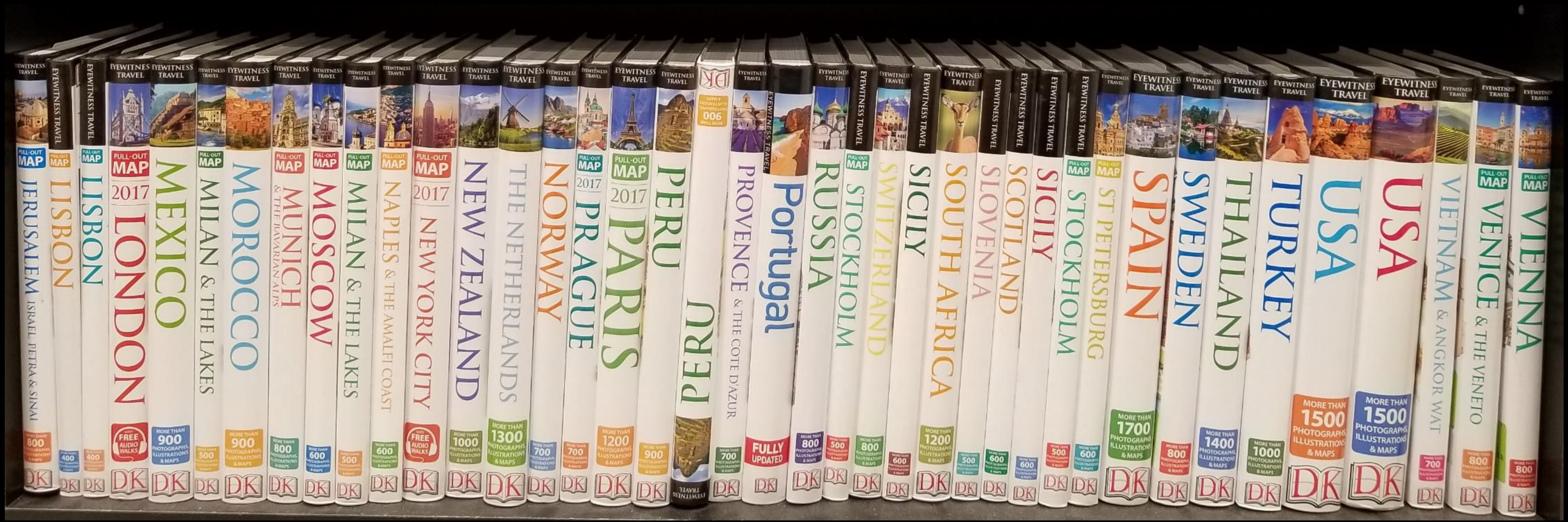
A (Centralized) Local Guide

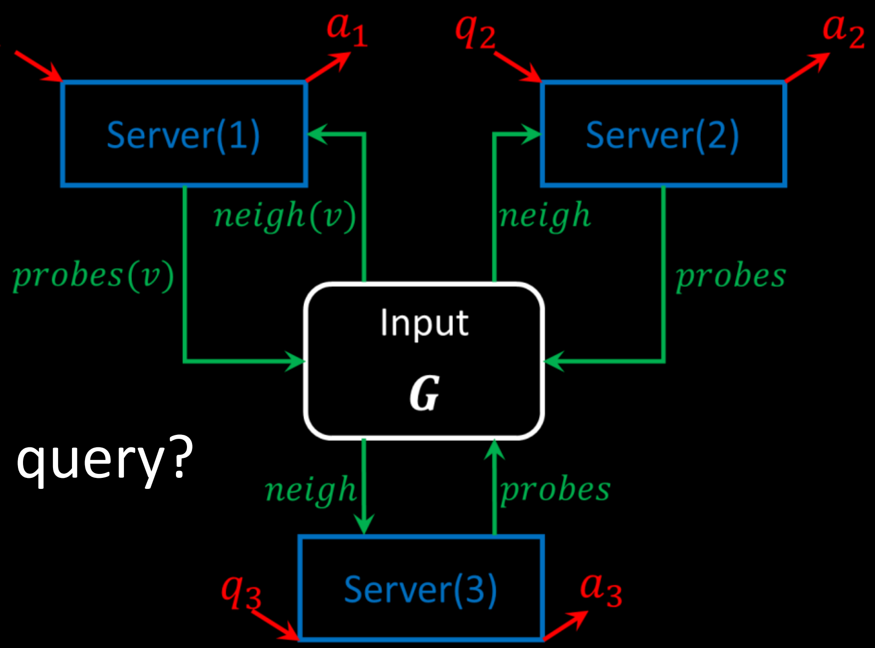
Moti Medina

ECE, Ben-Gurion University of the Negev



Motivation for Centralized Local Algs [Rubinfeld, Tamir, Vardi, and Xie 2011]

- Coping with "large" inputs
  - "Read" small portion of the input
- Coping with "large" outputs
  - Access part of output via queries.
  - Can we probe "small" parts of the input to answer a query?
- **Goal: sublinear number of probes per query.**
- Uncoordinated Servers (i.e., no comm)
  - Consistency
  - Stateless ⇒ no need for comm. (answers ind. of server)



Example: Maximal Independent Set

- Fix a graph $G = (V, E)$
  - Input: Sequence of queries $v_1, v_2, \ldots \in V$.
  - Output: Answer each query: Does v_i belong to $MIS(G)$?

- Required properties:
  - Cent. Local pretends to know a specific solution,
  - All the answers are based on the same solution,
  - No preprocessing,
  - Few probes per query,
  - No need to store info about previous queries/answers.

Other Local Models

- Distributed Local Model
  - input spread among network vertices
  - local communication & computation ($\#rounds = o(diameter)$)
- Property Testing
  - Access input via probes.
  - Output: YES/NO.
- Sub-linear approximation algorithms
  - Access input via probes.
  - Output: apx the size of the optimal solution.
- …

Outline

- Model
- Connections
- Techniques
- State-of-the-art Algs
- Local Graph Generators

Based on the survey “A (Centralized) Local Guide” by Reut Levi and Moti Medina

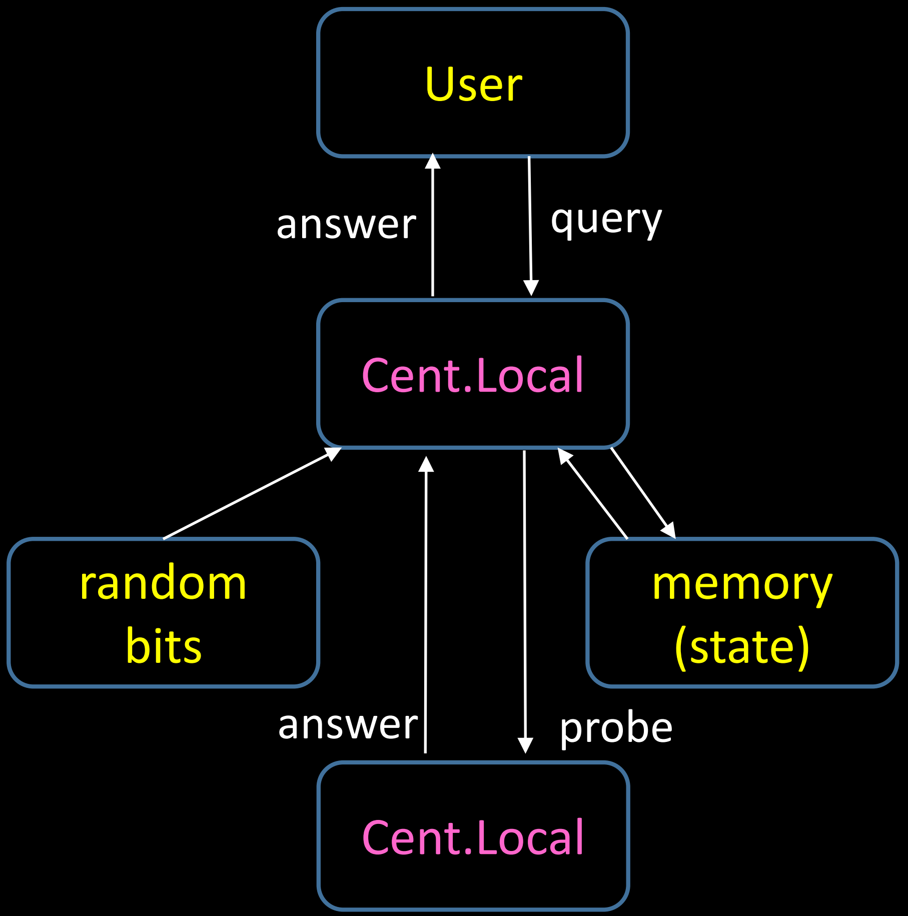
The Cent. Local Model

Centralized Local Algorithm
[Rubinfeld, Tamir, Vardi, Xie 2011, Alon, Rubinfeld, Vardi, Xie 2012]

Π – computation problem over a labeled graph G.

Given a probe access to G, the local algorithm:

- Provides query access to $f \in Sol(G, \Pi)$.
- Consistent with the same f
  - f is determined by G and internal randomness.
- For each oracle query, use small (sub.lin.) number of probes to G.
- (Sometimes the w.c. running time per query is also measured)



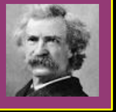
Query-Order-Oblivious [⋆] vs. Stateless [EMR14] Cent. Local Algs

- Query-Order-Oblivious
  - Global solution does not depend on the input sequence of queries.
- [Even, M, Ron 14]: $Stateless$ is $Query - order - oblivious$

[Göös, Hirvonen, Levi, M, Suomela 2016] Observation:

- $Query - order - oblivious$ can be sim. by $Stateless$
- $\Rightarrow Stateless = Query - order - oblivious$
- Also show that $Stateful \neq Stateless$
  - Variant of leader election
  - $O(\log n)$ state size
  - $\frac{Probe\ complexity(stateless)}{Probe\ complexity(stateful)} = \Omega(n)$

Our Motto: *If you tell the truth, you don't have to remember anything.[MT]*



We focus on: Stateless algs

Connections to Other Models

Distributed Algs,

Property Testing,

Sublinear approximation algorithms

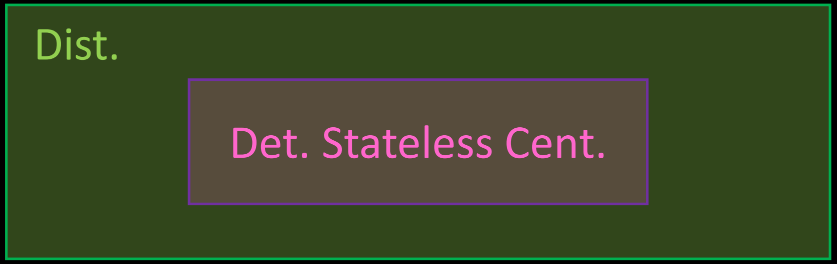
Centralized Local Algorithms vs. Distributed Local Algorithms

- Centralized Local Algorithms vs. Distributed Local Algorithms
  - Centralized: directly probe any part of the input.
  - Distributed: nodes communicate with their neighbors.

[Rubinfeld, Tamir, Vardi, Xie 2011, Alon, Rubinfeld, Vardi, Xie 2012] ⋆

[Linial 1992, Peleg 2000] ≡

- Is Cent. Local Algs $>$ Dist. Local Algs ?

- [GHLMS 16] show that stateless Cent. $=$ Dist.
  - For a large class of graph problems.
  - 

  - $\Rightarrow$ Transfer lower bounds from Dist. to Cent. !

“Non-Local Probes Do Not Help with Many Graph Problems” by
Goos, Hirvonen, Levi, M, Suomela
(DISC 2016)

Cent. Local Algs

- (“Shared” randomness.)
- IDs are known
  - Assume $\mathrm{IDs} = \{1, \dots, n\}$
- Known bounded degree Δ.
- Each $v \in \mathrm{V}$ is labeled with $\ell(v) \in \Sigma$.
- Structure of the input graph $G = (V, E)$ is unknown.
- Alg. Access $G = (V, E)$ via probes
  - Probe: Who are the neighbors of v_8?
  - Answer: $\{v_1, v_{20}, v_{9000}\}$
- User interface:
  - Input: User Query q (e.g., $v \in MIS_G$?)
  - Output: consistent $f(q)$ (e.g., Yes/No)
  - Desired property *Query Order Oblivious*
- Resources:
  - State size, (Random Seed), Computation is “for free”
- Complexity measure: #probes, State size, Seed length
  - Typically $o(n)$.

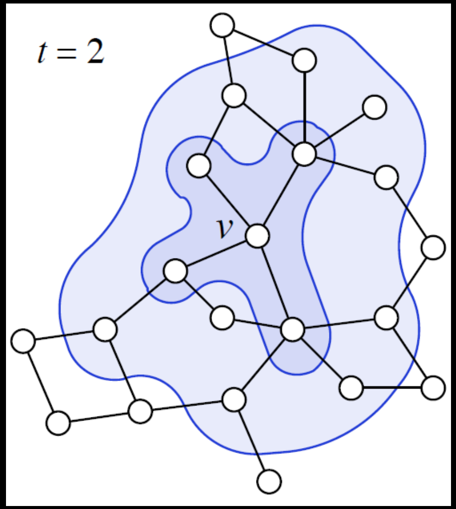
Dist. Local Algs

- (Private randomness.)
- IDs are unknown.
  - n processors
- Bounded degree Δ unknown.
- Each $v \in \mathrm{V}$ is labeled with $\ell(v) \in \Sigma$.
- Structure of the input graph $G = (V, E)$ is unknown.
- Each processor communicates with its neighbors in synch. rounds.
  - Each round, each processor: Sends messages, receives messages, performs local computation.
- After termination each processor q known its own part of the output $f(q)$.
- Resources:
  - Computation is “for free”.
- Complexity measure: #rounds
  - Typically $o(Diamter)$.

