Orienting (hyper)graphs under explorable stochastic uncertainty

Evripidis Bampis
Christoph Dürr
anonymous
Thomas Erlebach
Murilo Santos de Lima
Nicole Megow
Jens Schlöter

Motivating application

Different European countries need to choose the best vaccine for some new illness.

Each country imposes its own constraints, restricting the set of considered vaccines.

At European level minimize the total costs of the tests necessary to identify the best vaccines for each country.

Formally: G=(V,E), $E \subseteq 2^V$

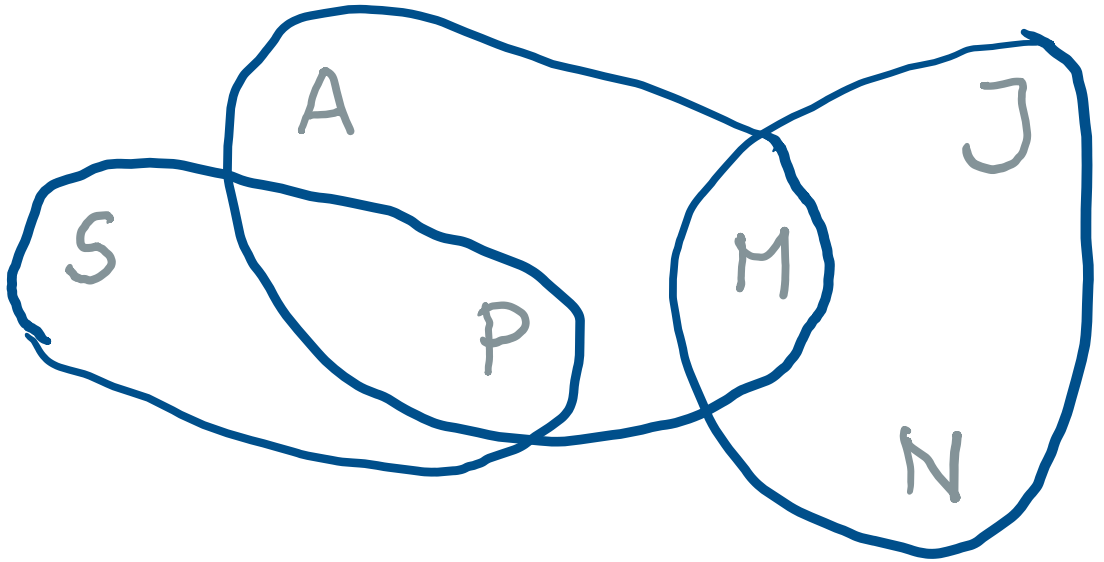
Every $v \in V$

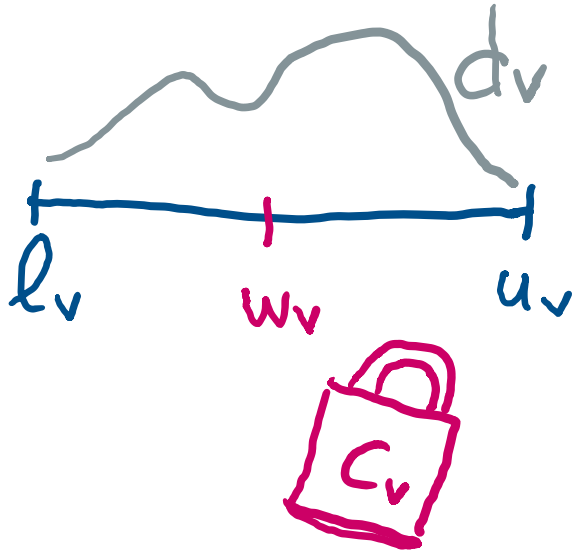
- has a hidden value $\ell_v < w_v < u_v$ drawn from a known distribution d_v
- and has a known query cost c_v

