A Time Hierarchy Theorem for the LOCAL Model

Yi-Jun Chang
University of Michigan

Seth Pettie
University of Michigan

S. Brandt, O. Fischer, J. Hirvonen, B. Keller, T. Lempaiäinen, J. Rybicki, J. Suomela, J. Uitto, STOC 2016
Y.-J. Chang, T. Kopelowitz, S. Pettie, FOCS 2016
Y.-J. Chang, S. Pettie, FOCS 2017
Y.-J. Chang, Q. He, W. Li, S. Pettie, J. Uitto, SODA 2018
A. Balliu, J. Hirvonen, J. Korhonen, T. Lempaiäinen, D. Olivetti, J. Suomela, STOC 2018
M. Ghaffari, D. Harris, F. Kuhn, *arxiv 2017.*

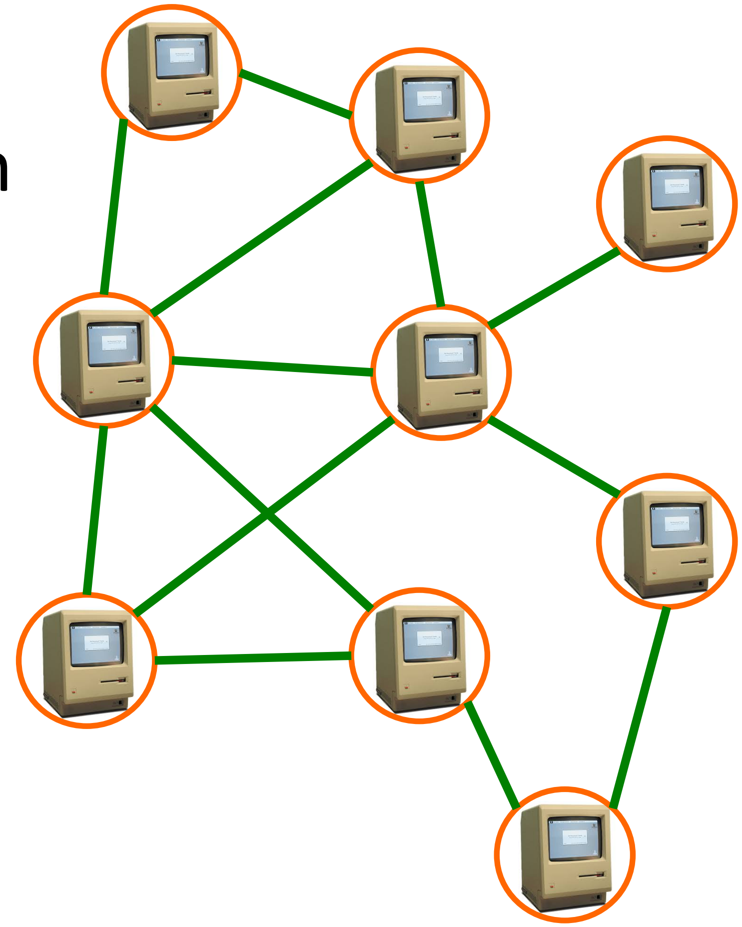
TM Complexity Theory (of the 60s and 70s)

- Does having more time let you solve more problems?
  - For any* (*time constructible) function T, there is a problem that can be solved in O(T(n)) time but not o(T(n)) time. [Hartmanis-Stearns'65], [Fürer'84]
- Are there "universal" (complete) problems for natural complexity classes?
  - Yes, e.g., thousands of NP-complete problems. [Karp'72]
- Is randomness useful (e.g., P=BPP?)
  - Up to polynomial slowdown, probably not.