What can be Explored?

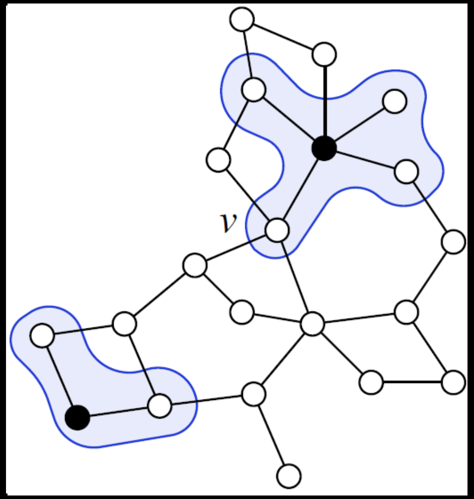
Dist. Local

- After t rounds, processor v knows $N_t(v)$.



Cent. Local

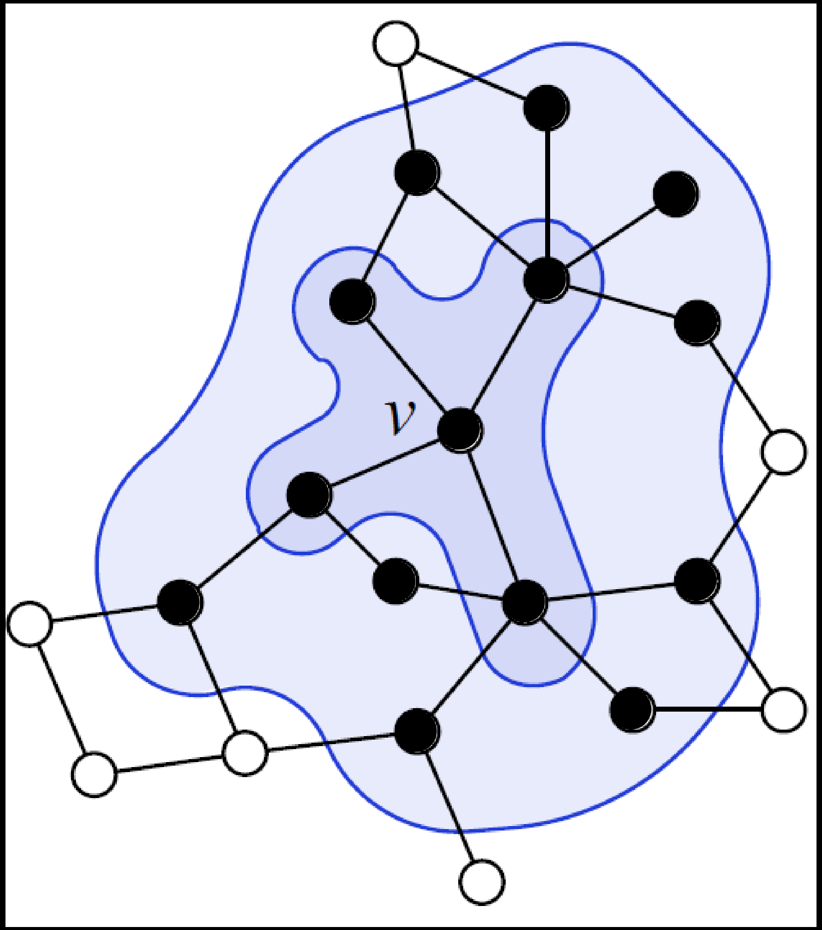
- After t probes, the alg queried on v knows $\{v\} \cup \bigcup_{i=1}^{t} N_1(p_i)$.



From Dist. Local to Cent. Local [Parnas-Ron 07]

- Simulate Dist. Local in Cent. Local.

Det. Dist. Local alg with t rounds $\Rightarrow$
Det. Cent. Local alg with $O(\Delta^t)$ probes.



From Cent. Local to Dist. Local ? In general, Impossible!

<u>Binary consensus:</u>

- Local input $\{0,1\}$
- Output:
  - $\exists\, u\, \forall\, v : \; ALG(v) = \ell(u)$
    - All nodes need to output the same output.
    - The output should equal to (at least) one node.
- Cent. Local: $\forall\, q : ALG(q) = \ell(1)$. 1 probe…Easy…
- Dist. Local: $\Omega(n)$ rounds!
  - 0000 0000
  - 1111 1111
  - 0000 1111

Nice Graph Problems

- Bounded degree Δ.
- Defined over labeled graphs $G = (V, E)$
  - $V = \{1, \dots, n\}$
- Given $\pi: \{1, \dots, n\} \to \{1, \dots, n\}$, the relabeling of G is $G^\pi = (V, E^\pi)$,
  - $E^\pi = \{\{\pi(u), \pi(v)\} \mid \{u, v\} \in E\}$
- Set of Solutions:
  - $\forall\ P \in \mathrm{NICE}\ \ \forall G : Sol(G, P) \subseteq Domain(G, P) \to Range(G, P)$
  - E,g., for MIS: $Sol(G, P) \subseteq V \to \{0,1\}$.
- Invariant under permutation π.
  - $\forall \pi: Sol(G, P) \circ \pi = Sol(G^\pi, P)$
- Every solution for G is also a solution when restricted to each connected component.
  - Binary consensus is not nice (0000,1111).
- Includes: LCL on bounded degree graphs, minimum spanning forest, MaxIS, MinDS, MinVC, $\Delta + 1$ coloring, MaxM, edge coloring,…, APX MCM, APX MWM, APX VC,…

Main Result: Simulating Cent. Local in Dist. Local

Thm.

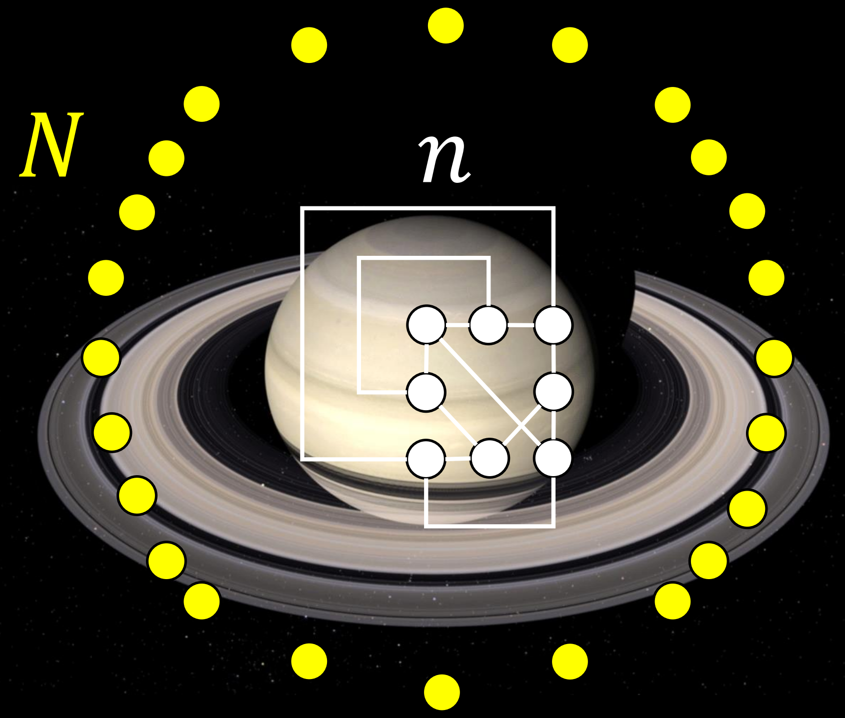
- ∀ Stateless det. Cent. Local alg. D, that solves a problem $\in\ NICE$
  - with probe complexity $t(n)\ =\ o\left(\sqrt{\log n}\right)$,
- ∃ Det. Dist. Local alg. that solves P by simulating D
  - $\#rounds \leq t\left(\Theta\left(n^{\log n}\right)\right)$.

Dist. Local$\left[t\left(\Theta\left(n^{\log n}\right)\right)\right]$

S.less det. Cent. Local$[t(n)]$

Proof Outline

- Goal: solve the problem on input graph G
  - n vertices, bounded degree Δ.
- Dist. Local Simulates Cent. Local on $G \cup H$:
  - Disjoint graphs,
  - H is a virtual graph of $\Theta(n^{\log n})$ vertices.
  - All vertices know H.
- Random "reshuffling" π of IDs
  - Known to all vertices (public randomness).
- ⇒Far probe "lands" w.h.p. in H
  - H is known ⇒ consistent answers to probes.
- Derandomization: There is a "good" π for **all** graphs
  - On n vertices, bounded degree Δ.



Coro: New Lower bounds in the Cent. Local Model

$$\Omega\left(\#Rounds\left(P, 2^{\sqrt{\log n}}\right)\right) = \Omega(Probe\ Complexity(P, n))$$

For $\Delta = O(1)$.

- Example:

$$\Omega\left(\log\log \ell|_{\ell=2^{\sqrt{\log n}}}\right) = \Omega(\log\log n)$$

- Hence:

Problem	Cent. Local #Probes (Det. Stateless) [EMR14]	Dist.Local Lower Bound [Linial 92, Lenzen Wattenhofer 2008]
MIS	$O(\log^* n)$	$\Omega(\log^* n)$
MM	$O(\log^* n)$	$\Omega(\log^* n)$
$(\Delta + 1)$-color	$O(\log^* n)$	$\Omega(\log^* n)$
$(1 - \epsilon)$-MCM	$O(\text{Polylog}^* n)$	$\Omega(\log^* n)$
$(1 - \epsilon)$-MWM	$O\left(\min\left\{\Gamma, \frac{n}{\epsilon}\right\} \cdot \log^* n\right)$	$\Omega(\log^* n)$

Stateless Cent. Local to "Localized" Stateless Cent. Local

Thm.

- ∀ Stateless Cent. Local alg. A, that solves a problem ∈ $NICE$
  - Probe complexity $t(n) = o(n^{1/4}/\Delta)$,
  - Seed length $s(n)$,
  - Error probability $\mathbb{Pr}(A)$.
- ∃ Stateless Cent. Local alg. that solves P by simulating A
  - Probe complexity $t(n^4)$,
  - Probe radius $t(n^4)$,
  - Seed length $s(n^4) + O(t(n^4) \cdot \Delta \cdot \log n)$,
  - Error probability $\mathbb{Pr}(A) + O(1/n)$.

Remarks:

- Polynomial "blow-up".
- Constructive.
- Applies also for randomized algs.
- [Levi, Rubinfeld, Yodpinyanee 2016]
  - Rand. $(1-\epsilon)$-MCM with remote probes,
  - Probe complexity $Poly(\Delta, \log n)$,
  - Underlying assumption that the input graph is connected,
  - ⇒simulation cannot be applied.

S.less Cent. Local$[t(n^4)]$

S.less Cent. Local$[t(n)]$

Conclusion

LAZY RULE #1:

CAN'T REACH IT.
DON'T NEED IT.

Open Questions

- Smaller "blow-up" for Cent. Local to Dist. Local ?
- Constructive simulation for Cent. Local to Dist. Local ?
- Carrying lower bounds that depend also on Δ .