Goal: identify minimum value in each hyperedge

(Adaptive) algorithm = decision tree

Competitive ratio := expected cost of algorithm over expected optimal cost





Small example

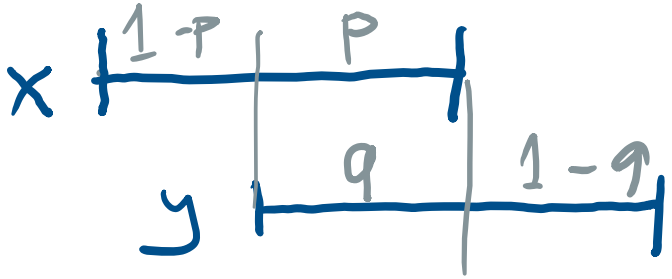
Optimum:

E[OPT] = 1 + pq

Algorithm 1

query x, then y if necessary

E[ALG1] = 1 + p



Algorithm 2

query y, then x if necessary

E[ALG2] = 1 + q

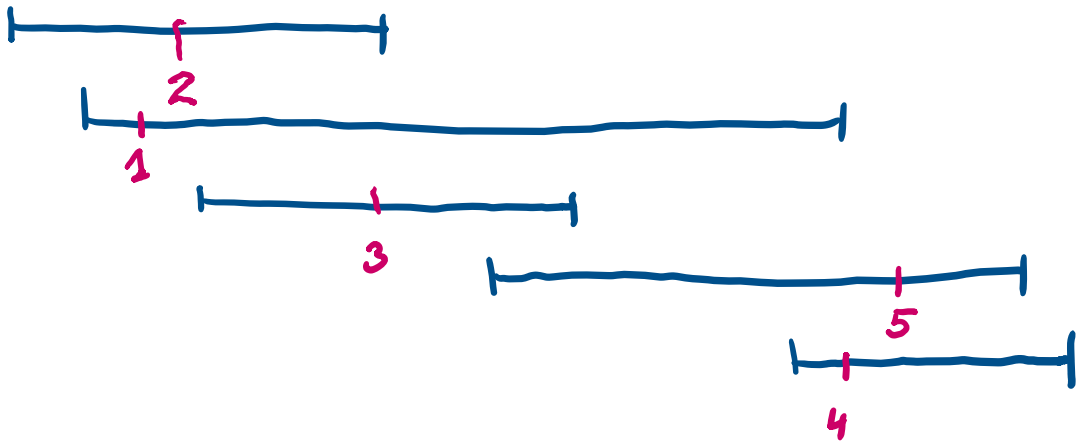
Competititive ratio = 1.207

Maximized for p=q=$\sqrt{2}-1$

previous work: Sorting

Orienting a complete graph (=sorting)
[Halldórsson, de Lima, 2019]
[Chaplick, Halldórsson, de Lima, Tonoyan, 2020]

- Cheapest decision tree can be computed in polynomial time
  ($O(n^6)$ by dyn. prog.)
- deterministic competitive ratio for uniform query costs is ≤ 1.5
- randomized competitive ratio for arbitrary query costs is ≤ 1.77



previous work: Finding the minimum of a single hyperedge (set)

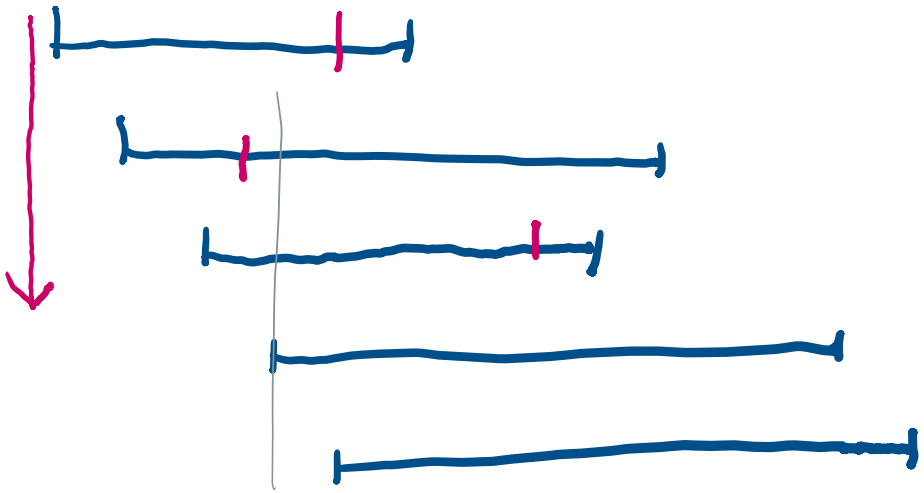
[Chaplick, Halldórsson, de Lima, Tonoyan, 2020]