The LOCAL Model

[Linial'92]

- A graph G=(V,E)
  - Vertex = processor
  - Edge = bidirected communication
  - Time: ***synchronized*** rounds. In each round, each vertex sends a message to each neighbor.
  - ***Computation is free.***
  - ***Message size is unbounded.***
  - "Time" = number of rounds
- ***Randomized*** LOCAL
  - Can generate an ***unbounded number of random bits***



The LOCAL Model

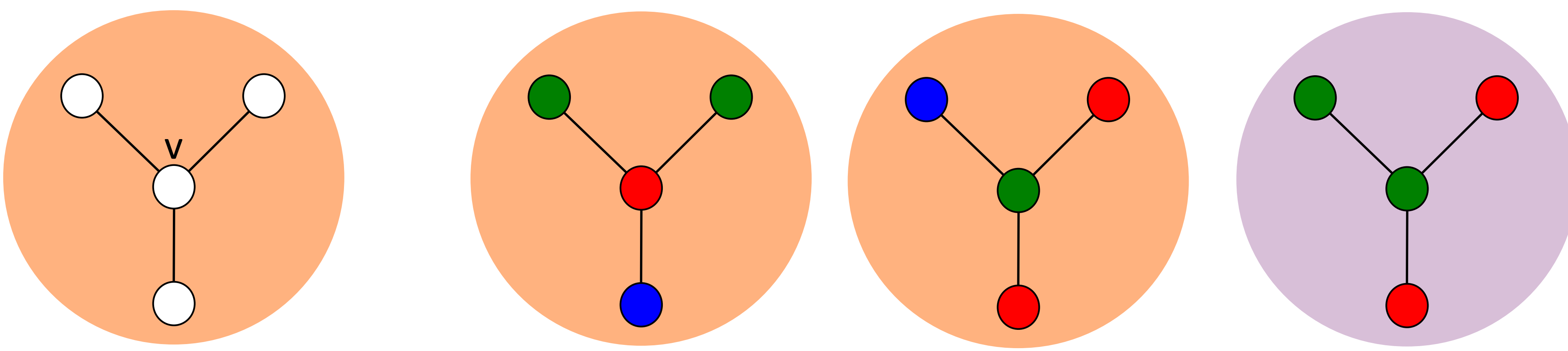
[Linial'92]

- What a vertex ***v*** knows:
  - Global graph parameters: ***n*** = |V|, Δ = $\max_u$ deg(u)
  - A unique O(log n)-bit ***ID***(***v***).
  - A ***port-numbering*** of its deg(***v***) incident edges.

What is a "natural problem"

[Naor-Stockmeyer'95]

- Locally Checkable Labeling (LCL) problem: NTIME(O(1))
  - Input and Output alphabets Σ_{in}, Σ_{out}, integer radius **r**.
    $|\Sigma_{in}|$, $|\Sigma_{out}|$ may depend on Δ, but are independent of **n**.
  - Set **C** of acceptable radius-**r** centered subgraphs.
- Problem: given $V \rightarrow \Sigma_{in}$, compute $V \rightarrow \Sigma_{out}$ such that every vertex's radius-**r** view is in **C**.



radius-1 view from v

some acceptable configurations for 3-coloring.

an unacceptable configuration

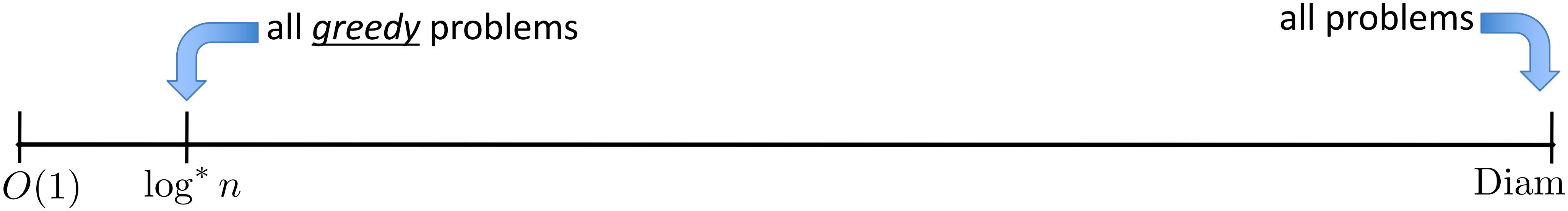
Greedy vs. Nongreedy LCL Problems

- The canonical ***greedy*** problems:
  - Maximal independent set
  - Maximal matching
  - (Δ+1)-vertex coloring
  - (2Δ–1)-edge coloring

  } every partial solution extends to a total solution
- Some ***non-greedy*** problems
  - 0.99-approximate maximum matching
  - Sinkless orientation
  - Δ-vertex coloring
  - (2Δ-2)-edge coloring
  - Frugal coloring, Defective coloring, etc.

