Distributed Complexity of Lovász Local Lemma

Based on [Brandt,Maus,Uitto + Brandt,Grunau,R]

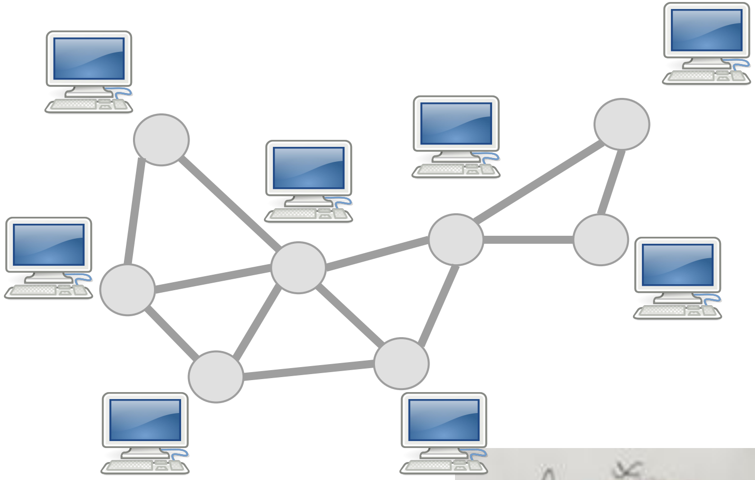
The LOCAL model of distributed graph algorithms

- Undirected graph on n nodes
- One computer in each node

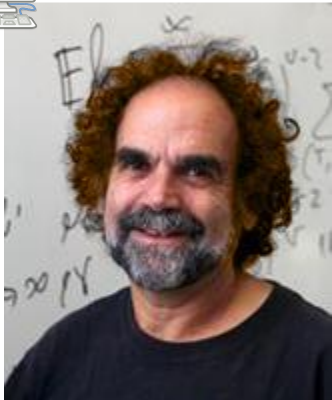
- Synchronous message passing rounds
- Unbounded message size and computation

- Initially, nodes know only (upper bound on) n and their unique $O(\log n)$ bit identifier
- In the end, each node should know its part of output

- Time complexity: number of rounds



LOCAL model
[Linial FOCS'87]



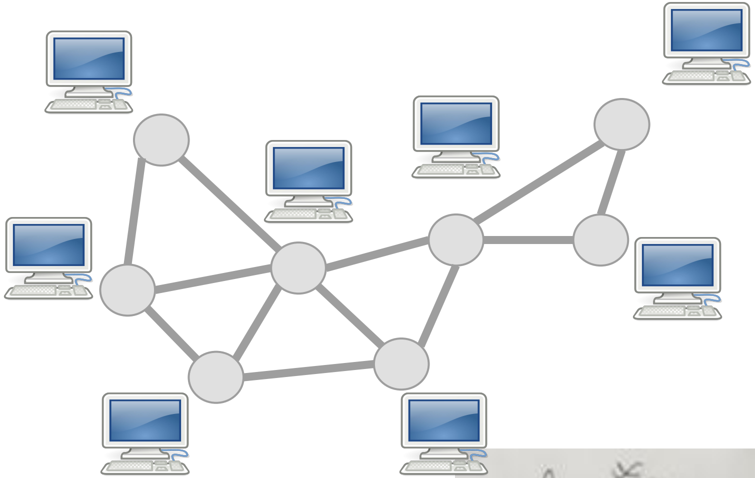
The **LOCAL** model of distributed graph algorithms

- Undirected graph on n nodes of constant degree
- One computer in each node

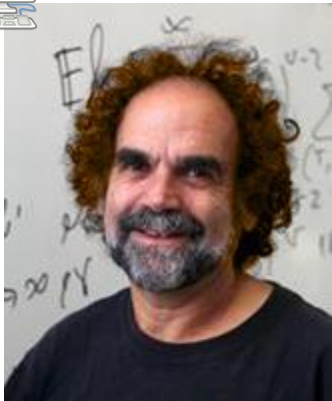
- Synchronous message passing rounds
- Unbounded message size and computation

- Initially, nodes know only (upper bound on) n and their unique $O(\log n)$ bit identifier
- In the end, each node should know its part of output

- Time complexity: number of rounds



LOCAL model
[Linial FOCS'87]

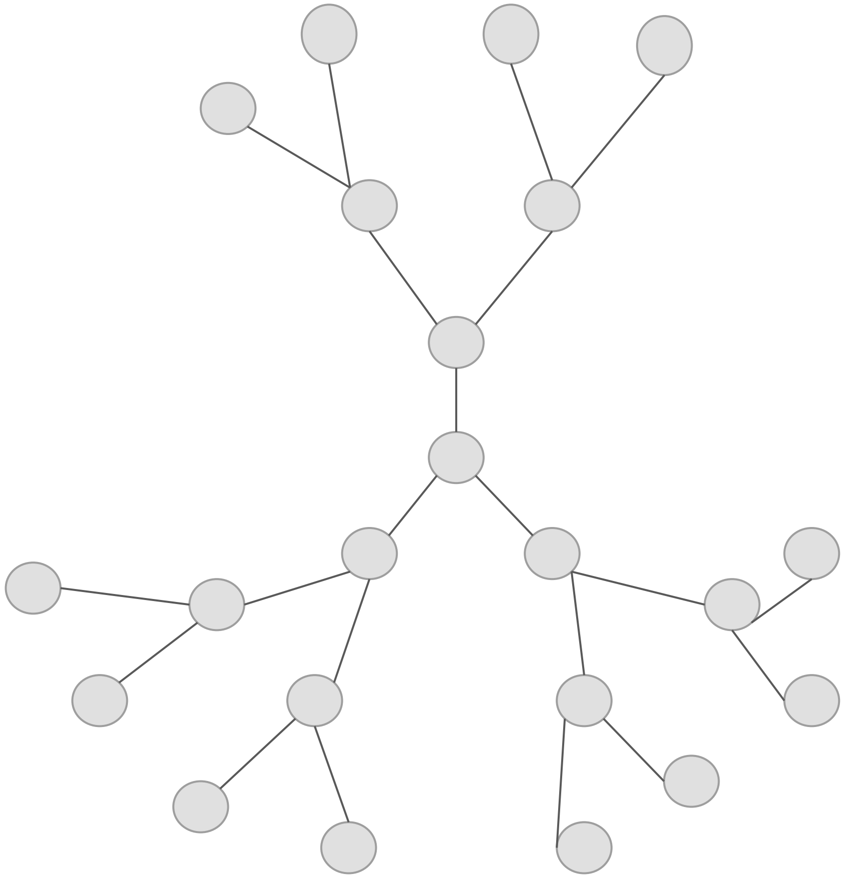


Sinkless orientation problem

Orient the edges of a graph so that every vertex has at least one outgoing edge.