- Provide a 1.5-competitive algorithm

Open: Compute optimal decision tree (NP-hard?)

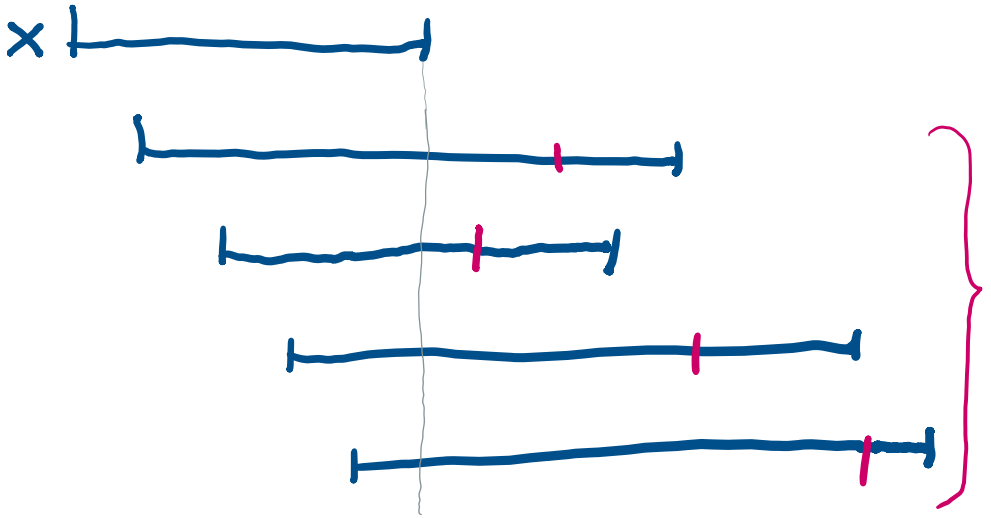
Two possible optimal query sets

Query vertices x in order of increasing lower bound ℓ_x until minimum is found



or

Query all but the vertex x with smallest lower bound ℓ_x, provided all queried weights are above u_x

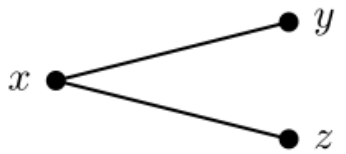


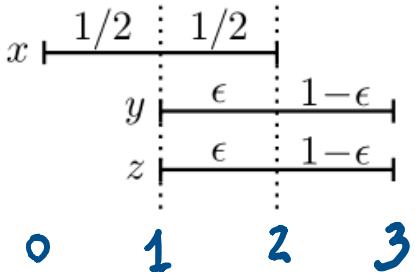
Graph orientation

Lower bound Competitive ratio $\geq \frac{4}{3}$
already for uniform query costs

OPT: cost$\approx 3/2$

$w_x \in (0,1]$?	Query only y	(proba ½)
else $w_y, w_z \in [2,3)$?	Query only y,z	(proba $(1-\epsilon)^2/2$)
else	Query x,y,z	(proba $\epsilon^2/2$)





ALG1: cost≈ 2

query x, $w_x \in (0,1]$? — Yes → Done; No → query y,z

ALG2: cost≈ 2

query y, $w_y \in (1,2)$? — Yes → Query x, $w_x \in (0,1]$? — Y → Done; N → query z

query y, $w_y \in (1,2)$? — No → query z, $w_z \in (2,3)$? — Y → Done; N → query x