Cent. Local vs. Property Testing

- Property Testing
  - Distinguish: Have a property/ϵ −far from having the property.
  - General scheme (one sided error):
    - Probe the object, $\#probes = f(\epsilon^{-1}), o(|Object|)$
    - If object have the property answer YES,
    - If the object is ϵ −far answer NO w.p $\geq 2/3$
- A tester answers a question about a global property by inspecting the object locally.
- Borrowing lower bounds from Property Testing to Cent. Local.
  - Example: Using LSSG Cent.Local in cycle-freeness testing [Levi Ron Rubinfeld 14]

Cent. Local to Sublinear Approximation
By Example: Vertex Cover (Adapted from [Parnas Ron 2007])

- Given a (det) Cent.Local alg ALG for α-apx VC
  - $\#probes = p$
  - $ALG(v) = 1 \leftrightarrow v \in VC$
- U.a.r select $s = O(\epsilon^{-2})$ vertices from G.
  - Denote the selected subset by S.
- For each $v \in S$,
  - $\chi_v \leftarrow ALG(v)$
- Output: $\widetilde{VC} = \frac{n}{s} \cdot \sum_{v \in s} \chi_v + \frac{\epsilon}{2} n.$

- We get:
  - $\alpha \cdot OPT \leq \widetilde{VC} \leq \alpha \cdot OPT + \epsilon \cdot n$ w.p. $\geq 2/3$
  - $\#probes = \epsilon^{-2} \cdot p$

Carrying Lower bounds from Sublinear Approximation to Cent. Local #1

By Example: Vertex Cover (Adapted from [Parnas Ron 2007])

- Given a (det) Cent.Local alg ALG for α-apx VC
  - $\#probes = p$
- We get:
  - $\alpha \cdot OPT \leq \widetilde{VC} \leq \alpha \cdot OPT + \epsilon \cdot n$ w.p. $\geq 2/3$
  - $\#probes = \epsilon^{-2} \cdot p$
- Sublin apx lower bound:
  - $\forall\ \alpha > 1, b \leq \frac{n-1}{4\alpha}, \epsilon < \frac{1}{4}$
  - $\forall(\alpha, \epsilon)$-apx VC alg requires $\Omega(b)$ probes
    - $\bar{\Delta} = \Theta(b)$

$\Rightarrow$ $\forall\ \alpha$-apx VC Cent.Local alg requires $\Omega(\bar{\Delta})$ probes

Carrying Lower bounds from Sublinear Approximation to Cent. Local #2

By Example: Vertex Cover (Adapted from [Trevisan] [Parnas Ron 2007])

- Given a (det) Cent.Local alg ALG for α-apx VC
  - $\#probes = p$
- We get:
  - $\alpha \cdot OPT \leq \widetilde{VC} \leq \alpha \cdot OPT + \epsilon \cdot n$ w.p. $\geq 2/3$
  - $\#probes = \epsilon^{-2} \cdot p$
- Sublin apx lower bound:
  - $\forall\ \gamma, \epsilon$ constants
  - $\exists\Delta$ constant
  - $\forall(2-\gamma,\epsilon)$-apx VC alg requires $\Omega(\sqrt{n})$ probes
    - For graphs of degree Δ

$\Rightarrow \forall 2-\gamma$-apx VC Cent.Local alg requires $\Omega(\sqrt{n})$ probes

Outline

- Model ✓
- Connections ✓
- Techniques
- State-of-the-art Algs
- Local Graph Generators

Techniques

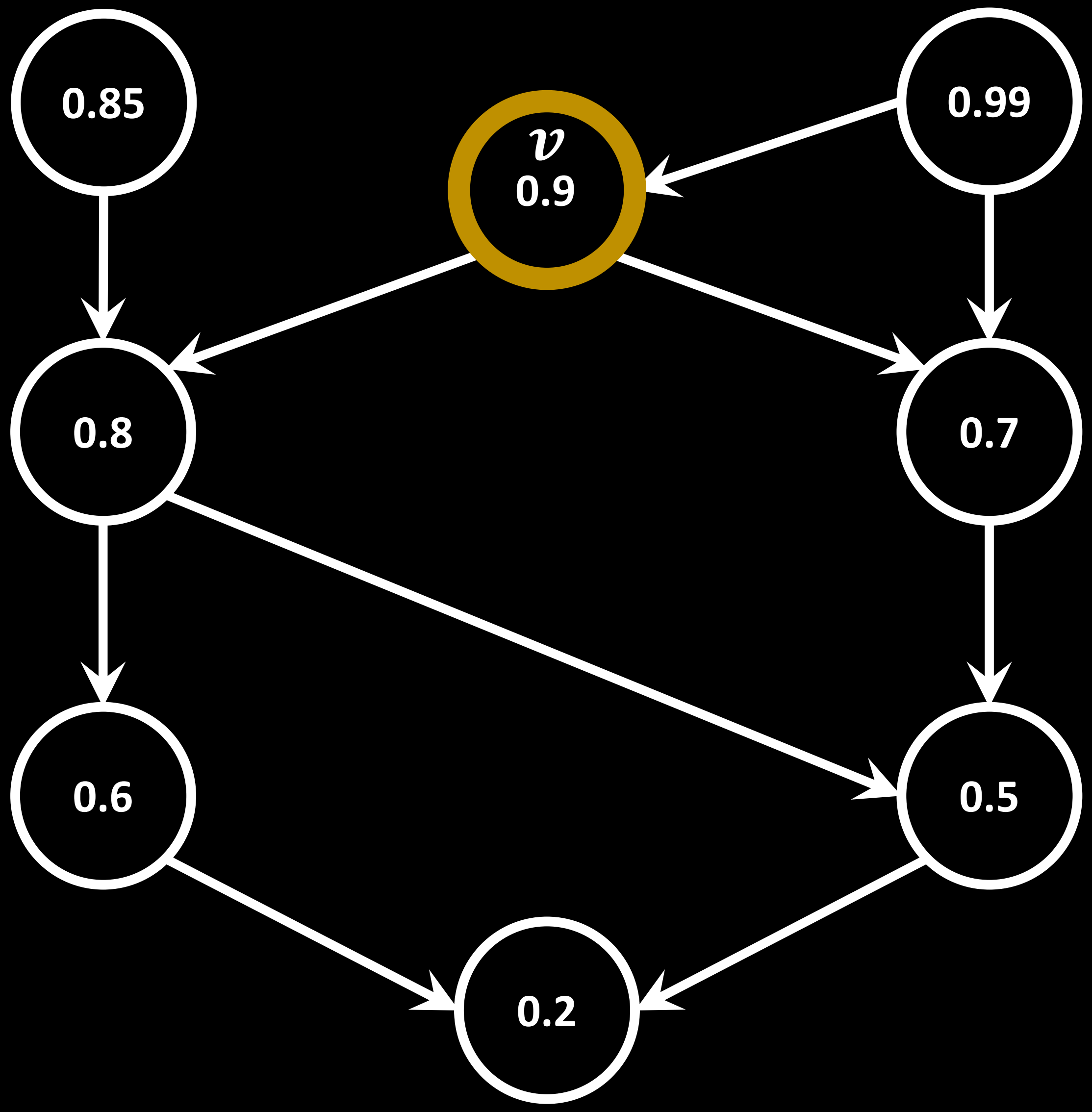
Localization of Local-Sequential Algs [Mansour, [Rubinstein, Vardi, Xie 2012], [Even,M,Ron 2014]

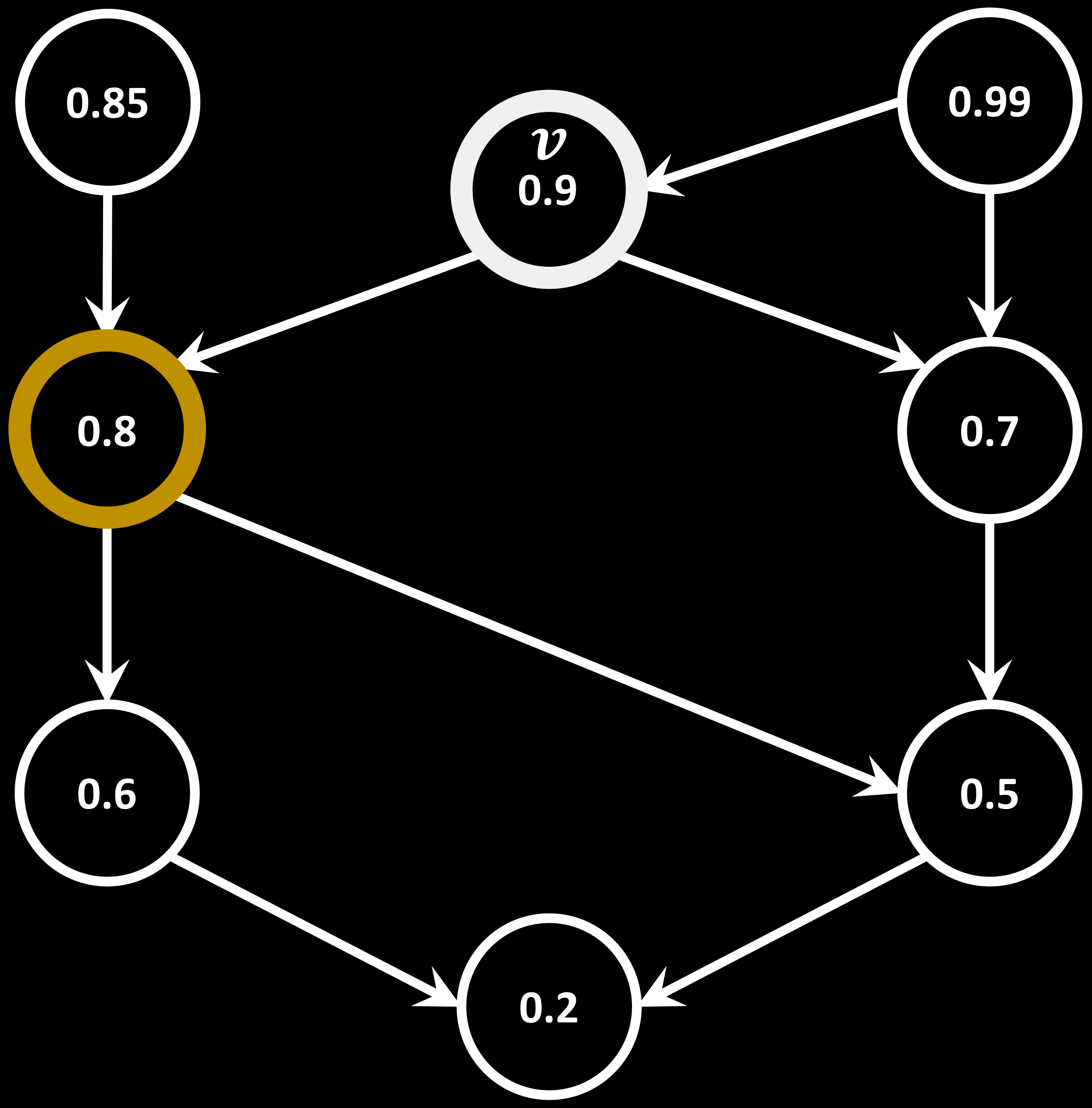
- Greedy Sequential MIS Algorithm:
  - $MIS \leftarrow \emptyset$
  - Fix vertex ordering $v_1, \dots, v_n$
  - For $i = 1$ to n:
    - Add v_i to MIS if $MIS \cap \Gamma(v_i) = \emptyset$
- Similar Greedy algs:
  - $\Delta + 1$ greedy vertex coloring
  - Maximal Matching
- **Question:** Can we simulate Greedy algs by a Cent. Local alg?

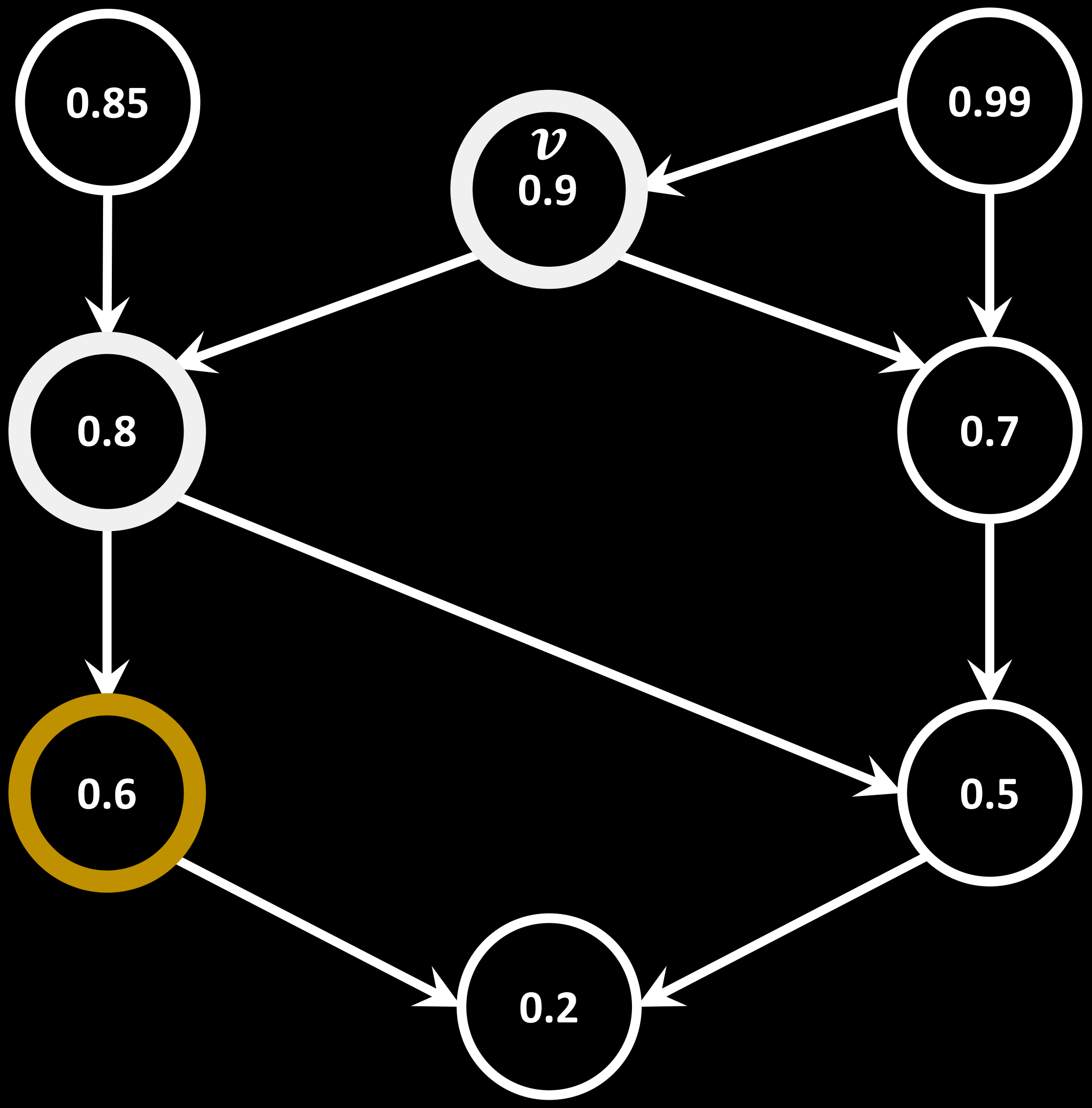
Localization of Local-Sequential Algs, cont.