Time Hierarchies: Δ=O(1)

1. O(Δ²)-color the graph in log*n time. [Linial's algorithm '92]
2. Apply greedy algorithm to each color class, one at a time.

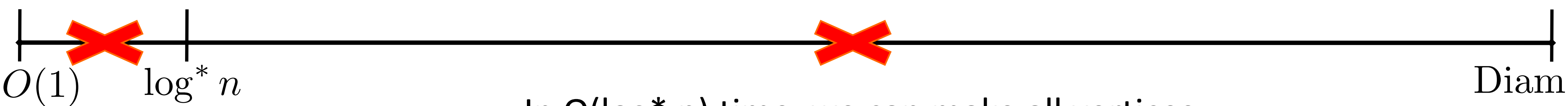


Time Hierarchies: Δ=O(1)

n-path/cycle, $(\sqrt{n} \times \sqrt{n})$-grid/torus

Naor, Stockmeyer'95

via hypergraph Ramsey argument…
any O(1) time algorithm can be made
order-invariant w.r.t. vertex IDs.



In O(log* n) time, we can make all vertices
think they are in an O(1)-size path/cycle/grid/torus.

Chang, Pettie'17

Brandt, Hirvonen, Korhonen, Lempiäinen, Östergård, Purcell, Rybicki, Suomela'17

Chang, Kopelowitz, Pettie'16

Brandt, Hirvonen, Korhonen, Lempiäinen, Östergård, Purcell, Rybicki, Suomela'17

Little white lies

- What does "n" refer to in the LOCAL model?
  (1) n = |V| = size of the graph.
  (2) O(log n) = bits in vertex IDs.
  (3) 1/poly(n) = standard error bound for ***<u>randomized</u>*** algs.

Time Hierarchies: Δ=O(1)

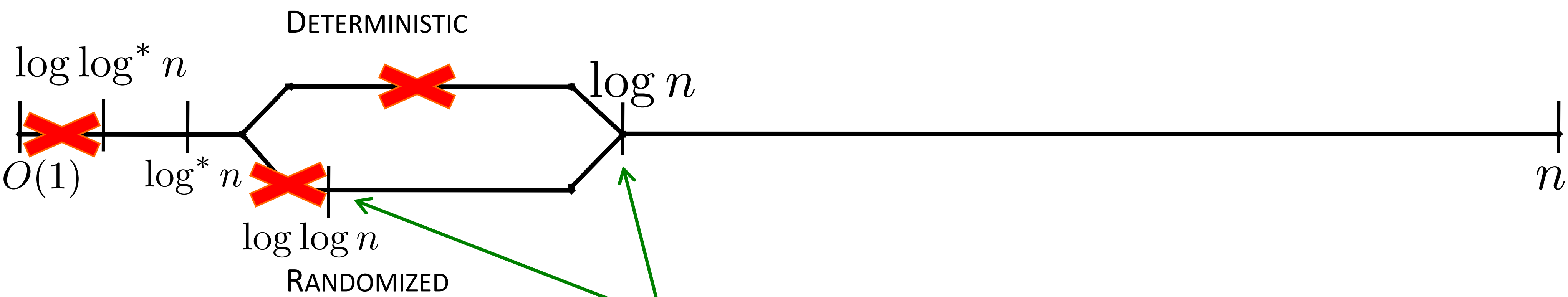
General graphs, Trees

Time Hierarchies: Δ=O(1)