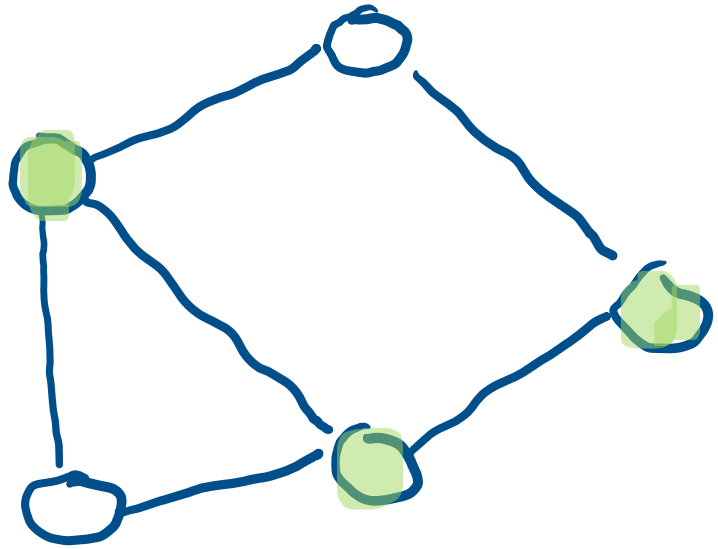
Graph orientation

Trivial fact: for every edge (x, y) we can assume that (ℓ_x, u_x) and (ℓ_y, u_y) intersect

At least one of x,y must be queried

Implications

- Every query set is a vertex cover
- There is a trivial 2-competitivealgorithm for uniform query costs



Vertex-cover based algorithm

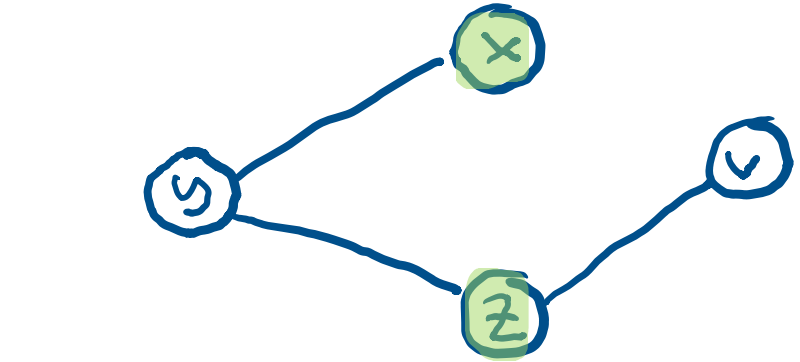
For a orienting a graph with arbitrary query costs

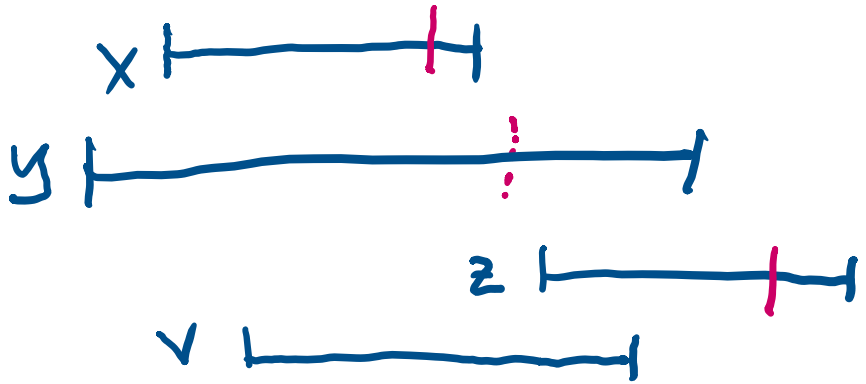
Vertex-cover based algorithm:

- Query non-adaptively a vertex cover
- While there are mandatory vertices:
  - Query a mandatory vertex

Competitive ratio $\geq 3/2$

(4/3 for bipartite graphs)





Vertex-cover based algorithm

Generalize to hypergraphs

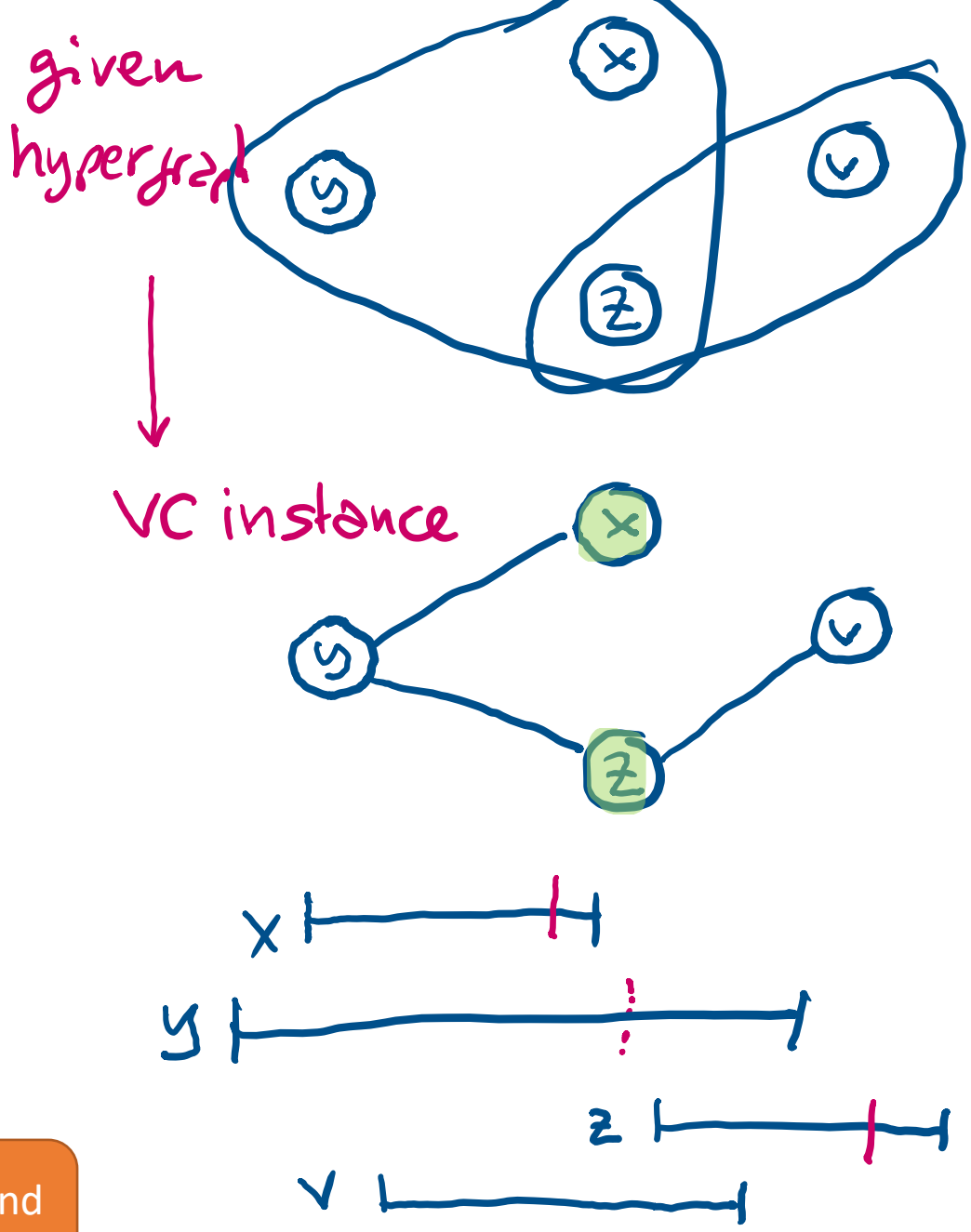
For a hyperedge $\{v_1, v_2, \dots, v_k\}$ with $\ell_{v_1} \leq \cdots \leq \ell_{v_k}$, for every $i = 2, \dots, k$ at least one of the vertices $\{v_1, v_i\}$ needs to be queried (is a witness set)