- Assume $OBR(p, r)$ Cent. Local alg
  - Query: $\{u, v\} \in E$
  - Output: $u \rightarrow v \; or \; v \rightarrow u$
  - Probe complexity p
  - Objective: compute an acyclic orientation with maximum rechability r.
- Simulate the Greedy Sequential MIS Algorithm – how?
  - DFS-MIS!

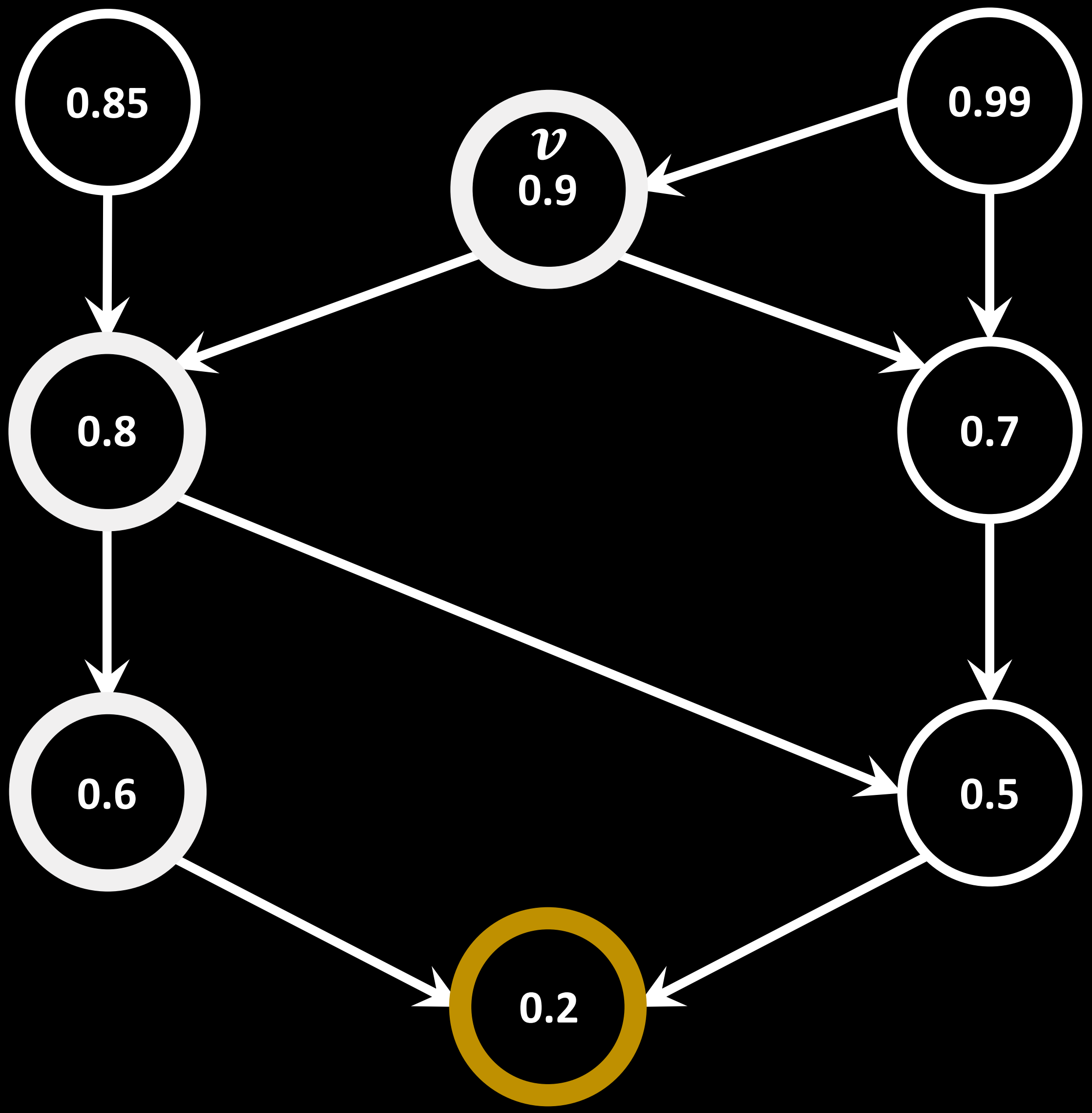
MIS with directed DFS. Query: is v in the MIS?

