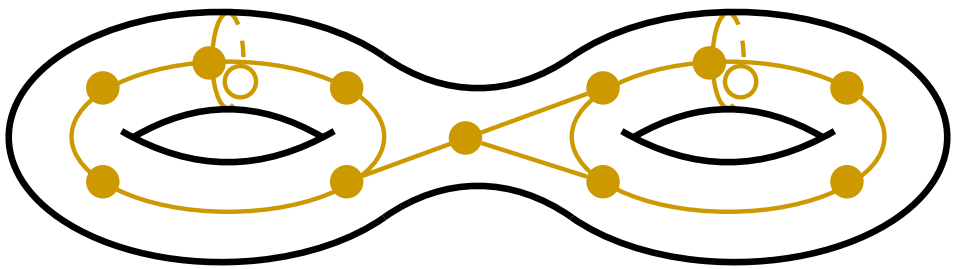
Restricted graph classes

- competitive ratio of 2 on unweighted graphs and 1.366 on cycles

(Miyazaki, Morimoto, Okabe, 2009)

- There is a constant-competitive algorithm on
  - planar graphs, (Kalyanasundaram, Pruhs, 1994)
  - bounded genus graphs. (Megow, Mehlhorn, Schweitzer, 2012)

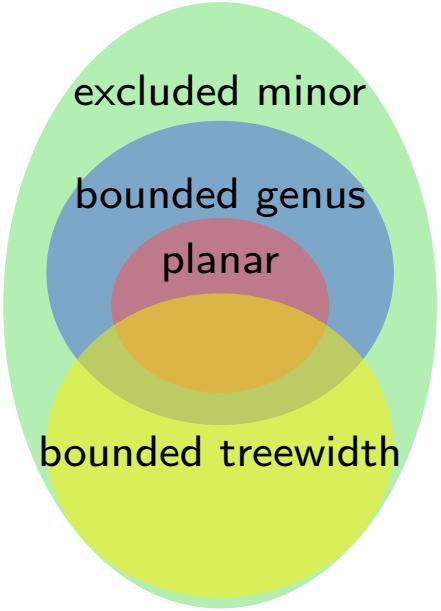




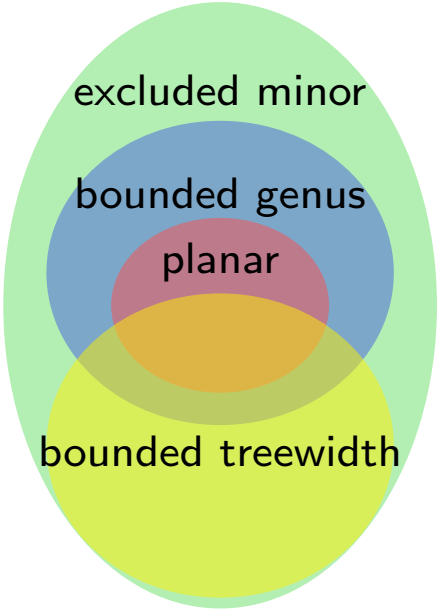
- in this talk: generalization to minor-free graphs

Generalization to minor-free graphs

Theorem. For every graph H, there is a constant-competitive exploration algorithm on H-minor-free graphs.



Generalization to minor-free graphs

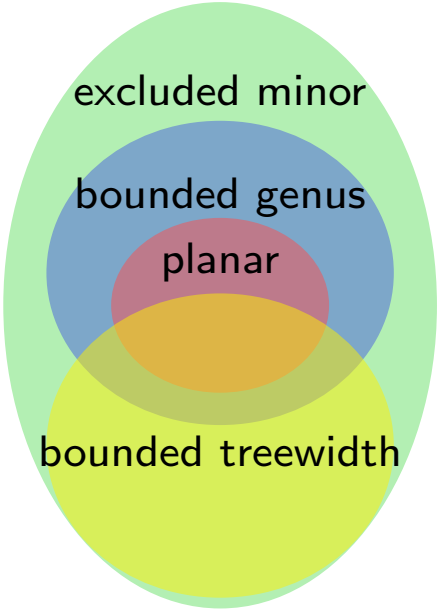


Theorem. For every graph H, there is a constant-competitive exploration algorithm on H-minor-free graphs.