Think of Δ-regular graphs for large constant Δ.

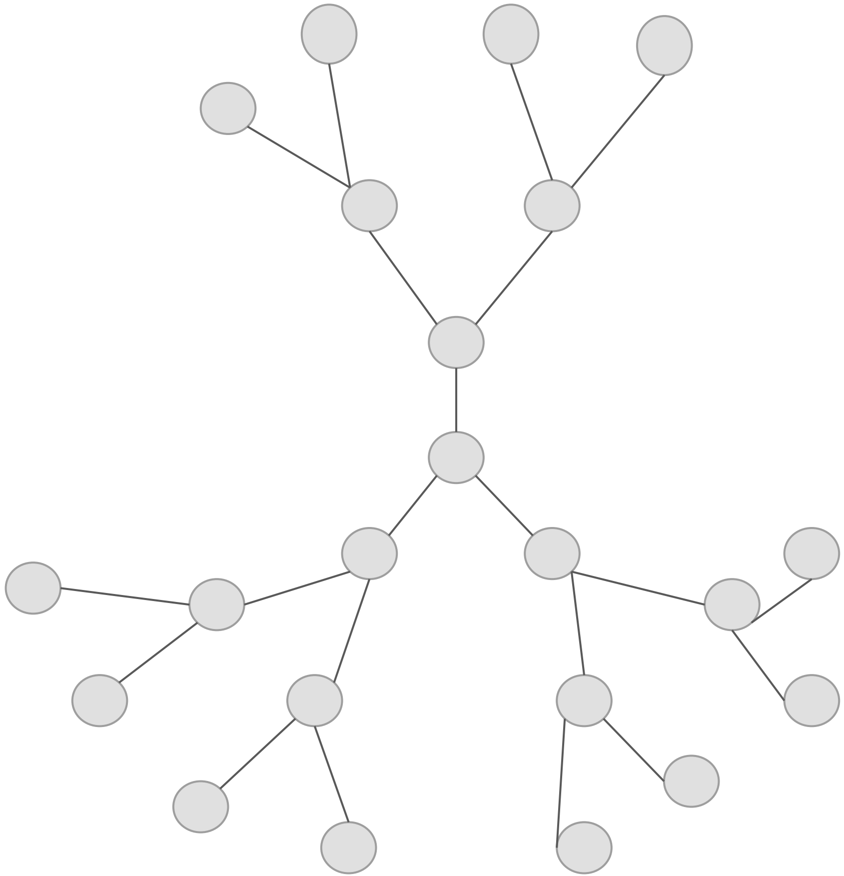
Vertices of degree less than Δ are not required to output anything



Sinkless orientation problem

How to approach this?

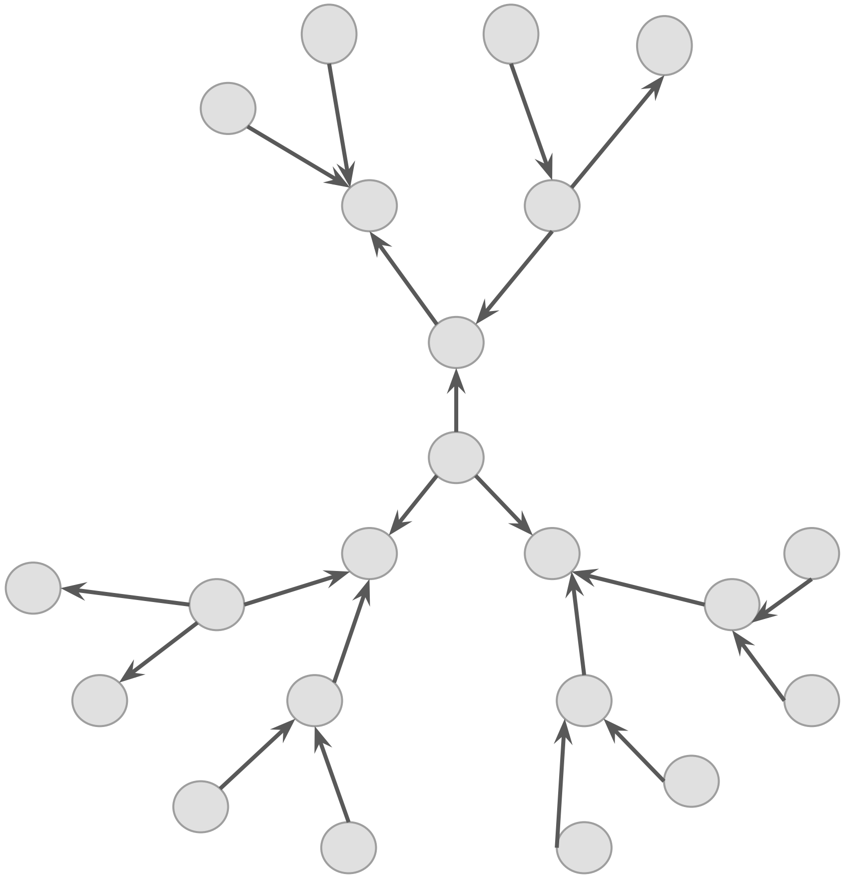
Random orientation is not so bad: a vertex has probability only $2^{-\Delta}$ that it is not solved.



Sinkless orientation problem

How to approach this?