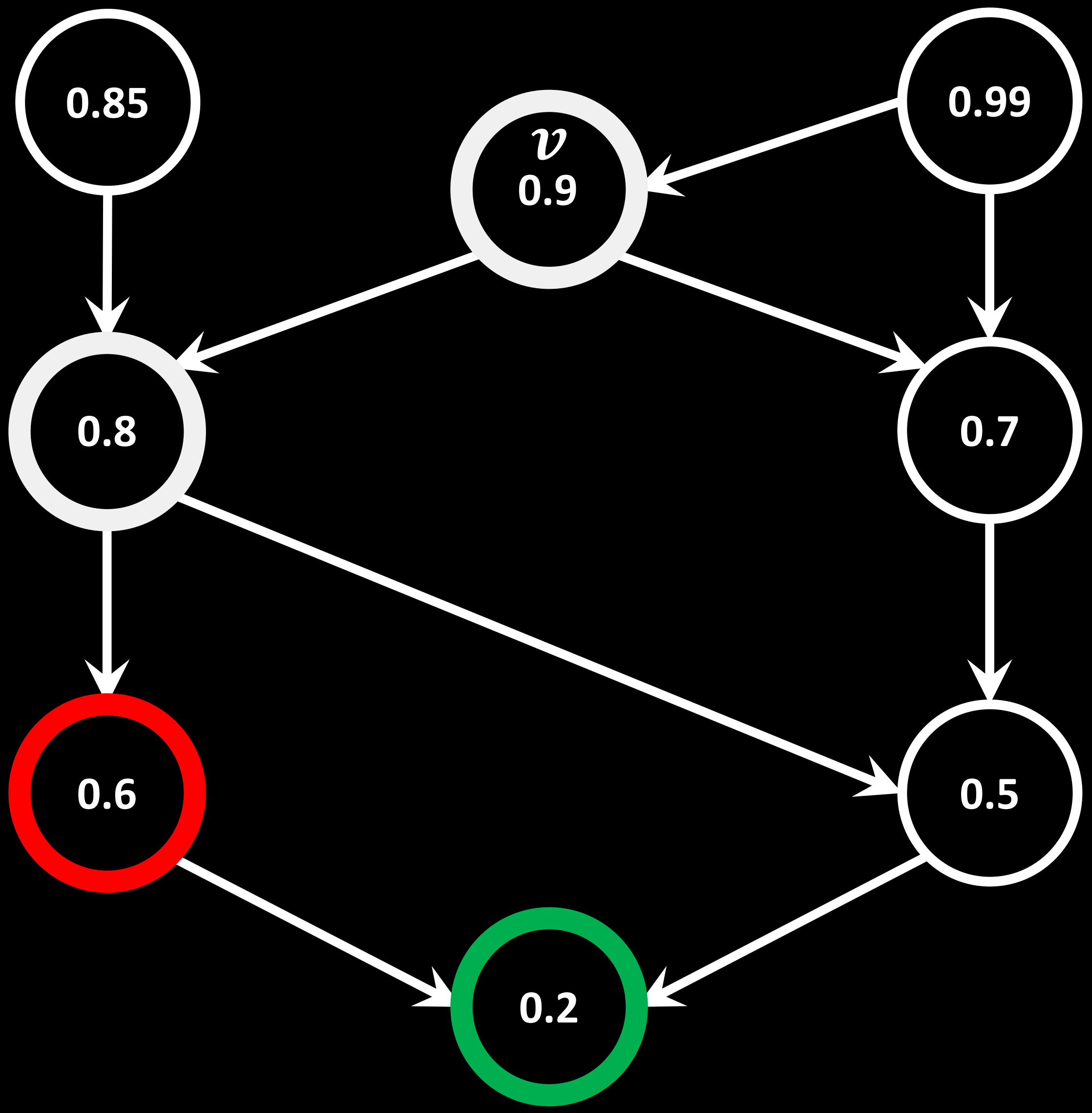


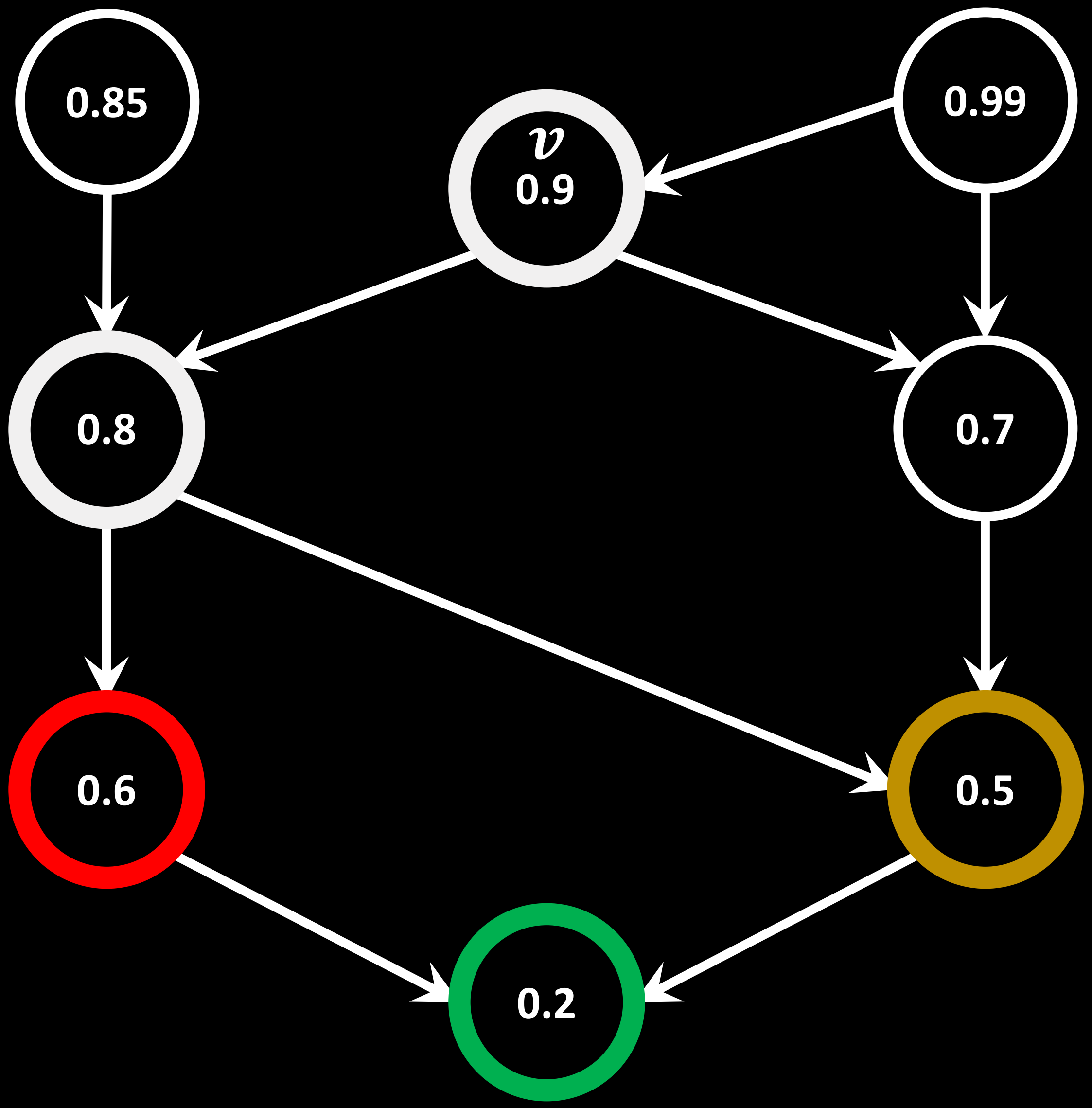

0.85
v
0.9
0.99
0.8
0.7
0.6
0.5
0.2

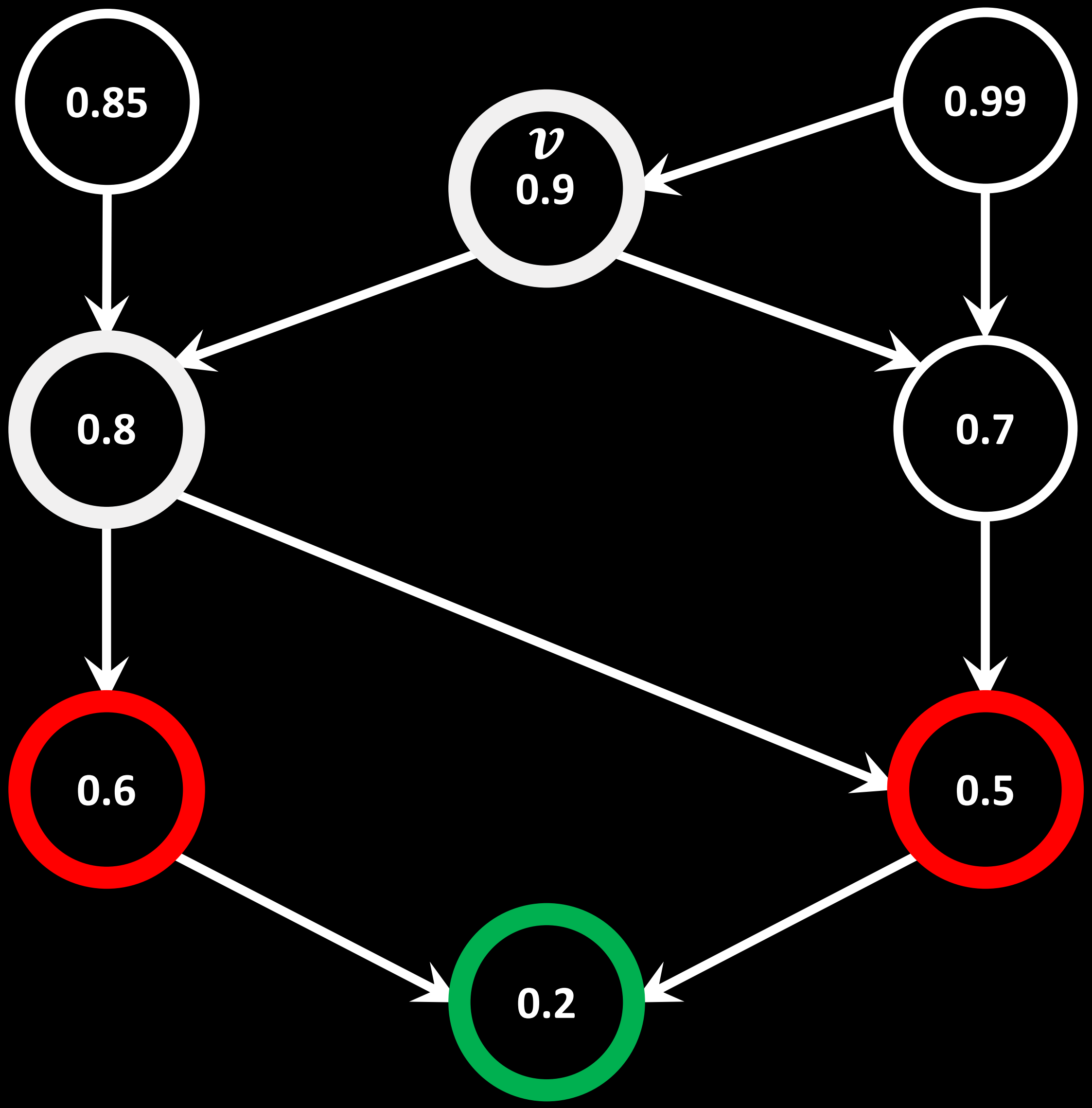


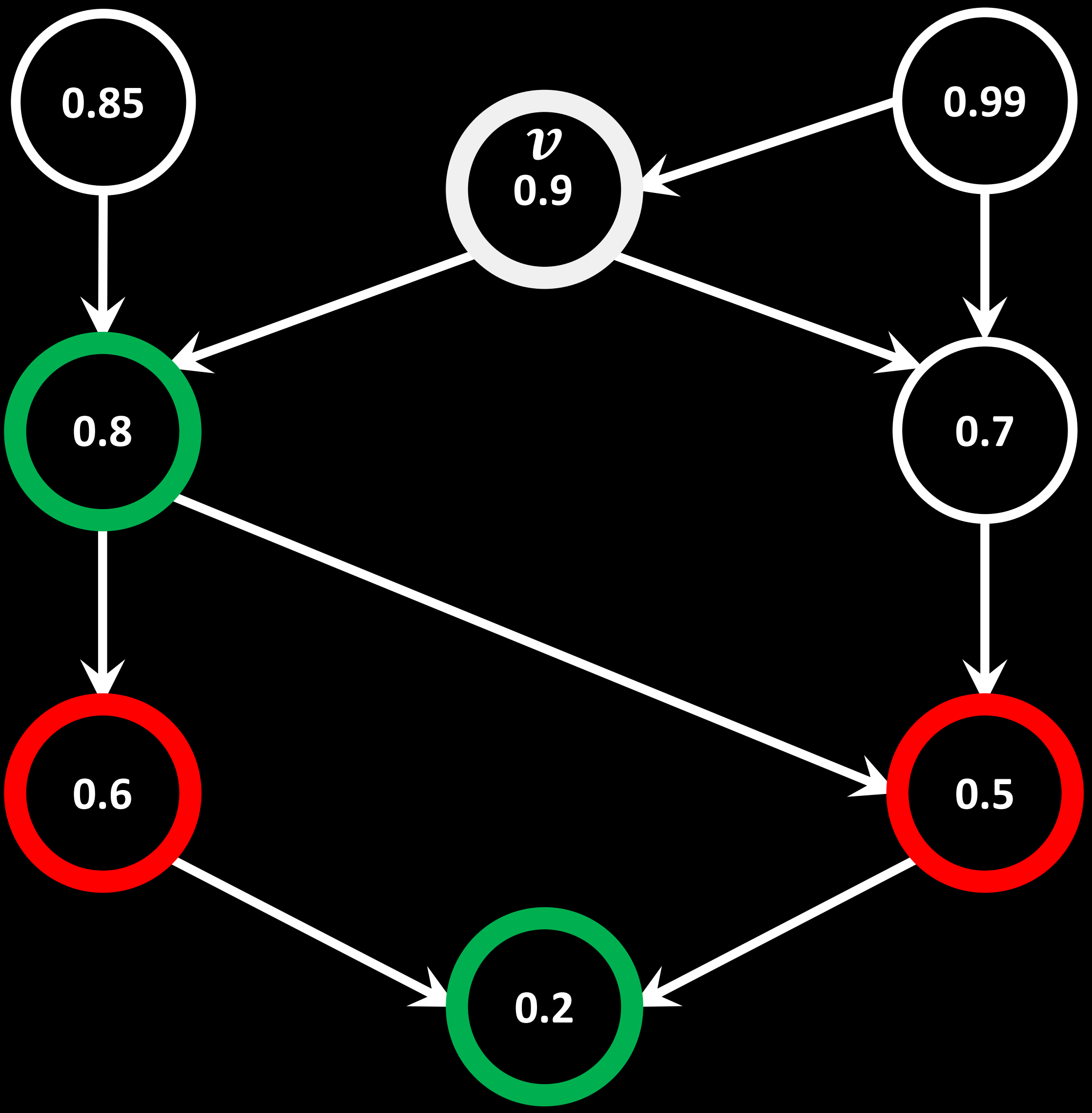
0.85
v
0.9
0.99
0.8
0.7
0.6
0.5
0.2

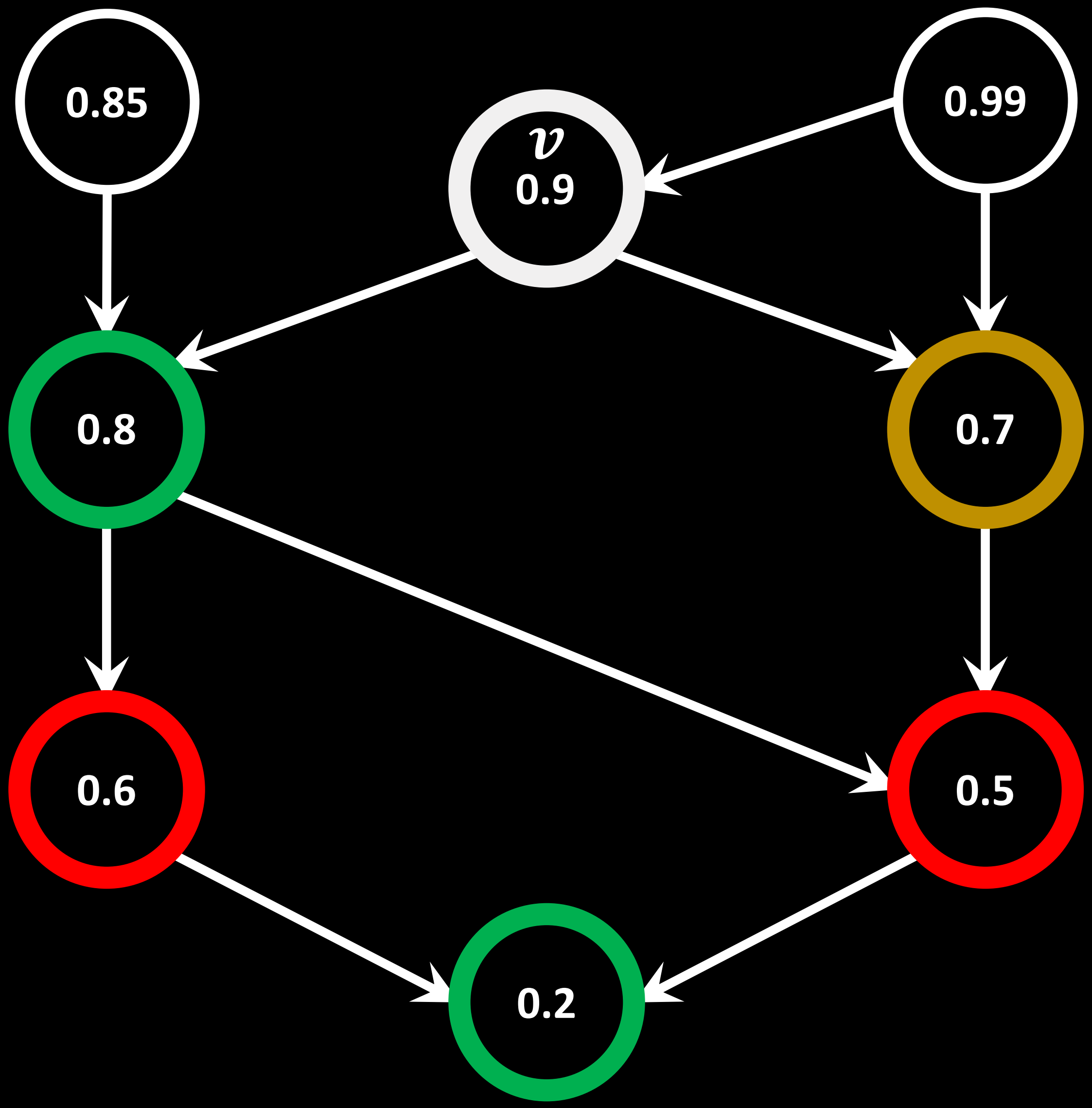

0.85
v
0.9
0.99
0.8
0.7
0.6
0.5
0.2

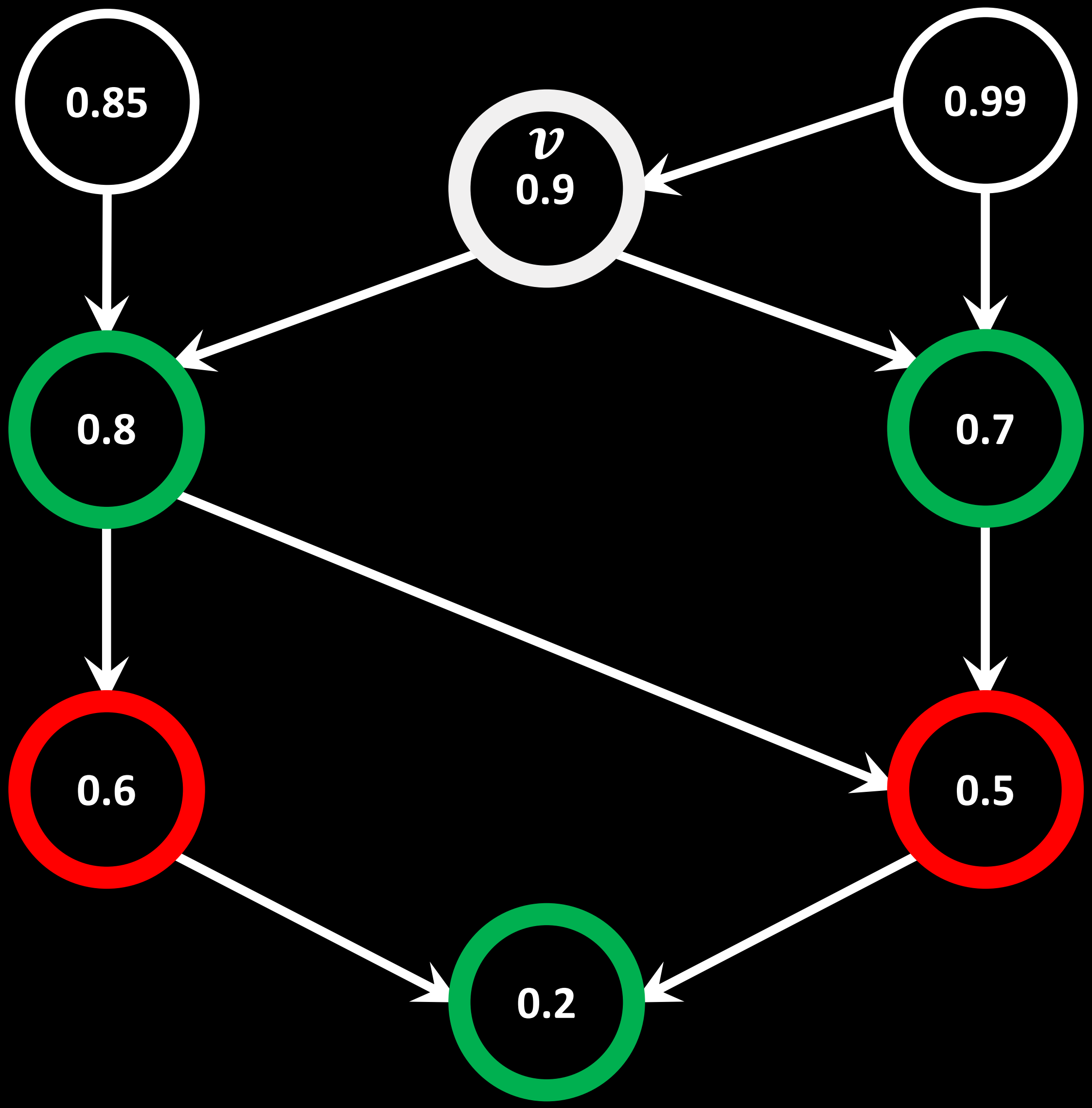

0.85
v
0.9
0.99
0.8
0.7
0.6
0.5
0.2


0.85
v
0.9
0.99
0.8
0.7
0.6
0.5
0.2

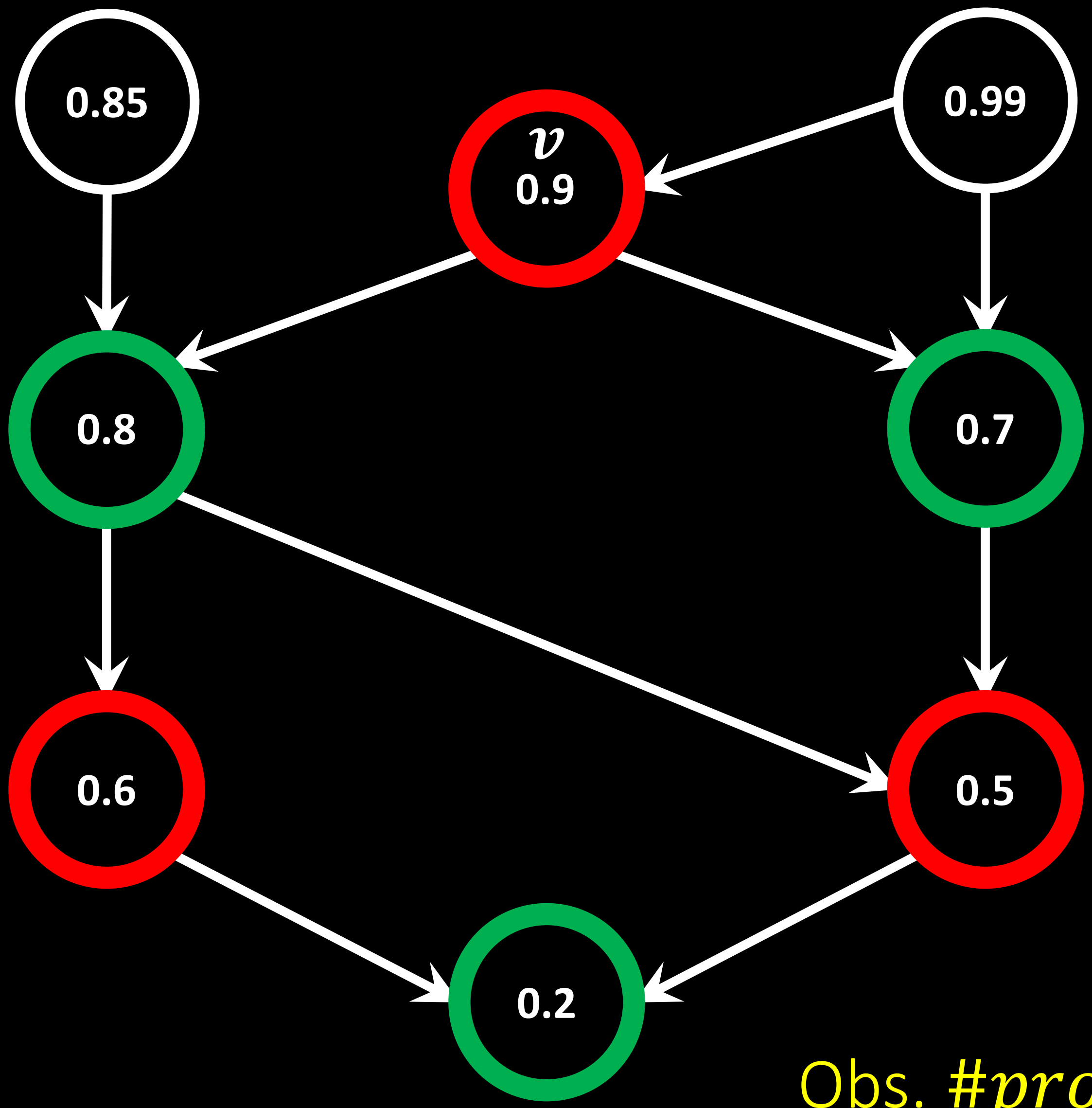

0.85
v
0.9
0.99
0.8
0.7
0.6
0.5
0.2


0.85
v
0.9
0.99
0.8
0.7
0.6
0.5
0.2


0.85
v
0.9
0.99
0.8
0.7
0.6
0.5
0.2

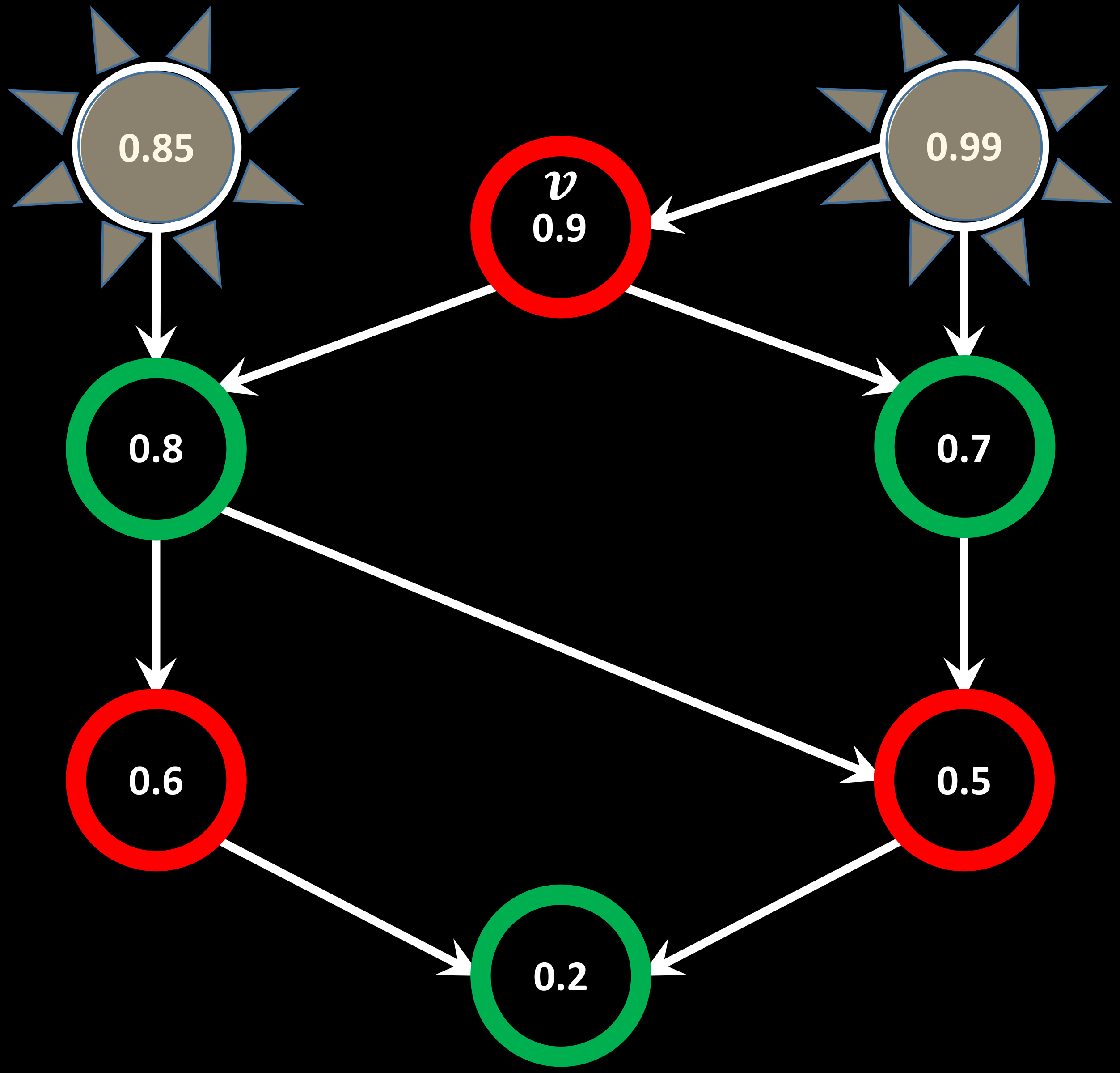

0.85
v
0.9
0.99
0.8
0.7
0.6
0.5
0.2

MIS with directed DFS. Query: is v in the MIS? No!



Obs. $\#probes =$ reachability set of v.

- $OBR(p,r) \Rightarrow \#probes = r$


0.85
0.99
v
0.9
0.8
0.7
0.6
0.5
0.2

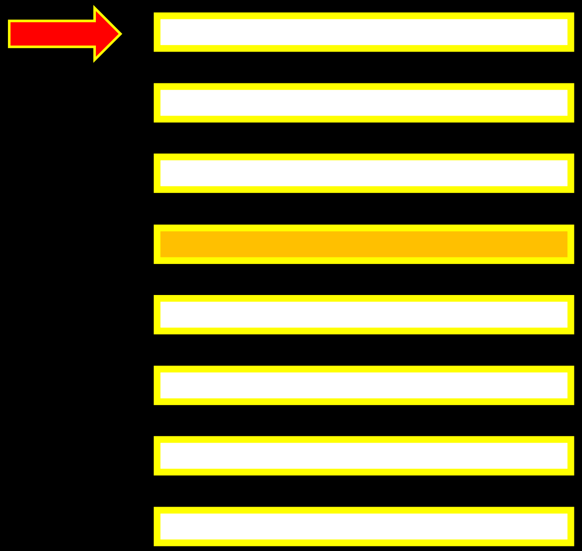
Amplification via Far Probes

- We saw that "far" probes are not useful for Nice problems.
- In sublinear apx, far probes are used for estimation.
  - Apx size of Maximum Matching, Minimum Vertex Cover, etc.

- Can also be used for **Amplification [Levi, Rubinfeld, Yodpinyanee 2016]**
  - **Given rand Cent.Local**
  - **Success prob.** $\geq 2/3$
  - $\Rightarrow$**Success prob.** $\geq 1 - 1/\mathrm{Poly}(n)$