Blocking-algorithm (Megow, Mehlhorn, Schweitzer, 2012):

- perform DFS-search ignoring "blocked" edges

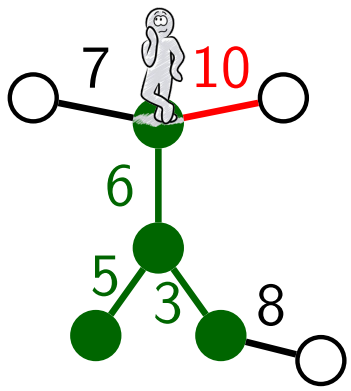
Generalization to minor-free graphs



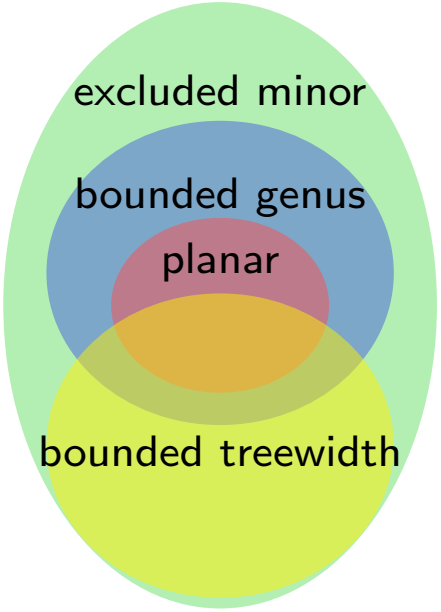
Theorem. For every graph H, there is a constant-competitive exploration algorithm on H-minor-free graphs.

Blocking-algorithm (Megow, Mehlhorn, Schweitzer, 2012):

- perform DFS-search ignoring "blocked" edges
- an edge is "blocked" if
  there is a lighter unexplored edge close-by



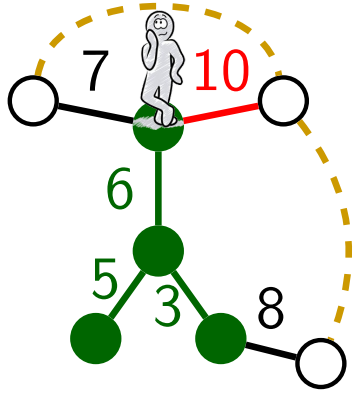
Generalization to minor-free graphs



Theorem. For every graph H, there is a constant-competitive exploration algorithm on H-minor-free graphs.

Blocking-algorithm (Megow, Mehlhorn, Schweitzer, 2012):

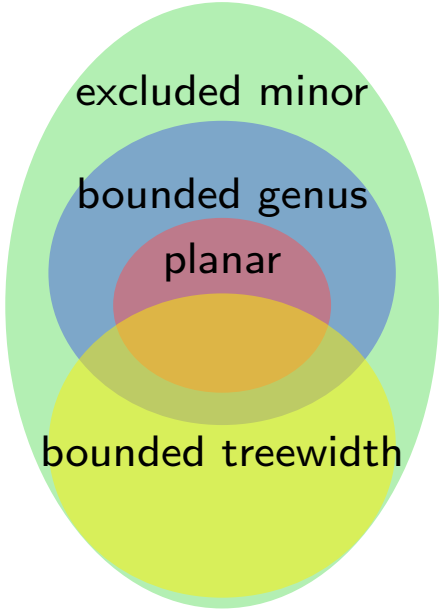
- perform DFS-search ignoring "blocked" edges
- an edge is "blocked" if
  there is a lighter unexplored edge close-by



Keys to analysis:

- cycles in (DFS-tree of Blocking $\cup$ MST) are long
  compared to their edge weights

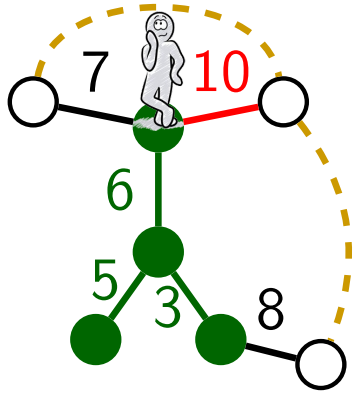
Generalization to minor-free graphs



Theorem. For every graph H, there is a constant-competitive exploration algorithm on H-minor-free graphs.

Blocking-algorithm (Megow, Mehlhorn, Schweitzer, 2012):

- perform DFS-search ignoring "blocked" edges
- an edge is "blocked" if
  there is a lighter unexplored edge close-by



Keys to analysis:

- cycles in (DFS-tree of Blocking $\cup$ MST) are long
  compared to their edge weights