Vertex cover instance := graph obtained by replacing hyperedges $\{v_1, v_2, \dots, v_k\}$ with edges (v_1, v_i)

Vertex-cover based algorithm:

- Query non-adaptively a vertex cover of the vertex cover instance
- While there are mandatory vertices:
  - Query a mandatory vertex

Open: characterize vertex cover instance graphs and find good vertex cover approximations for them



Price of adaptivity

Vertex-cover based algorithm:

- Query non-adaptively a vertex cover of the vertex cover instance
- While there are mandatory vertices:
  - Query a mandatory vertex (adaptively)

Two-phase algorithm:

- Query a set S1
- Depending on outcome query a set S2

Non adaptive algorithm:

- Query a set S

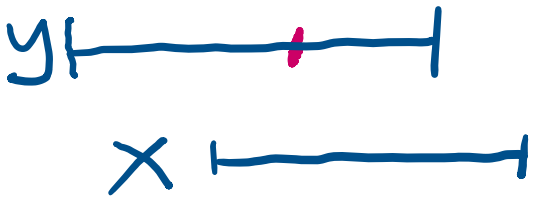
Class of algorithms	Competitive ratio already for a single set and uniform query costs
Vertex-cover based algorithm	$\geq 3/2$
Two-phase algorithm	$\Omega(\log n)$
Non-adaptive algorithm	n

Mandatory vertex

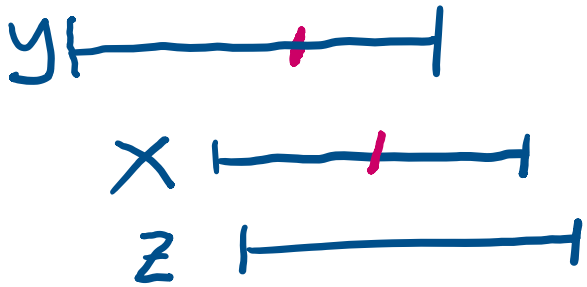
x is mandatory (belongs to every feasible query set), if here is a vertex y which has currently the minimum queried value for some hyperedge F containing both x and y and $w_y \in (\ell_x, u_x)$

p_x = probability that x is mandatory

Theorem p_v is #P-hard to compute for hypergraphs (reduction from #VertexCover) and can be approximated for graphs (sampling)