Amplification via Far Probes, cont.

- Can also be used for Amplification [Levi, Rubinfeld, Yodpinyanee 2016]
  - Given rand Cent.Local
  - Success prob. $\geq 2/3$
  - $\Rightarrow$Success prob. $\geq 1 - 1/\mathrm{Poly}(n)$
- Idea:
  1. Pick u.a.r. a random seed,
  2. Estimation of solution by a random sample (problem dep.),
     - Far probes.
  3. If estimation is “bad” then repeat.
  4. Fix the “good” seed.
- Total of $\approx \#probes(alg) \cdot \log n$



- Can be used every time before answering a query, or
- As a preprocessing stage.
- Example: $(1-\epsilon) - MCM$.

Local Improvement [Nguyen, Onak 2008]

By Example: Maximum Cardinality Matching.

- Following [Hopcroft, Karp 73]:
- $(1-\epsilon)$-apx "Global" alg:
  - For $i = 0$ to $1/\epsilon$ do
    - $P_{i+1} \leftarrow M_i$ −Aug. paths of length $2i + 1$,
    - $P^*_{i+1} \leftarrow MIS$(intersection graph over P_{i+1}),
    - $M_{i+1} \leftarrow Mi \oplus E(P^*_{i+1})$.
- Challenge [LPSP-08, NO-08, MV-13]
  - Simulate by a dist. alg/ CENTLOCAL?

Local Improvement [NO 08]

By Example: Maximum Cardinality Matching.

$(1-\epsilon)$-apx "Global" alg:

For $i = 0$ to $1/\epsilon$ do

$P_{i+1} \leftarrow M_i$ −Aug. paths of length $2i + 1$,

$P^*_{i+1} \leftarrow MIS$(intersection graph over P_{i+1}),

$M_{i+1} \leftarrow Mi \oplus E(P^*_{i+1})$.