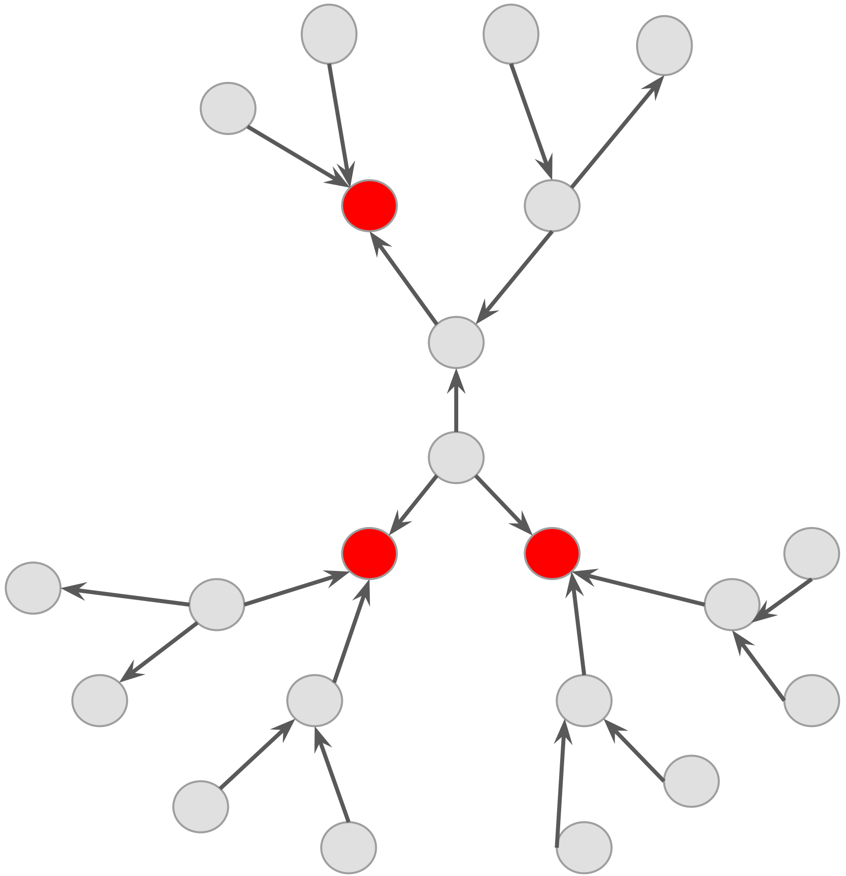
Random orientation is not so bad: a vertex has probability only $2^{-\Delta}$ that it is not solved.



Sinkless orientation problem

How to approach this?

Random orientation is not so bad: a vertex has probability only $2^{-\Delta}$ that it is not solved.



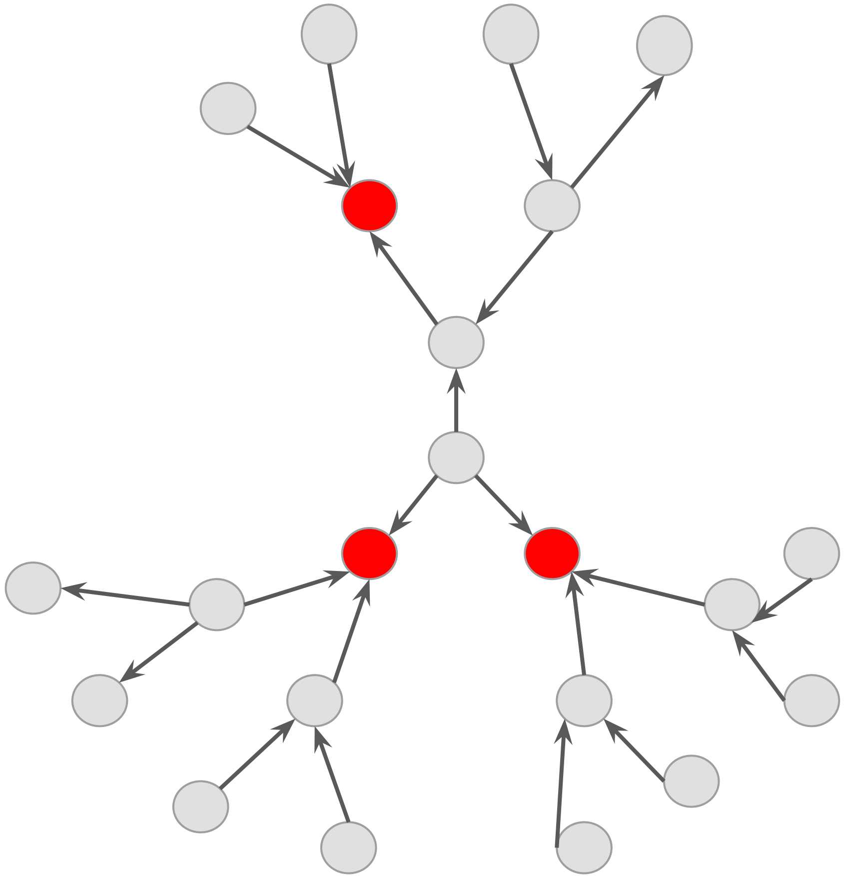
Sinkless orientation problem

How to approach this?

Random orientation is not so bad: a vertex has probability only $2^{-\Delta}$ that it is not solved.

Also, the bad events are independent for non-neighboring vertices.

This setup is known as Lovász local lemma (in general, each random choice can affect more than two bad events).