General graphs, Trees

Chang, Kopelowitz, Pettie'16

$$\mathrm{Det}_{\mathcal{P}}(n,\Delta) \le \mathrm{Rand}_{\mathcal{P}}(2^{n^2},\Delta)$$



Exponential Separations:

- Δ-coloring degree-Δ trees
- Sinkless Orientation
- (2Δ-2)-edge coloring trees

Brandt, Fischer, Hirvonen, Keller, Lempiäinen, Rybicki, Suomela, Uitto'16

Chang, Kopelowitz, Pettie'16

Pettie, Su'15

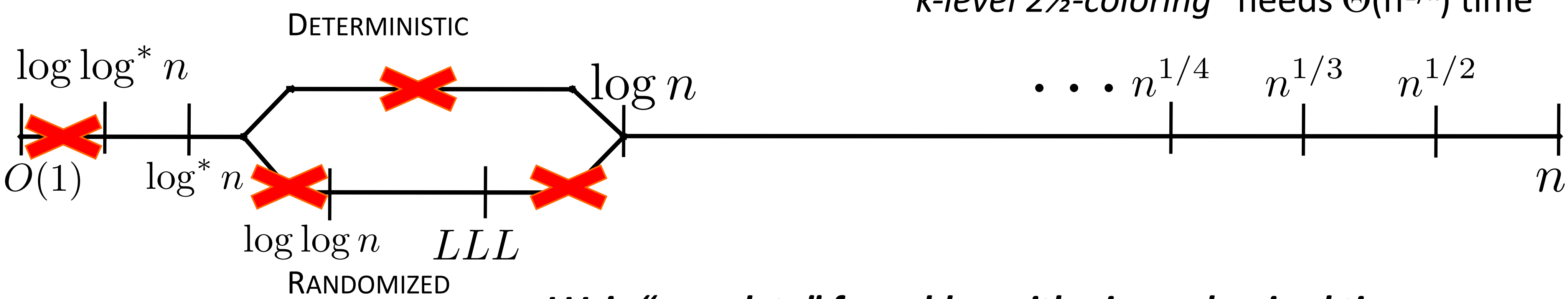
Ghaffari, Su'17

Chang, He, Li, Pettie, Uitto'18

Time Hierarchies: Δ=O(1)

General graphs, Trees

An infinite number of complexities.
"*k-level 2½-coloring*" needs $\Theta(n^{1/k})$ time