- Technique introduced for sublin-apx-algs
- Global alg with k phases to k Cent.Loal algs for each phase.
  - ith oracle gives access tp ith phase's output.
  - "Inner" queries are generated to "previous" oracles.
  - Each oracle probes also the graph.
- Simulation of $(1-\epsilon)$-apx global alg by Cent.Loal
  - Requires sim probes to P_{i+1}.

Oracle($e \in M_i$)

Return: $\begin{cases} 0, & \text{if } i = 0, \\ \text{Oracle}(e \in M_{i-1}) \oplus (e \in E(P^*_i)), & \text{if } i \geq 1. \end{cases}$

$e \in E(P^*_i)$

- Probe $Ball_{2i+1}(e)$.
- $\forall$ edge $e' \in Ball_{2i+1}(e)$: Oracle($e \in M_{i-1}$).
- List $P_i(e) \triangleq$ paths in P_i that contain e.
- **Return:** $P_i(e) \cap$ MIS(intersection graph over P_i) $\neq \emptyset$

Outline

- Model ✓
- Connections ✓
- Techniques ✓
- State-of-the-art Algs
- Local Graph Generators

State-of-the-Art Algs

Graph Coloring Algs

- Graph $G = (V, E)$
- c-coloring of G
  - $c: V \to [c], c \in \mathbb{N}$
  - $\forall \{u, v\} \in E : c(u) \neq c(v)$

Δ^2-coloring

- [Linial 1992]: Dist.Local$[O(\log^* n)]$
- Simple Dist.Local to Cent.Local: Cent.Local$\left[\Delta^{O(\log^* n)}\right]$
- [Even, M, Ron 2014]: Cent.Local$[\Delta^4 \cdot \log^* n]$
  - Partition G into edge-disjoint subgraphs od degree 2 [Barenboim, Elkin, Kuhn 2014]
  - ⇒ Simple Dist.Local to Cent.Local on each subgraph: Cent.Local$[O(\log^* n)]$
  - ⇒ 4^{Δ}-coloring
  - Apply color reduction tech. by [Linial 1992]: Dist.Local$[O(1)]$ ⇒Cent.Local$[\mathrm{Poly}(\Delta)]$.

$(\Delta+1)$-coloring

- [Even,M,Ron 2014]: Localization of Greedy coloring: Cent.Local$\left[\Delta^{O(\Delta^2)} \cdot \log^* n\right]$
- [Fraigniaud, Heinrich, Kosoeski 2016]: Cent.Local$\left[\Delta^{O(\sqrt{\Delta}\cdot\log^{2.5}\Delta)} \cdot \log^* n\right]$
  - Given Δ^2-coloring
  - From Δ^2 to $(\Delta+1)$-coloring: Dist.Local$\left[O\left(\sqrt{\Delta} \cdot \log^{2.5}\Delta\right)\right]$
  - We already know how to color efficiently in Δ^2 colors.
  - We get the new bound by applying Dist.Local to Cent.Local.

Coloring: Open questions

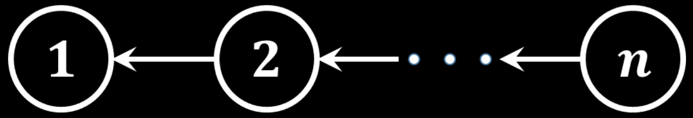
- Lower bounds in term of Δ.
- Randomized Algs?

Acyclic Orientation with Bounded Reachability (OBR)
[Even, M, Ron 2014]

- Instance: A graph $G = (V, E)$
- Solution: Directed acyclic graph $H = (V, A)$
  - Underlying graph is G
- Objective: Minimize max reachability
  - $\max_v |\{u \mid v \rightsquigarrow u\}|$

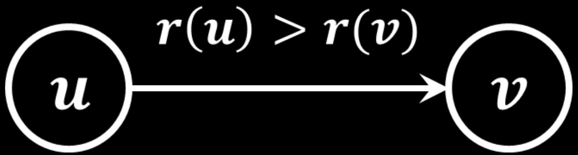
- Cent.Local version:
  - "is the edge from u to v is outgoing?

- Trivial bad solution: From high to low ID



OBR: Randomized Algs

- Randomized ranking [Nguyen, Onak 2008]:
  - $r(v) \leftarrow U[0,1]$



- $\mathbb{E}[Reach] = e^{\Delta}/\Delta$

- $Reach = 2^{O(\Delta)} \log n$ w.h.p. [Reingold, Vardi 2016]
  - $O(\log n)$ seed length.

OBR: Deterministic Algs

- [Even,M,Ron 2014]: Observation: use vertex c-coloring
  - $u \to v$ if $c(u) > c(v)$
  - $Reach \leq O(\Delta^c)$
- Apply coloring algs:
  - Δ^2-coloring: Cent.Local$[\Delta^4 \cdot \log^* n]$
  - $(\Delta + 1)$-coloring: Cent.Local$\left[\Delta^{O\left(\sqrt{\Delta} \cdot \log^{2.5} \Delta\right)} \cdot \log^* n\right]$
    - Looks “to expensive” at first – actually beneficial.

OBR: Open questions

- Optimization version: minimize maximum reachability.
  - Apx alg
- Lower bounds in terms of Δ.

Maximal Independent Set (MIS) (and Maximal Matching)

- Instance: Graph $G = (V, E)$
- $MIS \subseteq V$
  - Each pair in MIS is not an edge
  - Set is maximal w.r.t. inclusion

- Cent.Local version:
  - “is vertex v is in the MIS?”

MIS: Deterministic Algs

- [Even,M,Ron 2014]: Cent.Local$\left[\Delta^{O(\Delta)} \cdot \log^* n\right]$
  - Use Cent.Local$\left[\Delta^{O\left(\sqrt{\Delta} \cdot \log^{2.5} \Delta\right)} \cdot \log^* n\right]$
  - Obtain reach of $\Delta^{O(\Delta)}$
  - Follows by Localization of Greedy coloring.

MIS: Randomized Algs

- [Mohsen 2016]: Cent.Local$\left[2^{O(\log^2 \Delta)} \cdot \log^2 n\right]$ w.h.p
  - Space: $2^{O(\log^2 \Delta)} \cdot \log^3 n$
  - Shattering [Alon, Rubinfeld, Vardi, Xie 2012], [Beck 91]
  - “Brute force” on each piece.

Efficient Exploration of the Reachability Set

By example: MIS

- Also called “pruning”.
- Applied [Yoshida Yamamoto, Ito 2012], [Onak, Ron, Rosen, Rubinfeld 2012] in the context of Sublin-apx.

- A “twist” by [Yoshida Yamamoto, Ito 2012]:
  - Scan the reachability set from lower rank to higher rank
  - $\mathbb{E}[Reach] = e^{\Delta}/\Delta$ goes down to $\mathbb{E}[Reach] \approx O(\Delta^2)$

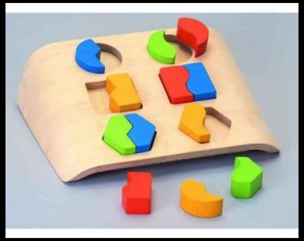
Pruning: “Open” questions

- Simpler proof of [Yoshida Yamamoto, Ito 2012].

Approximate Maximum (Weighted) Matching (MCM,MWM)

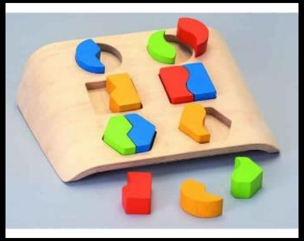
- Instance: Graph $G = (V, E)$
- $MCM \subseteq E$
  - MCM is a matching, subgraph of deg 1.
  - Maximum possible number of edge.
- α-apx version:
  - $ALG \geq \alpha \cdot MCM^*$
- Cent.Local version:
  - "is edge e is in the α-apx MCM?"

MCM,MWM: Deterministic Algs



- [Even,M,Ron 2014] $(1-\epsilon)$-MCM: Cent.Local$\left[(\log^* n)^{O(\frac{1}{\epsilon})} \cdot 2^{O(\Delta^{1/\epsilon})}\right]$
  - "Local improvement" over [Hopcroft and Karp 1973]
  - With Cent.Local MIS
- [Even,M,Ron] $(1-\epsilon)$-MWM: Cent.Local$\left[(\log^* n)^{O(\frac{1}{\epsilon}\log\frac{1}{\epsilon})} \cdot (w_{min}(\epsilon))^{O(\Delta^{1/\epsilon})}\right]$
  - "Local improvement" over [Hougardy, Vinkemeir 2006]
  - With (a variant of) Cent.Local MIS

MCM,MWM: Deterministic Algs, cont.



- [Fischer, Ghaffari 2017] $\Theta(1)$-MWM: Cent.Local$[2^{\log^3 \Delta} \cdot \log^* n]$
  - Transform the graph to Bipartite graph.
  - Compute frac apx MCM + rounding by Dist.Local$[O(\log^2 \Delta)]$
  - MIS in Dist.Local$[O(\log^* \Delta)]$ [Panconesi, Rizzi 2001]
    - Requires Δ^2-coloring

MCM,MWM: Randomzied Algs

- [Levi, Rubinfeld, Yodpinyanee 2016]: Cent.Local$[\mathrm{Poly}(\Delta) \cdot \log^2 n \log\log n]$
  - For Constant ϵ (exp dep.)
  - Seed length: $\mathrm{Poly}(\Delta) \cdot \log^3 n \log\log n$
  - Amplification via Far Probes over [Yoshida Yamamoto, Ito 2012]
    - Roughly: $\text{expected \#probes} = \mathrm{Poly}(\Delta)$.

MCM,MWM: Open questions

- Best of all worlds? $Poly(\Delta)$ and $\log^* n$? Study the t.off?
- Det MCM,MWM:
  - Gap in terms of n

Outline

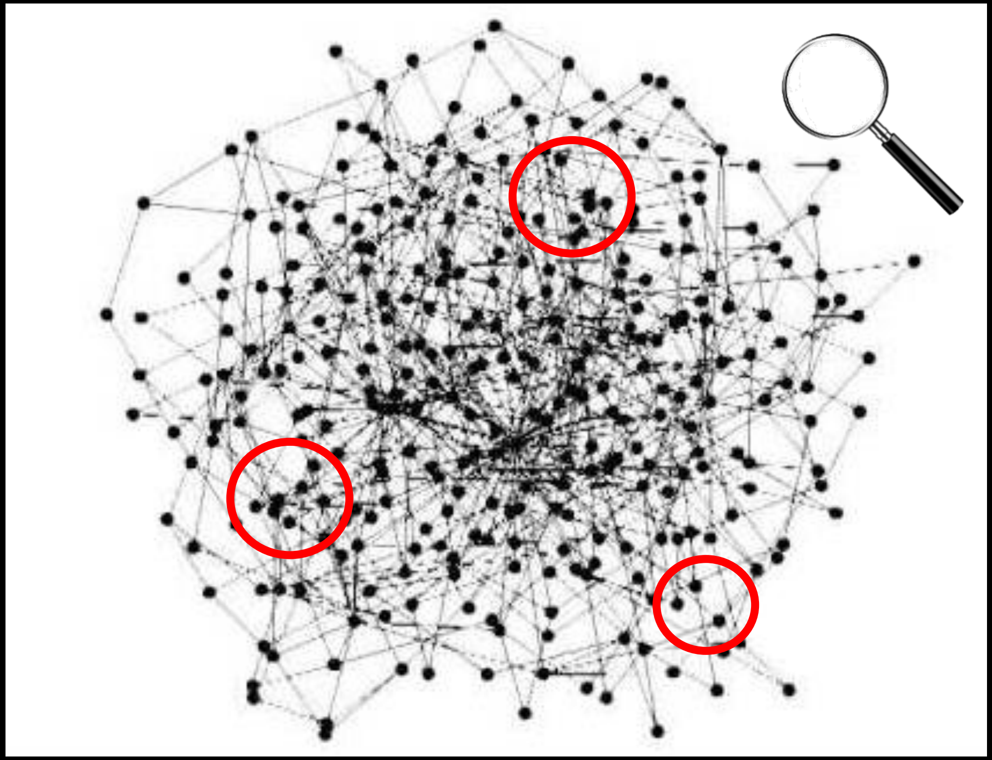
- Model
- Connections
- Techniques
- State-of-the-art Algs
- Local Graph Generators

Local Generation

Based on the paper “Sublinear Random Access Generators for Preferential Attachment Graphs”, by Guy Even, Reut Levi, Moti Medina, and Adi Rosén, (ICALP 2017).

Motivation

Can we give access to a very large random graph without generating the whole graph?



Random Access Generator

- $\mathfrak{D}_n$ - distribution over graphs with n vertices labelled by $1, \dots, n$.
- Random Access Generator for $\mathfrak{D}_n$ gives random access to a graph $G \sim \mathfrak{D}_n$
- Interface:

  Next-neighbor(i)
  - Returns next neighbor of vertex i (or $\perp$ if all neighbors already returned).
  - Neighbors are sorted by their labels.

- The time complexity of the generator per query should be small.