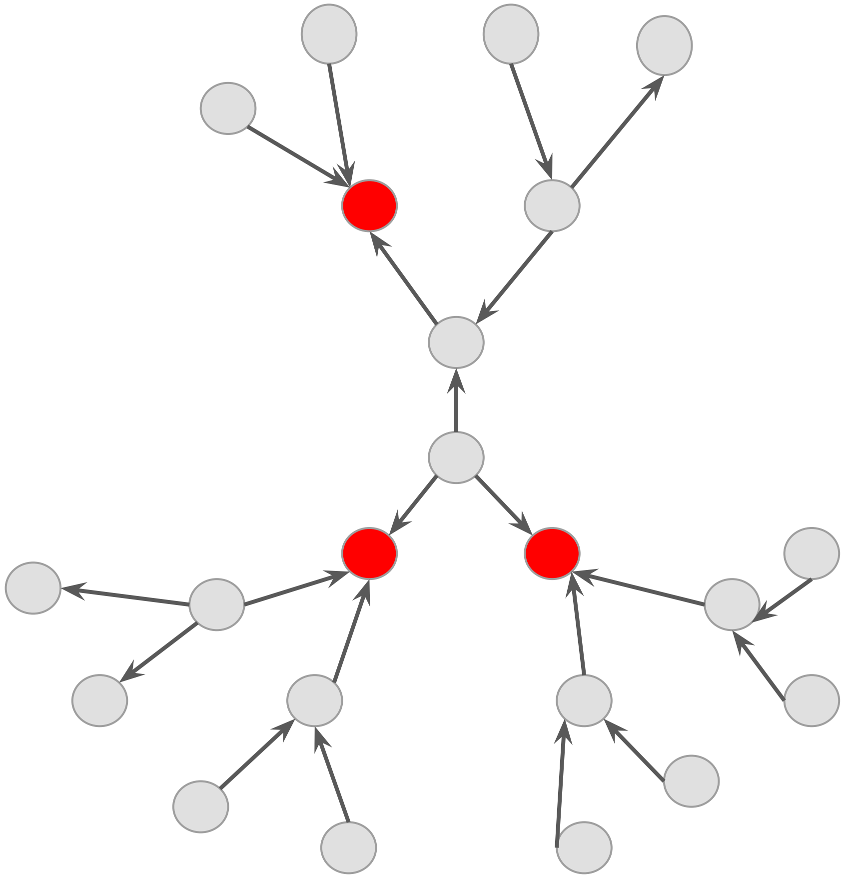


Lovász local lemma

Every such problem can be solved in poly(log n) rounds deterministically and poly(log log n) rounds randomized, given $p = 1/\text{poly}(\Delta)$ [Fischer,Ghaffari; Ghaffari et al.; Ghaffari et al.]

On the other hand, $\Omega(\log n)$ deterministic and $\Omega(\log \log n)$ randomized lower bound for sinkless orientation (remember that here $p = 2^{-\Delta}$). [Brandt et al.]

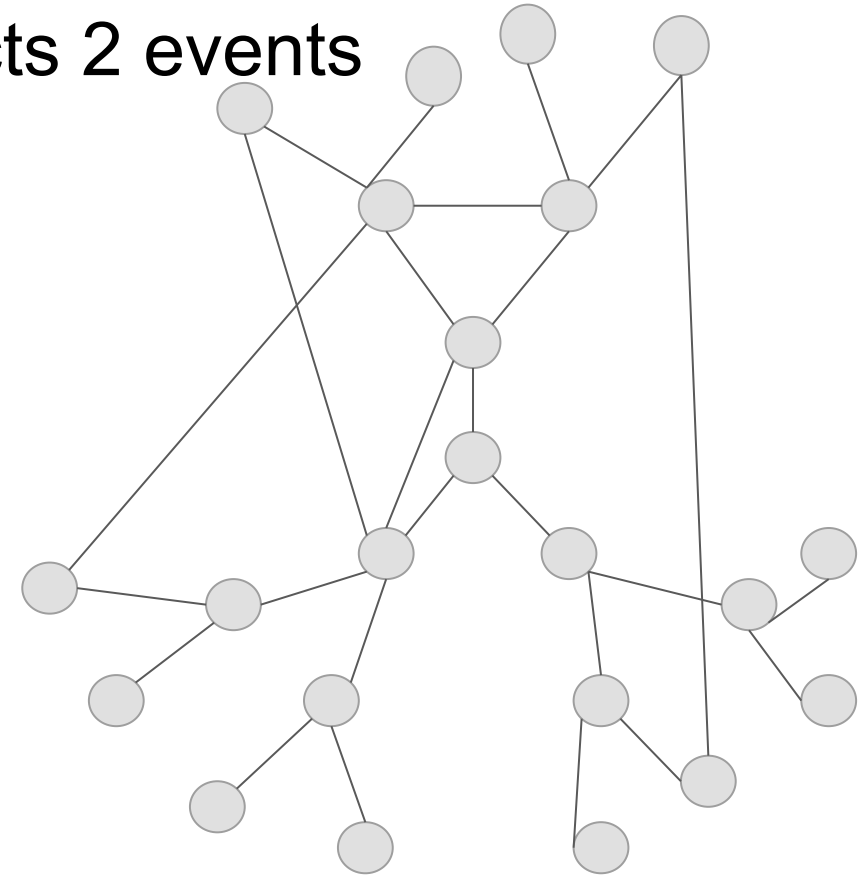
Today: for $p < 2^{-\Delta}$ the det./rand. complexity is $\Theta(\log^* n)$. [Brandt et al.; Brandt et al.]



Warmup: each variable affects 2 events

A sequential approach: iterate over edges and each time fix their randomness greedily. (To get a distributed algorithm, compute edge coloring in $\Theta(\log^* n)$ rounds first.)

Goal: each time we fix an edge, the probability of bad events on the two endpoints increases by at most a 2-factor.

