Generalizations of binary search to graphs

Dariusz Dereniowski

Gdańsk University of Technology

Thanks to Przemysław Uznański for some slides
Many thanks to Adrian Kosowski

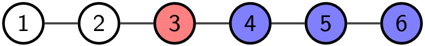
GRASTA 2023, October 22-27, Bertinoro

Model

- Input graph G
- adversary holds a hidden target vertex t
- query: e.g. vertex v for a direction $v \to t$
- Goal: find t minimizing the number of queries.

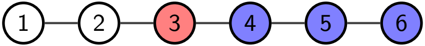
Vertex queries

binary search $\rightarrow$

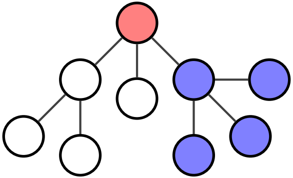


Vertex queries

binary search →



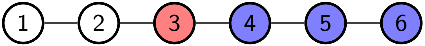
tree search →



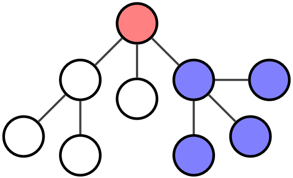
Onak and Parys [FOCS 2006]:

Vertex queries

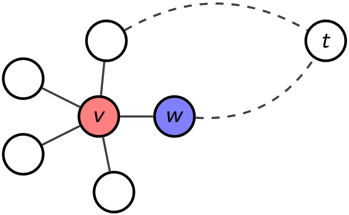
binary search →



tree search →
Onak and Parys [FOCS 2006]:



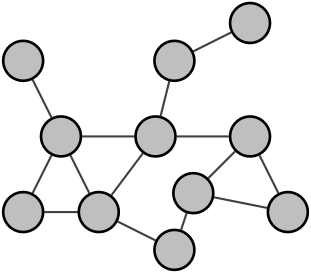
graph queries →
Emamjomeh-Zadeh, Kempe and Singhal [STOC 2016]:



For all of above models: query complexity $= \log_2 n$.

Graph queries: example

(For this graph, 3 queries are enough).



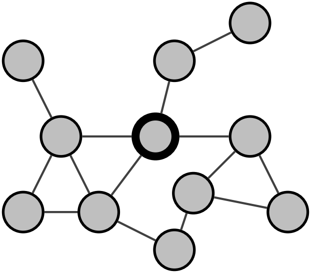
 - eliminated vertex

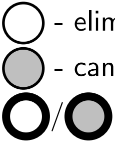
 - candidate vertex

 - query vertex

Graph queries: example

(For this graph, 3 queries are enough).

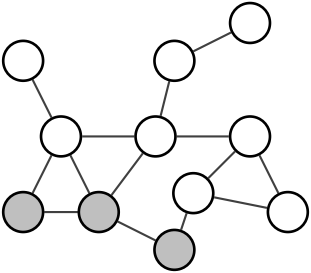




- eliminated vertex
- candidate vertex
- query vertex

Graph queries: example

(For this graph, 3 queries are enough).



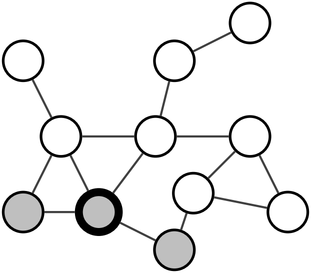
- eliminated vertex

- candidate vertex

/ - query vertex

Graph queries: example

(For this graph, 3 queries are enough).



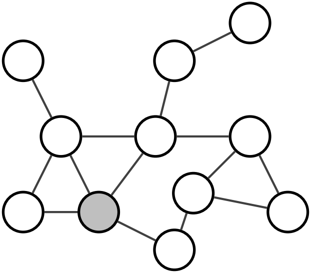
 - eliminated vertex

 - candidate vertex

 - query vertex

Graph queries: example

(For this graph, 3 queries are enough).



- eliminated vertex

- candidate vertex

/ - query vertex

Graph queries in general graphs

Theorem (Emamjomeh-Zadeh, Kempe and Singhal [STOC 2016])

There exists a strategy that finds the target vertex in at most $\log_2 n$ queries.