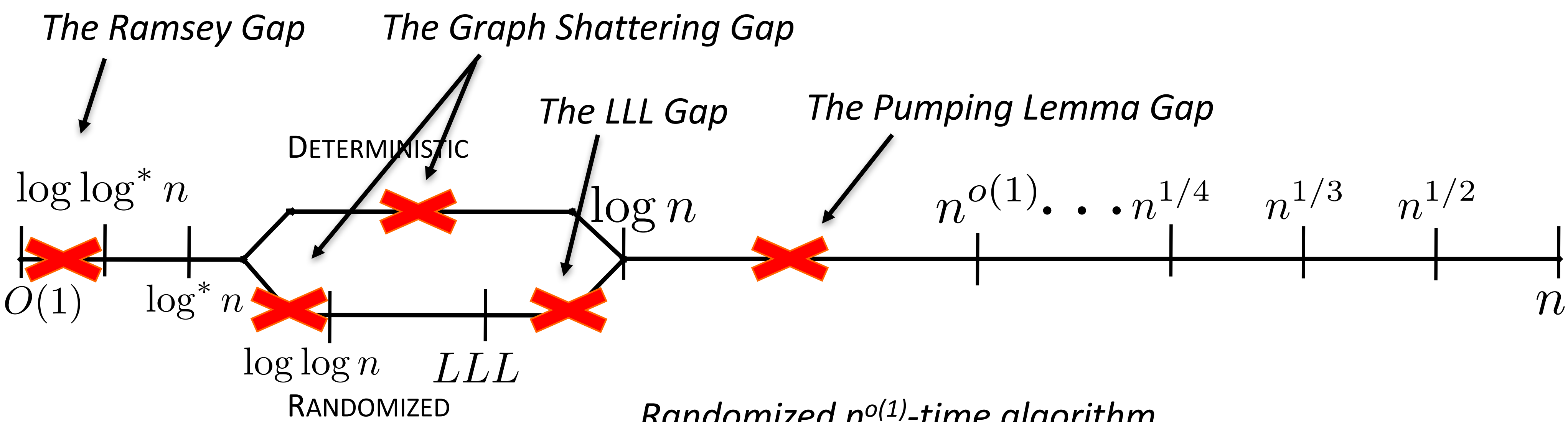


LLL is "complete" for sublogarithmic randomized time.

Every o(log n)-time randomized algorithm can be automatically sped up to run in O(LLL) time.

Time Hierarchies: Δ=O(1)

Trees



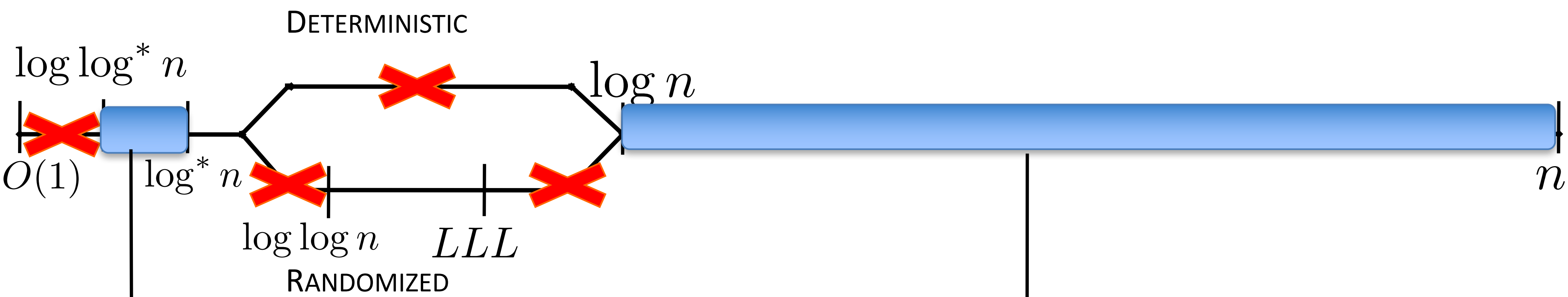
Randomized $n^{o(1)}$-time algorithm

↓

Deterministic O(log n)-time algorithm

Time Hierarchies: Δ=O(1)