we need to query x in order to find out which of x,y is smaller



now z becomes mandatory

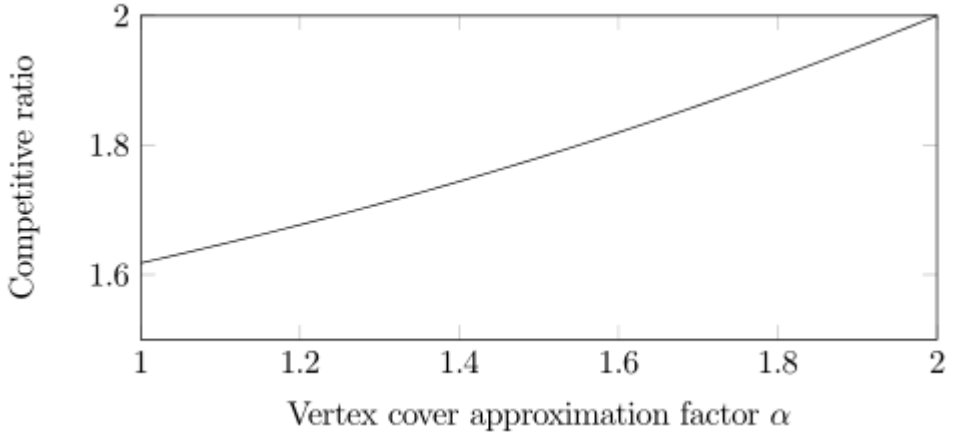
Threshold Algorithm

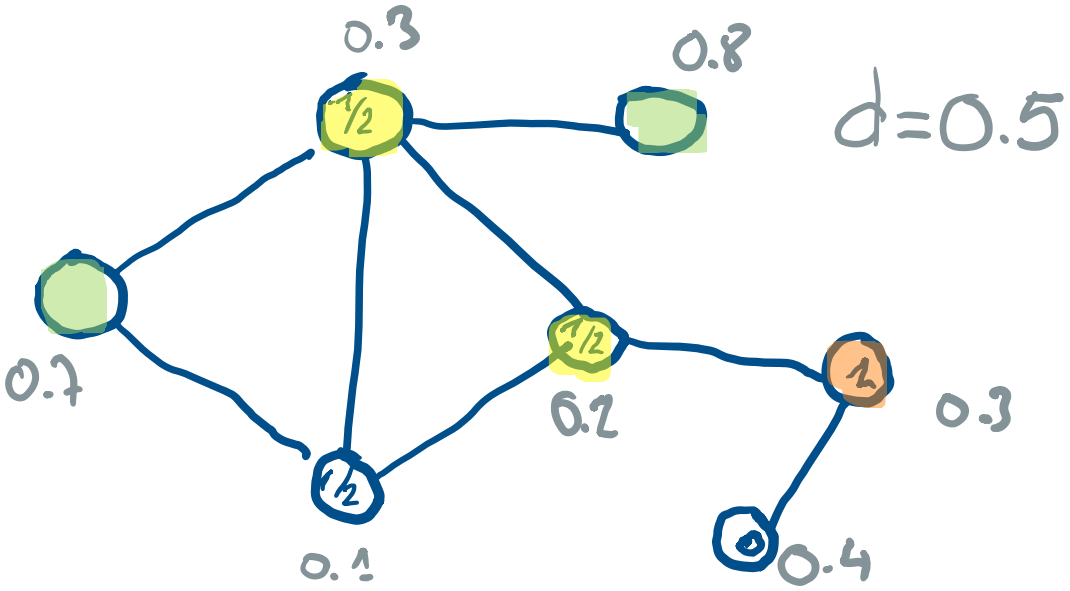
Threshold algorithm:

- Compute (approximatively for hypergraph) for every vertex v the probability p_v that v is mandatory
- Let $M = \{v: p_v \geq d\}$ for $d = 2/(\alpha + \sqrt{8 - \alpha(4-\alpha)})$
- Solve the vertex-cover linear program on G[V\M]
- Solution is half-integral and partitions V\M into V_0 $V_{½}$ V_1.
- Compute (α-approximatively) a vertex cover VC' for $G[V_{½}]$
- Query $M \cup V_1 \cup VC'$
- While there are mandatory vertices:
  - Query a mandatory vertex (adaptively)

Competitive ratio

$$= \frac{1}{2}(\alpha + \sqrt{8 - \alpha(4-\alpha)})$$





Competitive ratios at a glance

	Uniform costs	General query costs
Complete graph (sorting) [Halldórsson, de Lima, 2019]	$\leq \frac{3}{2}$ (randomized)	≤ 1.7698 (randomized)
Bipartite graph	=4/3	=4/3
General graph α=approx. ratio of VC instance	$\geq \frac{3}{2}$	$\leq \frac{1}{2}(\alpha - \sqrt{8 - \alpha(4-\alpha)})$
Hypergraph		Similar (up to approximating probabilities being mandatory)
Single set [Chaplick,et al,2020]	$\leq \frac{n}{n-1}$ $\geq n^2/(n^2 - n + 1)$	$= \frac{3}{2}$
Single edge	=1.207	