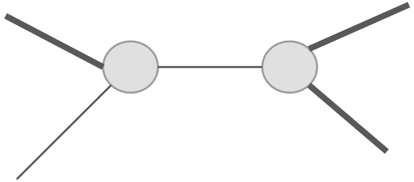


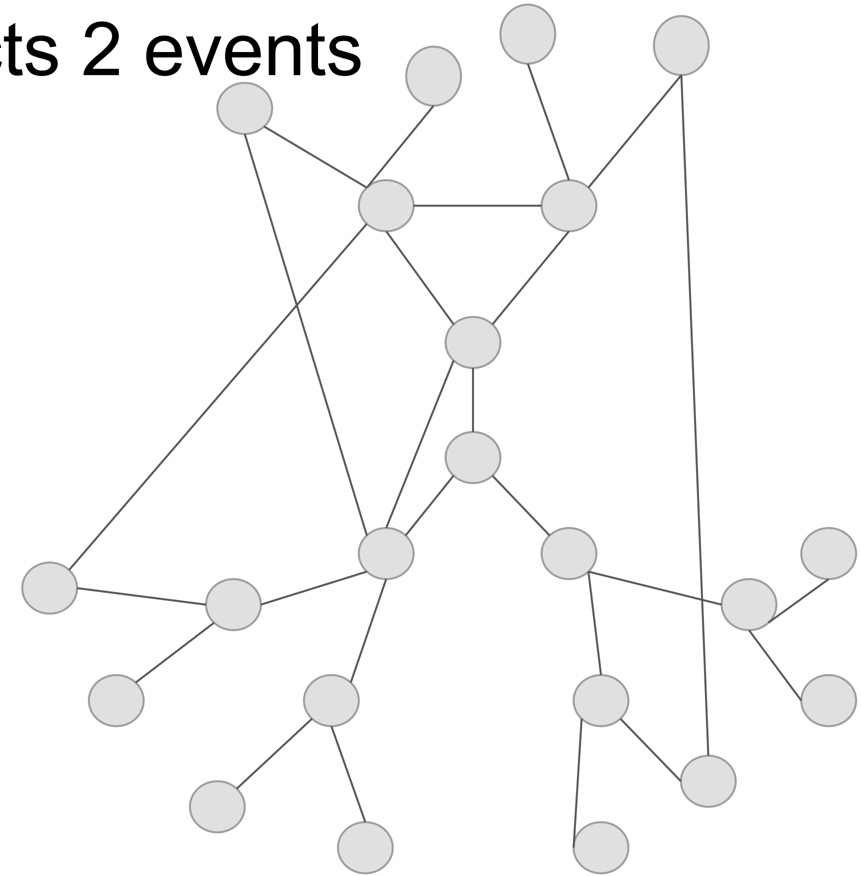
Warmup: each variable affects 2 events

A sequential approach: iterate over edges and each time fix their randomness greedily. (To get a distributed algorithm, compute edge coloring in $\Theta(\log^* n)$ rounds first.)

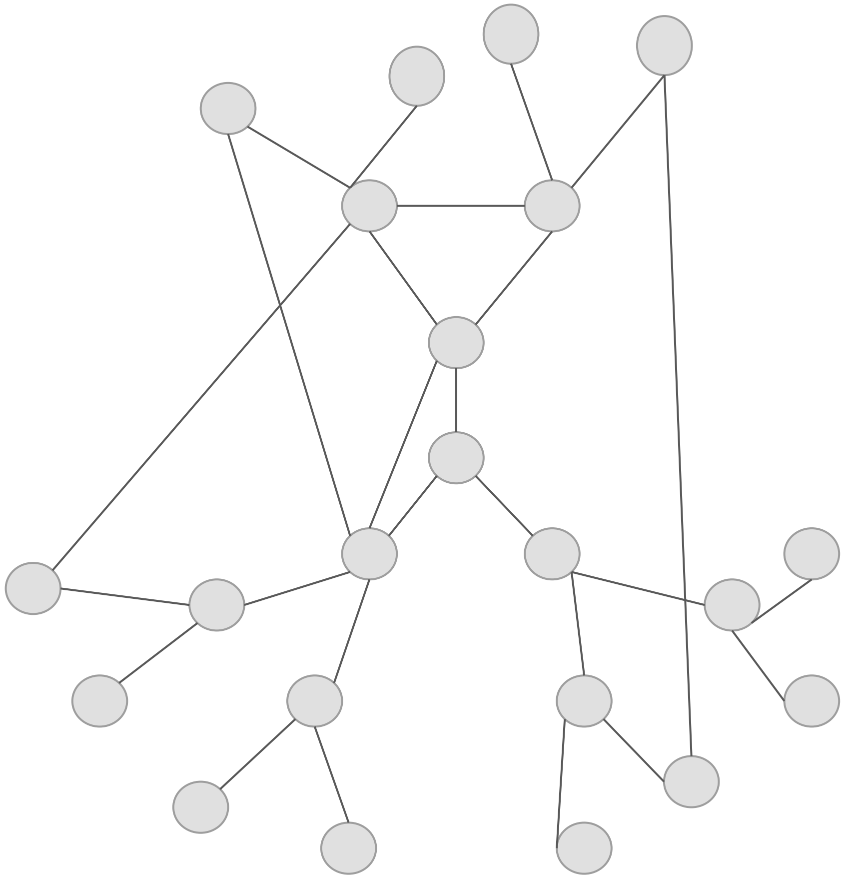
Goal: each time we fix an edge, the probability of bad events on the two endpoints increases by at most a 2-factor.

Holds by Markov inequality.





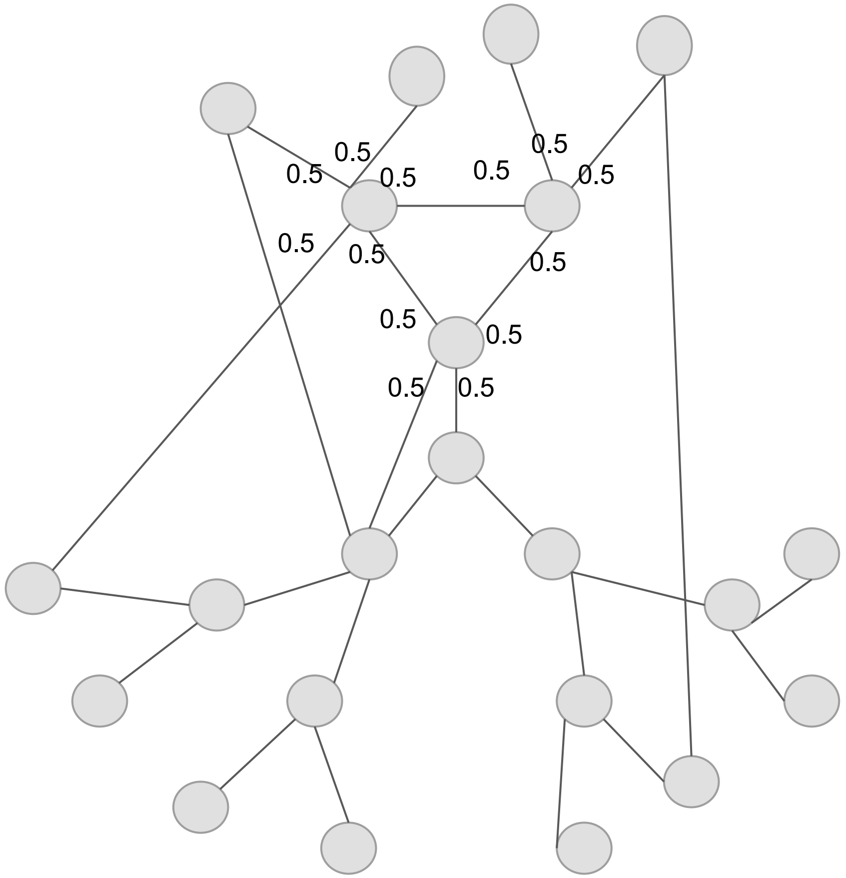
General case



General case

An algorithm that preserve two invariants:

- There are two nonnegative numbers on each edge summing up to ≤1,
- the probability of a bad event is smaller than the product of numbers next to the respective vertex.



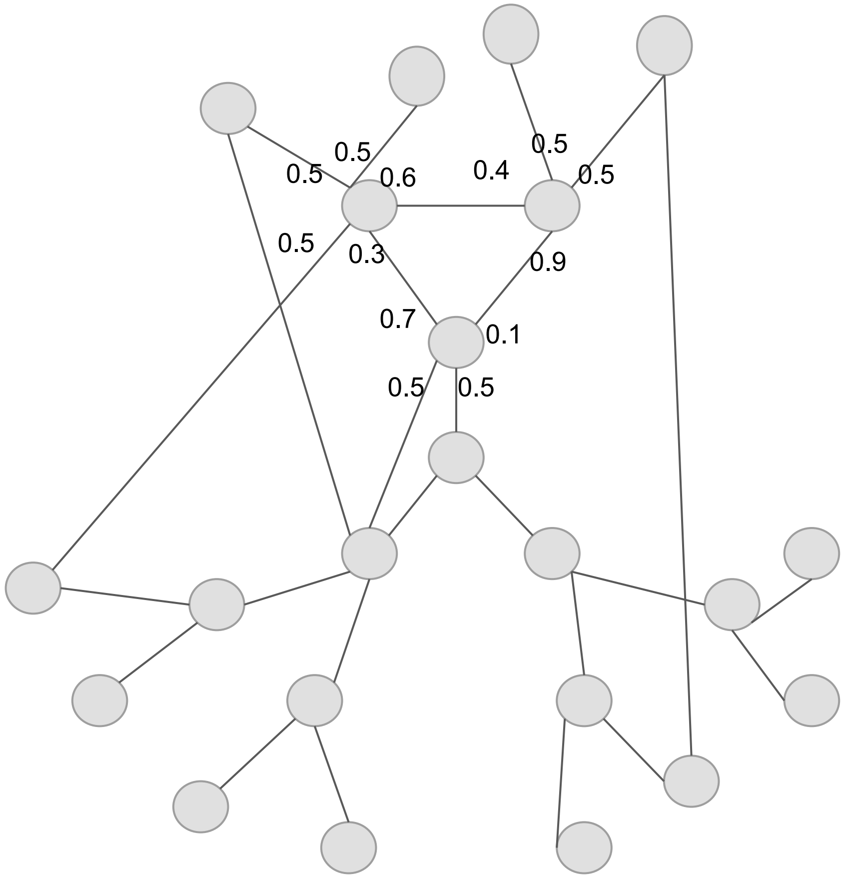
General case

An algorithm that preserve two invariants:

- There are two nonnegative numbers on each edge summing up to ≤1,
- the probability of a bad event is smaller than the product of numbers next to the respective vertex.

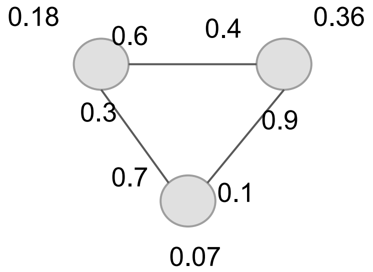
Goal: we can always derandomize so as to preserve this invariant.

This is just an existential question!



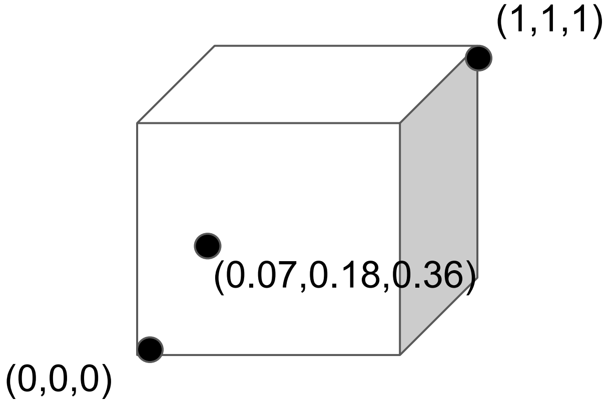
What needs to be proven (in case an event affects 3 variables)

Given a triangle labelled with 6 nonnegative numbers summing to ≤1 on each edge.



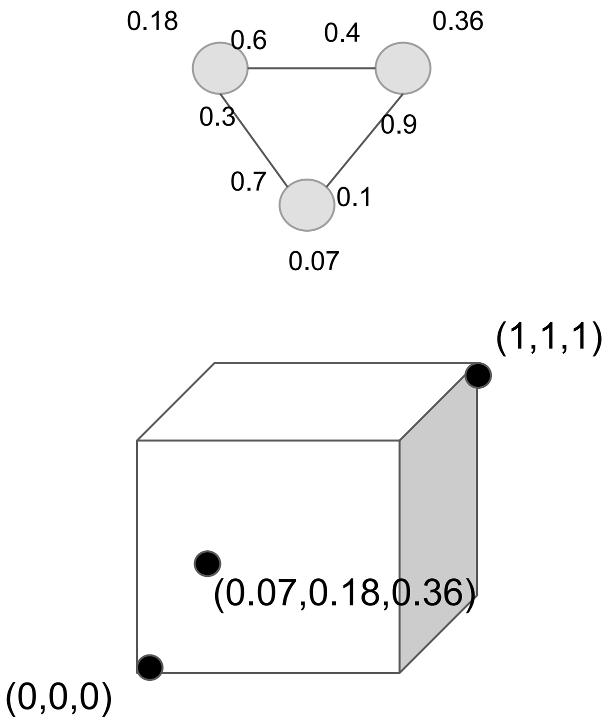
Multiply numbers around each vertex; let us call each triple we can get this way representable.