General Graphs



$\log^{p/q}\log^* n$

$2^{\log^{q/p}\log^* n}$

$(\log^* n)^{q/p}$

$\log^{p/q} n$

$2^{\log^{q/p} n}$

$n^{pq/(p+q)^2}$

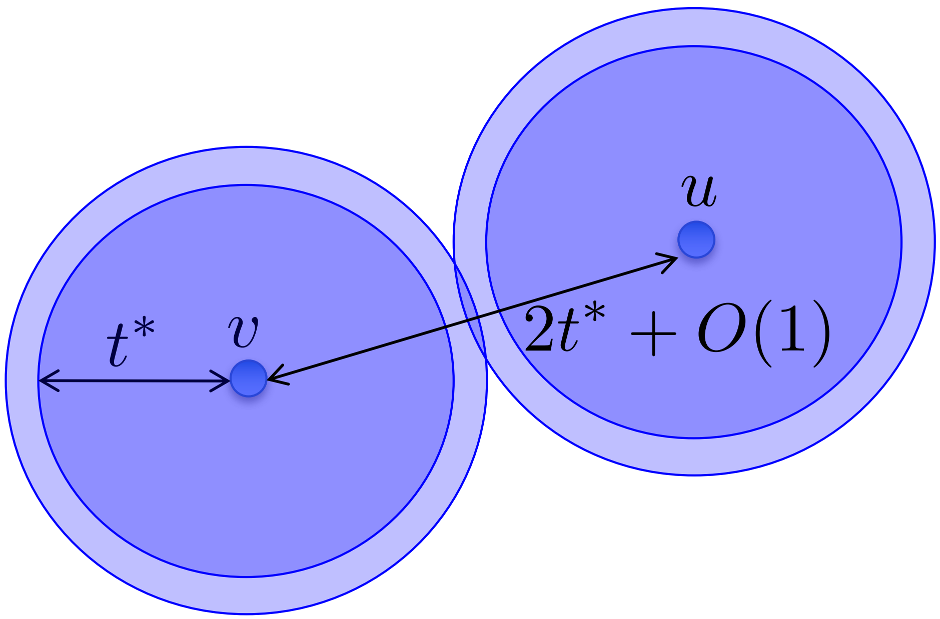
$p \geq q$

Balliu, Hirvonen, Korhonen, Keller, Lempiäinen, Olivetti, Suomela 2018

LLL complete for Randomized Sublogarithmic Time

- The ***distributed*** (symmetric) LLL problem:
  - Network and dependency graph $G=(V,E)$ are identical
  - V : "bad events"; $u \in V$ depends on set of discr. r.v.s $\mathrm{vbl}(u)$
  - $E = \{(u,v) : \mathrm{vbl}(u) \cap \mathrm{vbl}(v) \neq \emptyset\}$
  - ***d*** = maximum degree in G, ***p*** = maximum Pr(v).
    - Satisfies some ***LLL Criterion***, e.g., $ep(d+1) < 1$, $p(ed)^c < 1$.
- Compute a variable assignment such that no bad event occurs.

- Suppose A solves some LCL problem in *sub*logarithmic time with failure probability 1/n.
  - For any ε>0, can write time as $T(n,\Delta) \leq C(\Delta) + \epsilon \log_\Delta n$
- n* = min. value such that: $T(n,\Delta) = t^* \leq \frac{1}{2c} \log_\Delta n^* - O(1)$
  - Follows that t* = O(C(Δ)).



every vertex sees a subgraph that is *consistent with an n*-vertex graph.*

- Build the dependency graph:
  - X_v = the random bits generated locally at v.
  - vbl(v) = $\{X_u \mid u \in N^{t^*+O(1)}(v)\}$
  - E_v = the event that v's neighborhood is incorrectly labeled, when running alg. A with "n" = n^*.
  - $H = \left(\{E_v\},\ \{(E_u,E_v) \mid \text{dist}(u,v) \le 2t^*+O(1)\}\right)$
  - LLL parameters: $p = 1/n^*$, $d = \Delta^{2t^*+O(1)}$

$$pd^c = p \cdot \Delta^{c(2t^*+O(1))} < (1/n^*)n^* = 1$$

- Run a distributed LLL algorithm on "H."
  - 1 step in H simulated with $O(C(\Delta))$ steps in G.
  - Alg. A can be automatically sped up to $O(C(\Delta)\cdot T_{LLL})$ time.

The Distributed LLL

Time	LLL Criterion

The Distributed LLL

		Time	**LLL Criterion**
Moser, Tardos	2010	$O(\log^2 n)$	$ep(d+1) < 1$
Chung, Pettie, Su	2014	$O(\log n)$	$p(ed)^2 < 1$

The Distributed LLL

	Time	LLL Criterion
Moser, Tardos 2010	$O(\log^2 n)$	$ep(d+1) < 1$
Chung, Pettie, Su 2014	$O(\log n)$	$p(ed)^2 < 1$
Brandt, Fischer, Hirvonen, Keller, Lempiäinen, Rybicki, Suomela, Uitto 2016	$\Omega(\log \log n)$ (Rand.)	$p2^d < 1$
Chang, Kopelowitz, Pettie 2016	$\Omega(\log n)$ (Det.)	