$\rightarrow$ connection to spanners

Blocking and light spanners

Definition. A subgraph $H \subseteq G$ is a $(1+\varepsilon)$-**spanner** of G if

$$d_H(u,v) \leq (1+\varepsilon) d_G(u,v) \text{ for all } u, v \in G.$$

Blocking and light spanners

Definition. A subgraph $H \subseteq G$ is a $(1+\varepsilon)$-**spanner** of G if

$$d_H(u,v) \leq (1+\varepsilon) d_G(u,v) \text{ for all } u, v \in G.$$

- **lightness** of H: $w(H)/w(\mathrm{MST}_G)$

Blocking and light spanners

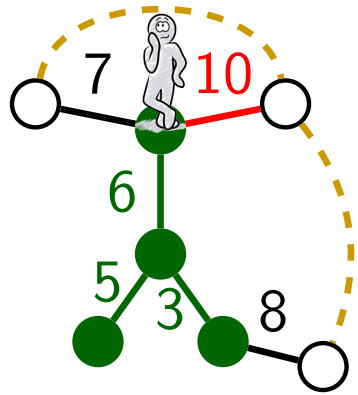
Definition. A subgraph $H \subseteq G$ is a $(1+\varepsilon)$-**spanner** of G if

$$d_H(u,v) \leq (1+\varepsilon) d_G(u,v) \text{ for all } u, v \in G.$$

- **lightness** of H: $w(H)/w(\mathrm{MST}_G)$

Recall:

- cycles in (DFS-tree of Blocking $\cup$ MST) are long compared to their edge weights



Blocking and light spanners

Definition. A subgraph $H \subseteq G$ is a $(1+\varepsilon)$-**spanner** of G if

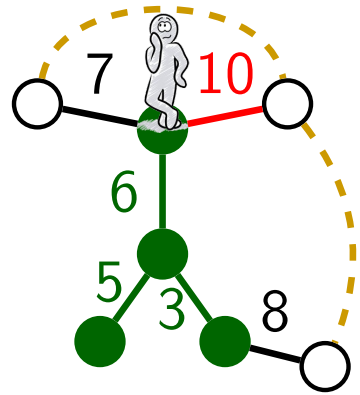
$$d_H(u,v) \leq (1+\varepsilon) d_G(u,v) \text{ for all } u, v \in G.$$

- **lightness** of H: $w(H)/w(\mathrm{MST}_G)$

Recall:

- cycles in (DFS-tree of Blocking $\cup$ MST) are long compared to their edge weights

$\Rightarrow$ no proper subgraph is a $(1+\varepsilon)$-spanner



Blocking and light spanners

Definition. A subgraph $H \subseteq G$ is a $(1+\varepsilon)$-**spanner** of G if

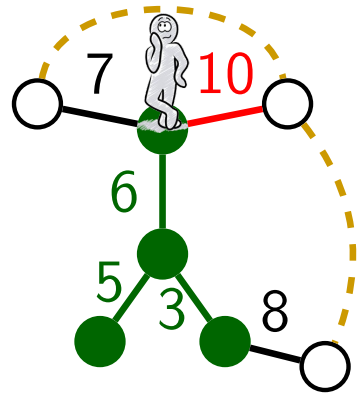
$$d_H(u,v) \leq (1+\varepsilon) d_G(u,v) \text{ for all } u, v \in G.$$

- **lightness** of H: $w(H)/w(\mathrm{MST}_G)$

Recall:

- cycles in (DFS-tree of Blocking $\cup$ MST) are long compared to their edge weights

$\Rightarrow$ no proper subgraph is a $(1+\varepsilon)$-spanner



Theorem. BLOCKING is constant-competitive on monotone graph classes "admitting light spanners".

Blocking and light spanners

Definition. A subgraph $H \subseteq G$ is a $(1+\varepsilon)$-**spanner** of G if

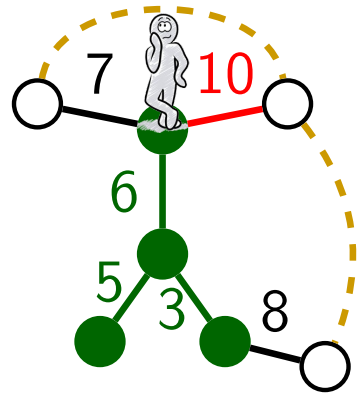
$$d_H(u,v) \leq (1+\varepsilon) d_G(u,v) \text{ for all } u, v \in G.$$

- **lightness** of H: $w(H)/w(\mathrm{MST}_G)$

Recall:

- cycles in (DFS-tree of Blocking $\cup$ MST) are long compared to their edge weights

$\Rightarrow$ no proper subgraph is a $(1+\varepsilon)$-spanner



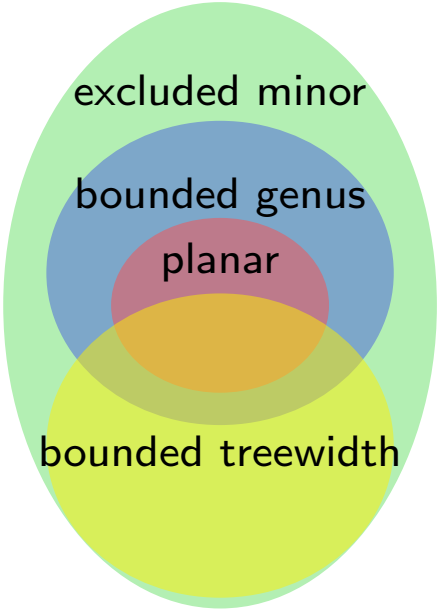
Theorem. BLOCKING is constant-competitive on monotone graph classes "admitting light spanners".