Prove that all nonrepresentable triples from $[0,1]^3$ form a convex set.



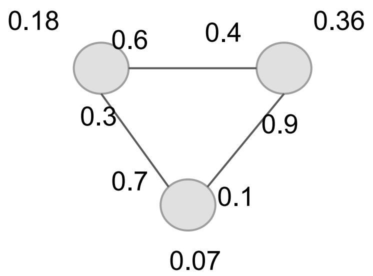
How to prove a set is convex?

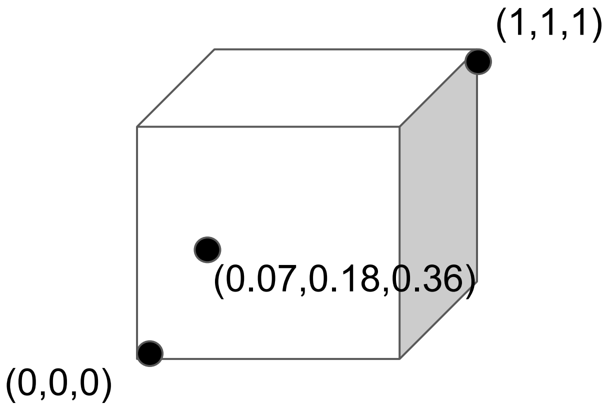
- differentiate everything (works for ≤3 events, but not clear for higher dimensions)
- take two nonrepresentable triples, prove their convex combination is not representable
- take a representable triple at the boundary and find a supporting hyperplane, i.e. a plane containing just representable triples



Finding supporting plane

Let’s start by finding a supporting line.

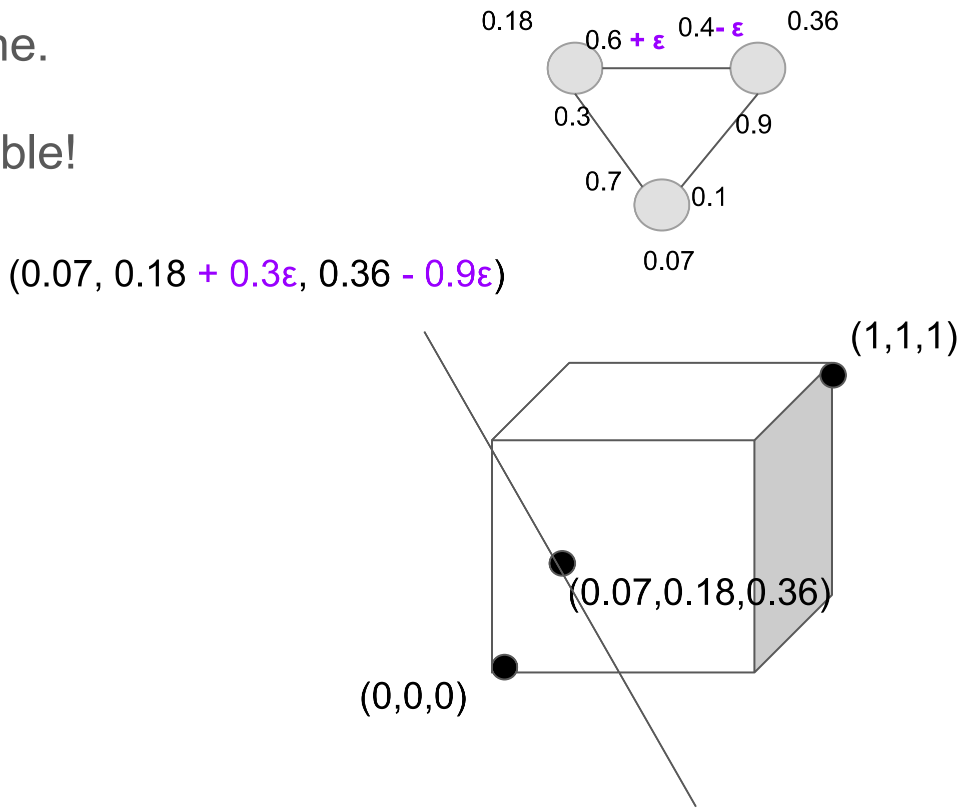




Finding supporting plane

Let's start by finding a supporting line.

All triples on the line are representable!