The Distributed LLL

	Time	LLL Criterion
Moser, Tardos 2010	$O(\log^2 n)$	$ep(d+1) < 1$
Chung, Pettie, Su 2014	$O(\log n)$	$p(ed)^2 < 1$
Brandt, Fischer, Hirvonen, Keller, Lempiäinen, Rybicki, Suomela, Uitto 2016	$\Omega(\log\log n)$ (Rand.)	$p2^d < 1$
Chang, Kopelowitz, Pettie 2016	$\Omega(\log n)$ (Det.)	
conjecture	$\Theta(\log\log n)$	Some: $p(ed)^c < 1$

The Distributed LLL

	Time	LLL Criterion
Moser, Tardos 2010	$O(\log^2 n)$	$ep(d+1) < 1$
Chung, Pettie, Su 2014	$O(\log n)$	$p(ed)^2 < 1$
Brandt, Fischer, Hirvonen, Keller, Lempiäinen, Rybicki, Suomela, Uitto 2016	$\Omega(\log\log n)$ (Rand.)	$p2^d < 1$
Chang, Kopelowitz, Pettie 2016	$\Omega(\log n)$ (Det.)	
conjecture	$\Theta(\log\log n)$	Some: $p(ed)^c < 1$
Fischer, Ghaffari 2017	$O(d^2 + (\log n)^{O(1/c)})$	$p(ed)^c < 1$
	$2^{O(\sqrt{\log\log n})}$	$d < (\log\log n)^{1/5}$

The Distributed LLL

	Time	LLL Criterion
Moser, Tardos 2010	$O(\log^2 n)$	$ep(d+1) < 1$
Chung, Pettie, Su 2014	$O(\log n)$	$p(ed)^2 < 1$
Brandt, Fischer, Hirvonen, Keller, Lempiäinen, Rybicki, Suomela, Uitto 2016	$\Omega(\log \log n)$ (Rand.)	$p2^d < 1$
Chang, Kopelowitz, Pettie 2016	$\Omega(\log n)$ (Det.)	
conjecture	$\Theta(\log \log n)$	Some: $p(ed)^c < 1$
Fischer, Ghaffari 2017	$O(d^2 + (\log n)^{O(1/c)})$	$p(ed)^c < 1$
	$2^{O(\sqrt{\log \log n})}$	$d < (\log \log n)^{1/5}$
Chang, He, Li, Pettie, Uitto 2018 **(TREES ONLY)**	$O(\log \log n)$	$p(ed)^{O(1)} < 1$

The Distributed LLL

	Time	LLL Criterion
Moser, Tardos 2010	$O(\log^2 n)$	$ep(d+1) < 1$
Chung, Pettie, Su 2014	$O(\log n)$	$p(ed)^2 < 1$
Brandt, Fischer, Hirvonen, Keller, Lempiäinen, Rybicki, Suomela, Uitto 2016	$\Omega(\log\log n)$ (Rand.)	$p2^d < 1$
Chang, Kopelowitz, Pettie 2016	$\Omega(\log n)$ (Det.)	
conjecture	$\Theta(\log\log n)$	Some: $p(ed)^c < 1$
Fischer, Ghaffari 2017	$O(d^2 + (\log n)^{O(1/c)})$	$p(ed)^c < 1$
	$2^{O(\sqrt{\log\log n})}$	$d < (\log\log n)^{1/5}$
Chang, He, Li, Pettie, Uitto 2018 **(TREES ONLY)**	$O(\log\log n)$	$p(ed)^{O(1)} < 1$
Ghaffari, Harris, Kuhn 2017	$\exp^{(c)}(O(\sqrt{\log^{(c+1)} n}))$	$pd^8 = O(1)$

Open Questions

- What is the LOCAL complexity of the Lovász Local Lemma with some poly. criterion $p(ed)^{O(1)} < 1$?
  - Probably need to solve rand. and det. complexities simultaneously. $\Theta(\log\log n)$ rand. and $\Theta(\log n)$ det.?

- Define LLL[c] to be the problem with criterion pd^c=O(1). Is LLL[1] (the "real" LLL) strictly harder than LLL[O(1)] ?

- Is there an $\omega(1)$—$o(\log^* n)$ complexity gap on trees?