Graph queries in general graphs

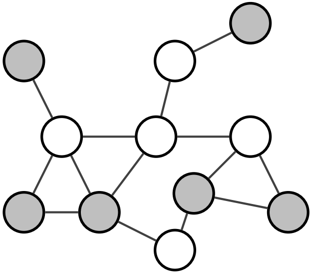
Theorem (Emamjomeh-Zadeh, Kempe and Singhal [STOC 2016])

There exists a strategy that finds the target vertex in at most $\log_2 n$ queries.

Strategy (no errors/lies):

Query 1-median of set of targets.

$\arg\min_{v \in V} \sum_{x \in X} d(v, x)$



Graph queries in general graphs

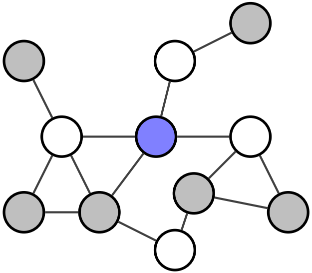
Theorem (Emamjomeh-Zadeh, Kempe and Singhal [STOC 2016])

There exists a strategy that finds the target vertex in at most $\log_2 n$ queries.

Strategy (no errors/lies):

Query 1-median of set of targets.

$\arg\min_{v \in V} \sum_{x \in X} d(v, x)$



Graph queries in general graphs

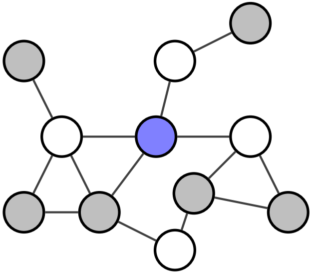
Theorem (Emamjomeh-Zadeh, Kempe and Singhal [STOC 2016])

There exists a strategy that finds the target vertex in at most $\log_2 n$ queries.

Strategy (no errors/lies):

Query 1-median of set of targets.

$$\arg\min_{v \in V} \sum_{x \in X} d(v, x)$$



Graph queries in general graphs

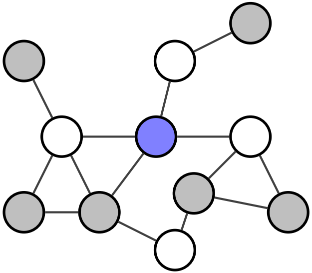
Theorem (Emamjomeh-Zadeh, Kempe and Singhal [STOC 2016])

There exists a strategy that finds the target vertex in at most $\log_2 n$ queries.

Strategy (no errors/lies):

Query 1-median of set of targets.

$\arg\min_{v \in V} \sum_{x \in X} d(v, x)$



Lemma:

- set S of targets
- q is 1-median of S
- any answer $v \in N(q)$

$$|X \cap S| \geq \frac{1}{2}|S|$$

inconsistent with the answer

Graph queries: applications

Graph queries: applications

Searching general graphs captures many robust interactive learning scenarios:

- binary and non-binary classifiers, orderings/rankings of items, clusterings [Emamjomeh-Zadeh, Kempe NIPS 2017],
- hierarchical clustering [Emamjomeh-Zadeh, Kempe SODA 2018].

Graph queries: applications

Searching general graphs captures many robust interactive learning scenarios:

- binary and non-binary classifiers, orderings/rankings of items, clusterings [Emamjomeh-Zadeh, Kempe NIPS 2017],
- hierarchical clustering [Emamjomeh-Zadeh, Kempe SODA 2018].

Tree search allows automated software testing [Y.Ben-Asher, E.Farchi, I.Newman, SICOMP 99]

Vertex rankings

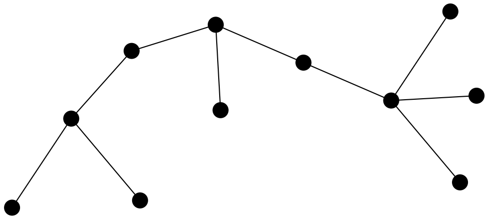
Definition

A vertex ranking is an assignment of positive integers (colors) to the vertices of a graph in such a way that for any two vertices u and v with the same color, each path between u and v has at least one vertex with a higher color. The goal is to minimize the number of colors.