Minor-free graphs admit light spanners! (Borradaile, Le, Wulff-Nilsen, 2017)

Our results

connection between BLOCKING and spanners

Our results

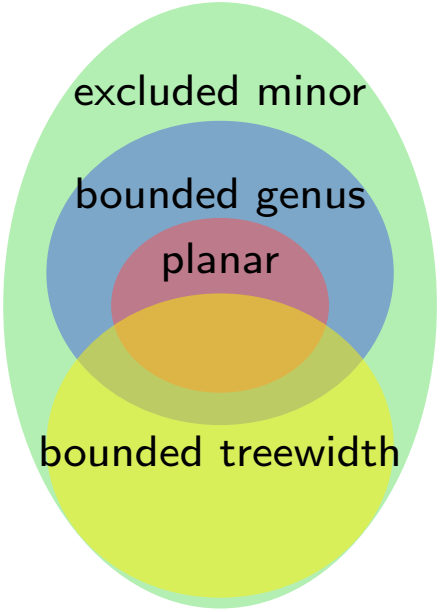


Theorem. There is a constant-competitive exploration algorithm on minor-free graphs.

minor-free graphs have light spanners
(Borradaile, Le, Wulff-Nilsen, 2017)

connection between BLOCKING and spanners

Our results



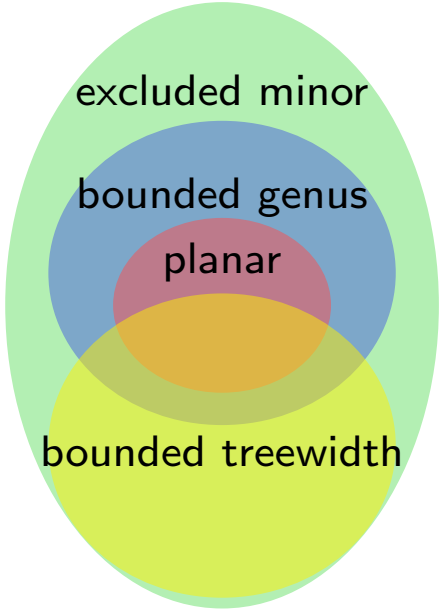
Theorem. There is a constant-competitive exploration algorithm on minor-free graphs.

minor-free graphs have light spanners
(Borradaile, Le, Wulff-Nilsen, 2017)

connection between BLOCKING and spanners

upper bound on the lightness of spanners in bounded-genus graphs
(improvement on [Grigni, 2000])

Our results



Theorem. There is a constant-competitive exploration algorithm on minor-free graphs.

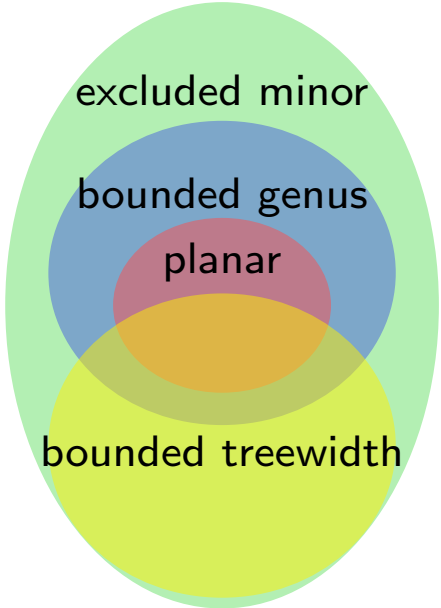
minor-free graphs have light spanners
(Borradaile, Le, Wulff-Nilsen, 2017)

connection between BLOCKING and spanners

$\omega(1)$ lower bound on BLOCKING

upper bound on the lightness of spanners in bounded-genus graphs
(improvement on [Grigni, 2000])

Our results



Theorem. There is a constant-competitive exploration algorithm on minor-free graphs.

minor-free graphs have light spanners
(Borradaile, Le, Wulff-Nilsen, 2017)

connection between BLOCKING and spanners

$\omega(1)$ lower bound on BLOCKING

upper bound on the lightness of spanners in bounded-genus graphs
(improvement on [Grigni, 2000])

Open question: (Kalyanasundaram and Pruhs, 1994)
Is there a $O(1)$-competitive algorithm for online graph exploration?