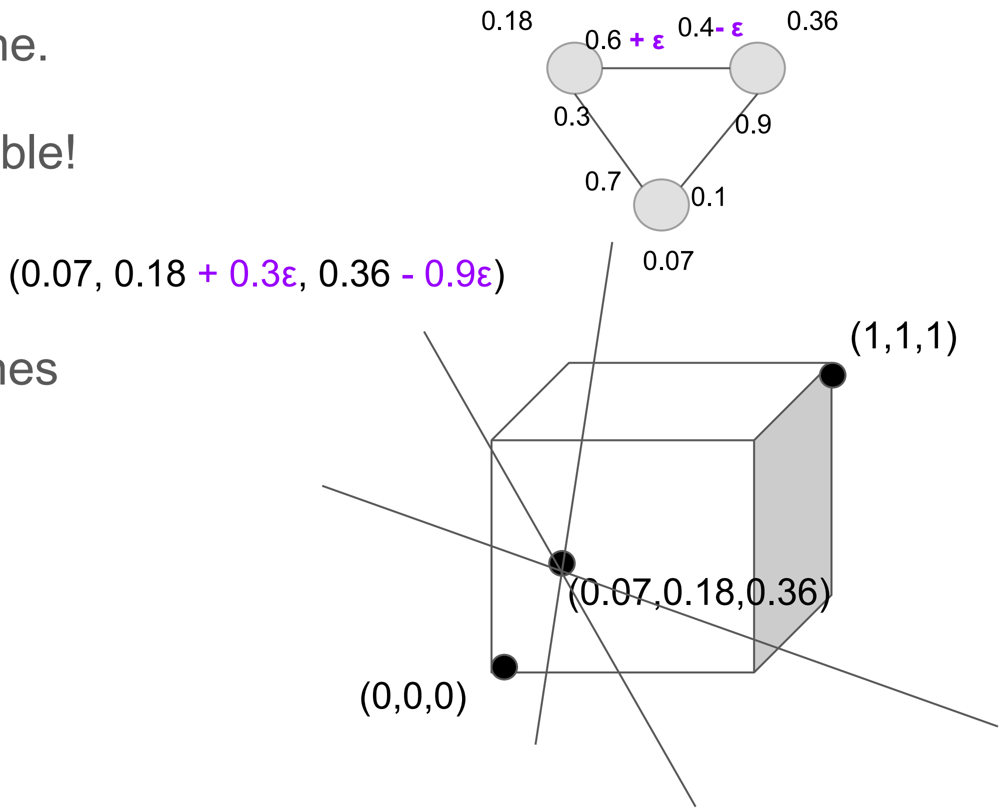


Finding supporting plane

Let's start by finding a supporting line.

All triples on the line are representable!

Even better, there are even three lines containing representable triples.



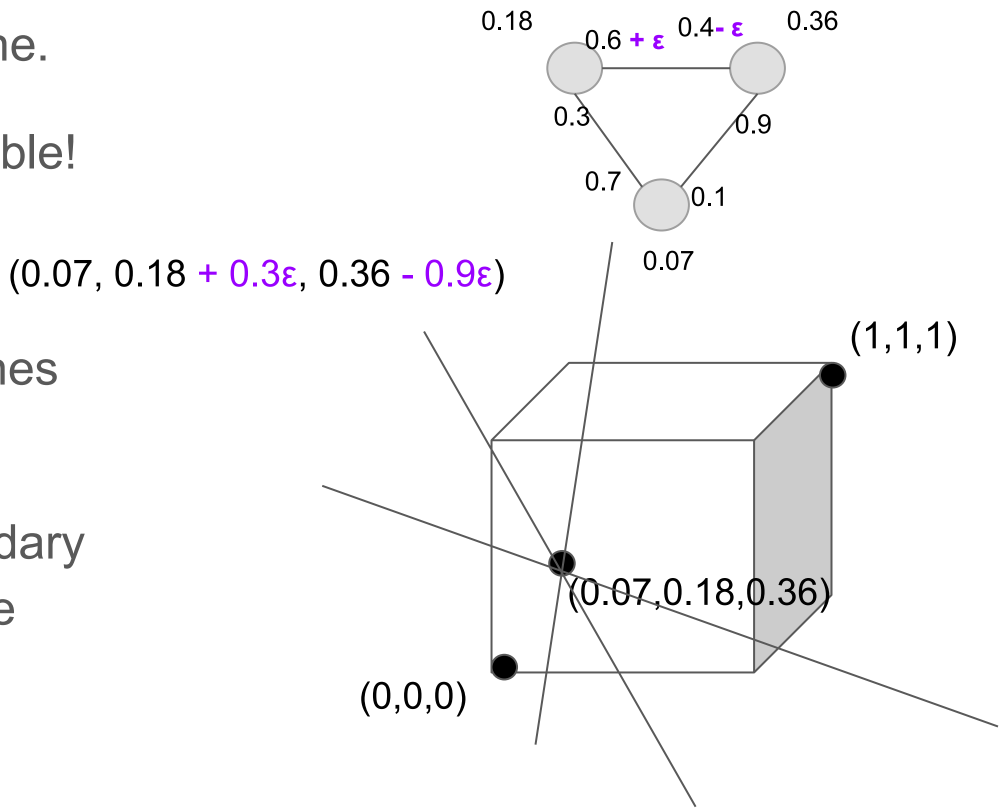
Finding supporting plane

Let's start by finding a supporting line.

All triples on the line are representable!

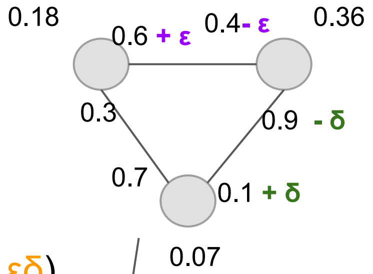
Even better, there are even three lines containing representable triples.

If the starting triple lies on the boundary of the representable set, these three lines define a common plane H.

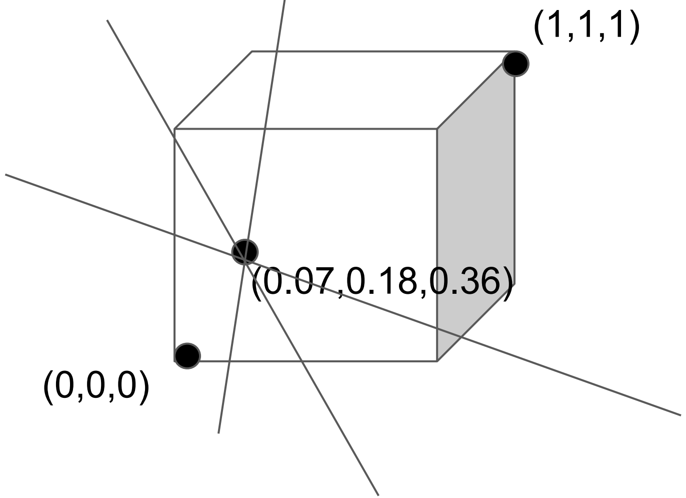


Finding supporting plane

Up to the orange term, we defined a representable plane.



$(0.07 + 0.7\delta,\ 0.18 + 0.3\varepsilon,\ 0.36 - 0.9\varepsilon - 0.4\delta + \varepsilon\delta)$



We actually have three possible coordinate systems for the plane H.

One can prove that for each $x \in H$, one choice of basis yield positive orange term.

A nice fact

body = compact subset of R^n with nonempty interior

A body is convex if and only if one can find a supporting hyperplane at each boundary point.

A connected body is convex if and only if one can find a locally supporting hyperplane at each boundary point.