Vertex rankings

Definition

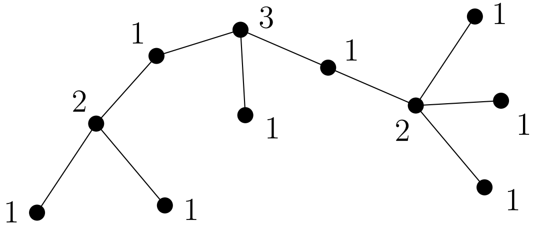
A vertex ranking is an assignment of positive integers (colors) to the vertices of a graph in such a way that for any two vertices u and v with the same color, each path between u and v has at least one vertex with a higher color. The goal is to minimize the number of colors.



Vertex rankings

Definition

A vertex ranking is an assignment of positive integers (colors) to the vertices of a graph in such a way that for any two vertices u and v with the same color, each path between u and v has at least one vertex with a higher color. The goal is to minimize the number of colors.



Other connections

In case of trees, the following problems are equivalent:

- our graph search by asking vertex queries
- vertex ranking (sometimes called ordered coloring)
- treedepth
- minimum height elimination tree computation
- LIFO-search

The complexity

Theorem (Schäffer [IPL 1989], Onak and Parys [FOCS 2006])

There exists a linear-time algorithm for finding an optimal search strategy for any tree.

A generalization: weighted trees

Definition

The search problem remains the same except that:

- each vertex v has a weight $\omega(v)$
- the time (or cost) to perform a query on v is $\omega(v)$
- the goal it to find a strategy that has minimum cost in the worst case

A generalization: weighted trees

Definition

The search problem remains the same except that:

- each vertex v has a weight $\omega(v)$
- the time (or cost) to perform a query on v is $\omega(v)$
- the goal it to find a strategy that has minimum cost in the worst case

Theorem (DD [IPL 2006])

Finding an optimal search strategy for a wieghted tree is strongly NP-hard.

A generalization: weighted trees

Definition

The search problem remains the same except that:

- each vertex v has a weight $\omega(v)$
- the time (or cost) to perform a query on v is $\omega(v)$
- the goal it to find a strategy that has minimum cost in the worst case

Theorem (DD [IPL 2006])

Finding an optimal search strategy for a wieghted tree is strongly NP-hard. (This en route proves that treedepth is hard for chordal graphs.)

A generalization: weighted trees

Definition

The search problem remains the same except that:

- each vertex v has a weight $\omega(v)$
- the time (or cost) to perform a query on v is $\omega(v)$
- the goal it to find a strategy that has minimum cost in the worst case

Theorem (DD [IPL 2006])

Finding an optimal search strategy for a wieghted tree is strongly NP-hard. (This en route proves that treedepth is hard for chordal graphs.)

Theorem (DD, Kosowski, Uznański and Zou [ICALP 2017])

There exists a polynomial-time algorithm with approximation ratio of $\mathcal{O}(\sqrt{\log n})$ *for searching a weighted tree.*

Open problems

- main one: is there a polynomial-time constant factor approximation for weighted trees?

Open problems

- main one: is there a polynomial-time constant factor approximation for weighted trees?
- analyze *edge search* instead?

Open problems

- main one: is there a polynomial-time constant factor approximation for weighted trees?
- analyze *edge search* instead?
- analyze restricted weight functions for trees (we already have solved *down-monotonic* and *up-monotonic* [DD, I.Wrosz MFCS 22])

Open problems

- main one: is there a polynomial-time constant factor approximation for weighted trees?
- analyze *edge search* instead?
- analyze restricted weight functions for trees (we already have solved *down-monotonic* and *up-monotonic* [DD, I.Wrosz MFCS 22])
- how about classes of graphs different than trees?

Open problems

- main one: is there a polynomial-time constant factor approximation for weighted trees?
- analyze *edge search* instead?
- analyze restricted weight functions for trees (we already have solved *down-monotonic* and *up-monotonic* [DD, I.Wrosz MFCS 22])
- how about classes of graphs different than trees?
- more real-world applications of such search, or theoretical connections?

Thank you!