3 Deterministic vs Randomized complexities

Theorem 0.1 (Linial [Lin92]). *Let* $\mathbf{G}$ *be the class of all graphs of degree bounded by* Δ *and* Π *be the proper vertex* $(\Delta+1)$*-coloring problem. Then we have*

$$\Pi \in \mathsf{LOCAL}(O(\log^* n)).$$

Proof Sketch. The proof is based on an iterated application of the following combinatorial result on *cover free set systems.*

Claim 0.2 (Section 4.8.2 in [**?**]). *For every* $n > \Delta > 2$ *there is* $m \leq 4(\Delta+1)^2 \log^2(n)$ *and a family of subsets* $(S_i)_{i\in[n]}$ *of* $[m]$ *such that*

$$S_{i_0} \not\subseteq \bigcup_{\ell\in[\Delta]} S_{i_\ell},$$

whenever $i_0 \neq i_\ell$ *for every* $\ell \in [\Delta]$.

Start with a graph G on n vertices with unique identifiers from $\mathrm{poly}(n)$ and then use Claim 0.2 iteratively to decrease the colors to $\mathrm{poly}(\Delta)$ in $\log^* n$ rounds as follows. Suppose that we have coloring c_x with x-colors and x is larger than e.g. $(11(\Delta+1))^3$, see [**?** , Section 4.8.3]. Pick m and $(S_i)_{i\in x}$ as in Claim 0.2. In one communication round let $c'(v)$ be the lexicographically minimal color in

$$S_{c_x(v)} \setminus \bigcup_{(v,w)\in E(G)} S_{c_x(w)}.$$

Then c' is a well defined m-coloring. It can be shown that as long as $\log^{(i)} x \geq \log 36 + 2\log(\Delta+1)$, then we decrease the number of colors from

$$(6(\Delta+1)\log^{(i-1)} x)^2 \text{ to } (6(\Delta+1)\log^{(i)} x)^2$$

in the i-th step, where $i > 2$.

Last step, as we treat Δ as a constant, is to use the greedy algorithm. In general, this takes as many rounds as we have colors, in our case $\mathrm{poly}(\Delta)$. □

Another general deterministic speed up is from $o(\log\log^* n)$ to $O(1)$, [NS95, CP17].

Randomness. Every deterministic local algorithm can be simulated as a randomized algorithm of the same complexity.

Claim 0.3. *Let* $\Pi \in \mathsf{LOCAL}(O(f(n)))$. *Then* $\Pi \in \mathsf{RLOCAL}(O(f(n)))$.

Proof. Let $m \geq n$ and split $[0,1]$ into m-intervals $(I_j)_j$ of the same size. Let $v \neq w \in V(G)$. Then the probability that their labels do not differ is $1/m$. By the union bound, we have that the probability of having a pair with a label in the same interval is at most $\frac{n^2}{m}$. Consequently, taking $m = n^3$ yields the result. □

Results in the opposite direction are called *derandomization*. It is known that this can be done in the regime $o(\log\log n)$, [CKP16]. In particular cases, e.g., grids this can be improved to $o(\log n)$.

Proposition 0.4 (Corollary 1 in [**?**]). *Let* $\mathbf{G}$ *be closed under disjoint unions of finite graphs, or suppose that each* $G \in \mathbf{G}_n$ *can be found as an induced subgraph of some graph in* $\mathbf{G}_m$ *for every* $m > n$*. Then*

$$\mathsf{RLOCAL}(O(f(n))) = \mathsf{LOCAL}(O(f(n)))$$

for every $f \in O(\log^* n)$*.*

Proof Sketch. Let $\mathcal{A}$ be a randomized local algorithm of complexity $(r_n)_n \in O(\log^* n)$ that solves an LCL Π of locality $t \in \mathbb{N}$. Fix $n \in \mathbb{N}$. The idea is to find a (deterministic) function $\varphi : [n] \to [0,1]$ and $m(n) > n$ such that running $\mathcal{A}$ for $r_{m(n)}$ many rounds on any $G \in \mathbf{G}_n$ labeled with images of unique identifiers under $\varphi(-)$ produces Π-coloring. This will certainly produce a deterministic local algorithm of complexity $(r_{m(n)})_n$.

We show that $m(n) = 2^{n^2}$ works as required. Note that in that case, if $f \in O(\log^* n)$, then $f(m(n)) \in O(f(n))$. Let $\mathcal{G}_n \in \mathbf{G}_{m(n)}$ be a graph that contains as induced subgraph the graph $\mathcal{H}_n$ that is created by taking disjoint copies of all graphs from $\mathbf{G}_n$ labeled with unique identifiers. Note that $|\mathcal{H}_n| \ll m(n)$. Pick φ uniformly at random. By the definition the probability of the event that $\mathcal{A}_{m(n)}$ fails at a given vertex $v \in V(G)$ for some $G \in \mathbf{G}_n$ (labeled with images of unique identifiers under φ) is at most $\frac{1}{m(n)}$ by the definition. As $|\mathcal{H}_n| \ll m(n)$ we get by the union bound that there exists φ that does not fail at any vertex. □

In general, randomness does help in the regime $o(\log n)$. Picture. For example, deterministic complexity of the *sinkless orientation* problem on trees is $\Omega(\log n)$, however, randomized complexity is $O(\mathrm{poly}(\log \log n))$. This is the most significant complexity class that is tightly connected with the distributed complexity of solutions to the *Lovász Local Lemma (LLL)*. On a high level, given a randomized algorithm $\mathcal{A}$ that solves Π in $o(\log n)$ rounds. We consider the *bad events* around vertices, when $\mathcal{A}$ fails. Then fixing k large enough we try to avoid bad events of $\mathcal{A}$ when applied to graphs of size k. Note that uniformly the probability of bad event p is less than $\frac{1}{k}$ and the degree d of bad event in the auxiliary graph of bad events is at most $\Delta^{o(\log k)} \ll k$. That is the LLL condition $p \cdot d \ll 1$ is satisfied. More on this later.

References

[CKP16] Yi-Jun Chang, Tswi Kopelowitz, and Seth Pettie. An exponential separation between randomized and deterministic complexity in the LOCAL model. In *Proc. 57th IEEE Symp. on Foundations of Computer Science (FOCS)*, 2016.

[CP17] Yi-Jun Chang and Seth Pettie. A time hierarchy theorem for the LOCAL model. In *Proc. 58th IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 156–167, 2017.

[Lin92] Nati Linial. Locality in distributed graph algorithms. *SIAM Journal on Computing*, 21(1):193–201, 1992.

[NS95] Moni Naor and Larry Stockmeyer. What can be computed locally? *SIAM Journal on Computing*, 24(6):1259–1277, 1995.