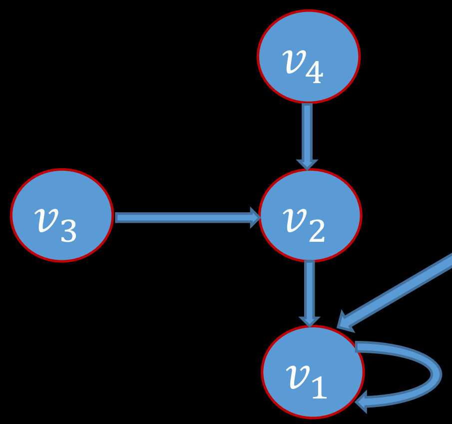
Barabási–Albert Preferential Attachment Model (BA-graphs)

Out-degree $m = 1$.

Vertex v_n points to vertex v_i with probability $\frac{\deg(i, G_{n-1})}{2(n-1)}$.

power-law degree distribution

rich get richer

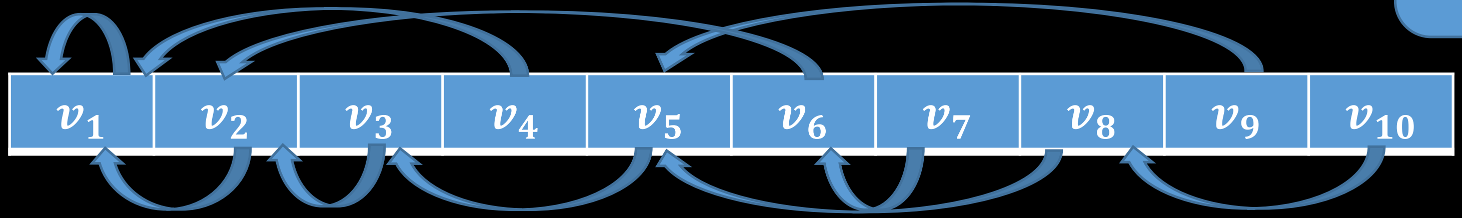


Recursive Tree Model

Out-degree $m = 1$.

Vertex v_n points to vertex v_i with probability $\frac{1}{n-1}$.

uniform parent

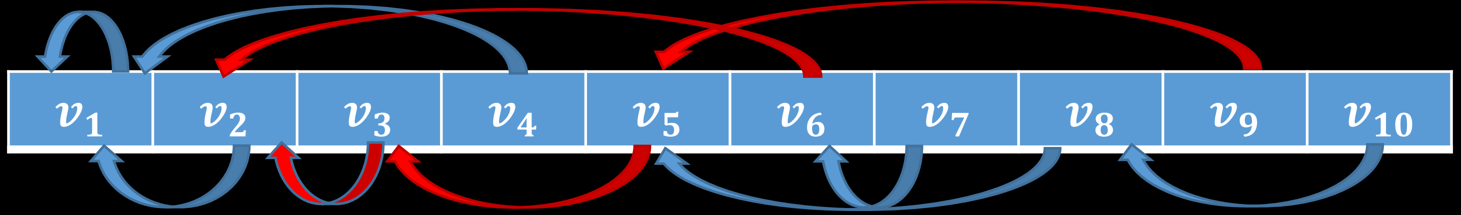


Reduction from BA-graphs to Recursive Tree

In BA-graphs: v_n points to v_i with probability:

$$\frac{\deg(i, G_{n-1})}{2(n-1)} = \frac{1}{2} \cdot \left(\frac{1}{n-1} + \frac{\deg_{in}(i, G_{n-1})}{n-1} \right)$$

Pointers are direct (blue) or in-direct (red) with probability $1/2$.



u → v means: parent(u) = v

u → v means: parent(u) = parent(v)

Model

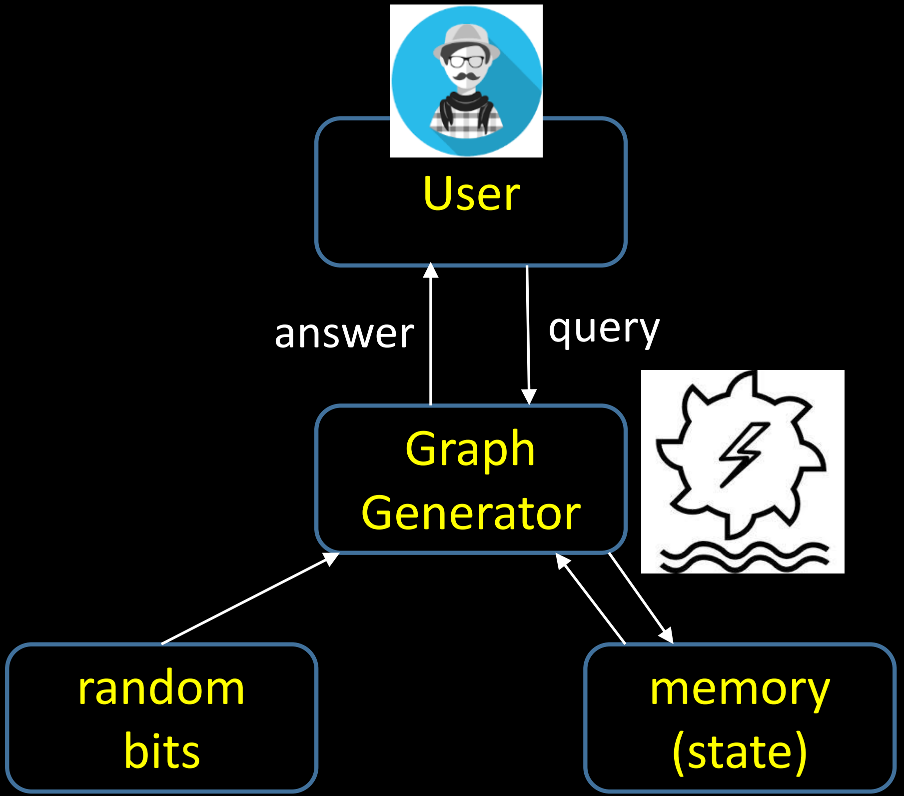
The performance is measured by:

1) running time (per query)
2) memory
3) random bits

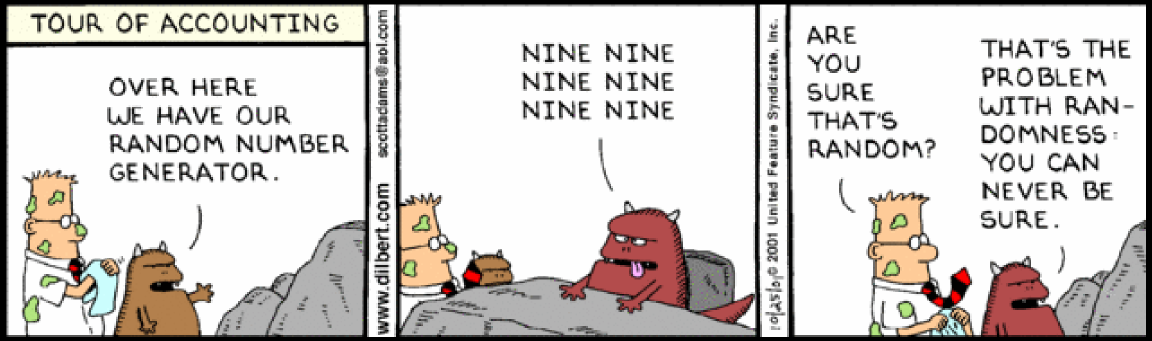
as a function of:

n - size of the graph

q - the number of queries



Result

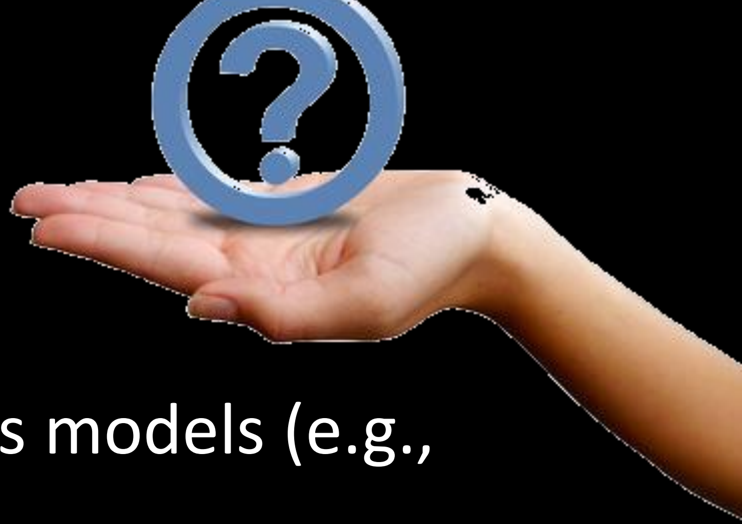


Random Access Generator for BA-graphs:

1) running time (per query): Poly$(\log n)$
2) memory: $q \cdot$ Poly$(\log n)$
3) random bits: $q \cdot$ Poly$(\log n)$

n = size of the graph
q = total #queries

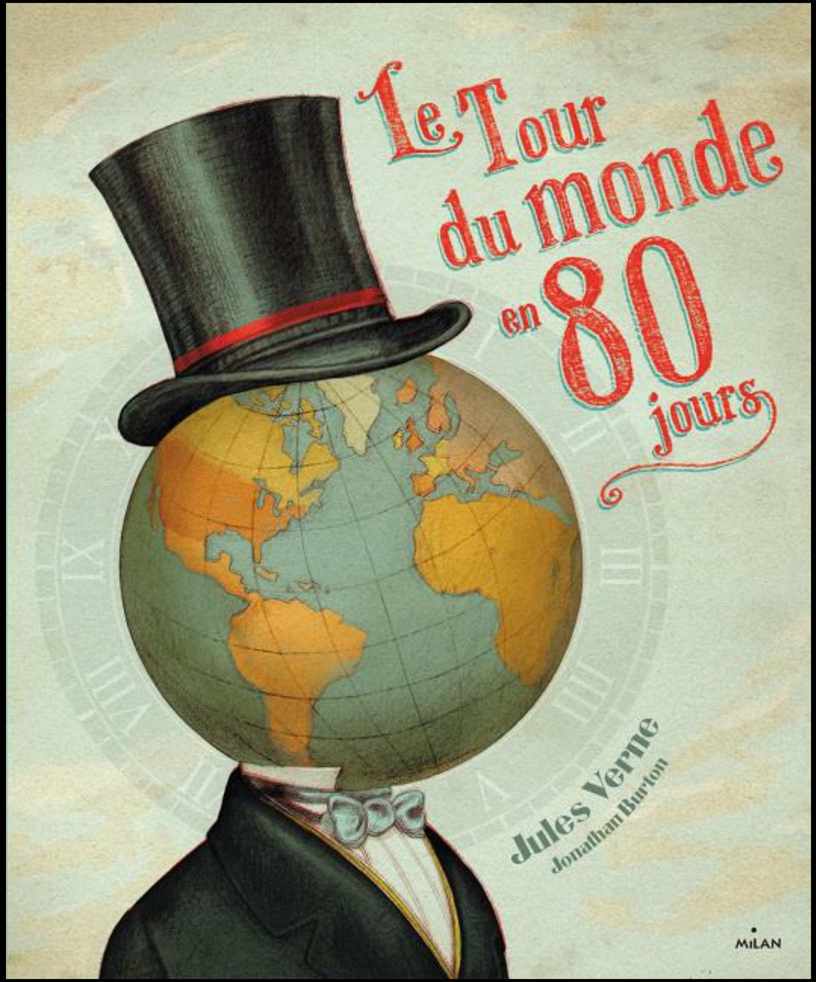
Open Questions



- Random Access Generators for other evolving graphs models (e.g., Forest-Fire Model, Random Surfer Webpage Model).
- Random Access Generators for other Markov Chains.

I didn’t cover

- Cent. Local Mechanism Design [Hassidim, Mansour, Vardi 2009].
- Apx Maximum Weight Spanning tree [Mansour, Patt-Shamir, Vardi 2015].
- Local Sparse Spanning Graphs [Levi, Ron, Rubinfeld 2014] [Levi, Ron, Rubinfeld 2016], [Levi, Moshkovitz, Ron, Rubinfeld, Shapira 2016], [Lenzen, Levi]
- Random access support for Lempel-Ziv compression [Dutta, Levi, Ron, Rubinfeld 2013]
- Partition Oracle [Hassidim, Kelner, Nguyen, Onak 2009], [Levi, Ron 2013]
- Shattering technique [Beck 1991] [Rubinfeld Tamir Vardi Xie 2011] [Alon Rubinfeld Vardi Xie 2012] [Barenboim Elkin Pettie Schneider 2012][Ghaffari 2016] [Levi, Rubinfeld, Yodpinyanee 2016]
- 2-coloring of Bipartite Graphs [Czumaj, Mansour, Vardi 2017]
- Apx Vertex Cover [Feige, Mansour, Schapire 2015]
- MWM to MCM [Mansour, Patt-Shamir, Vardi], Set cover [Indyk, Mahabadi, Rubinfeld, Vakilian, Yodpinyanee 2018]…



Thank you!

- moti.medina@gmail.com
- Survey is available in: https://sites.google.com/site/motimedina/publications/LocalGuide
- These slides are based on talk slides by Reut Levi and myself.