4 Uniform complexities

Suppose that we have a uniform local algorithm $\mathcal{A}$. Then it is easy to see that we can run it as a randomized local algorithm. Simply pick a function $(r_n)_n$ and simulate the algorithm on the family $\mathbf{G}_n$ with the additional constraint that it has to stop after r_n-many rounds. It is clear that if $r_n = \max\{\mathrm{diam}(G) : G \in \mathbf{G}_n\}$, then, as $\mathcal{A}$ is defined on $\mathbf{G}$, it will output correct solution with probability 1. As we are allowed to have an error of order $1/n$ we get the following result by a simple use of the union bound.

Proposition 0.1. *Let $\mathcal{A}$ be a uniform local algorithm of uniform complexity $f(\epsilon)$ that is defined on a class of graphs $\mathbf{G}$. Then the nonuniform randomized complexity of $\mathcal{A}$ is $f(1/n^2)$.*

Proof. Let $G \in \mathbf{G}_n$ and $r_n = f(1/n^2)$. As the probability of each vertex to need a radius bigger than r_n is bounded by $\frac{1}{n^2}$ by the definition of uniform complexity, we see that the probability that the output is not defined on some vertex of G is bounded by $n(1/n^2) = 1/n$. Note that once the output is defined, then it is correct with probability 1, consequently the radius r_n works as required. □

It can, of course, happen that $f(1/n^2) > \max\{\mathrm{diam}(G) : G \in \mathbf{G}_n\}$ in which case we do not get anything non-trivial. For example, if the complexity is $f(\epsilon) = 1/\sqrt{\epsilon}$, then we get $r_n = n$.

Next we investigate the other implication. Suppose that we have a uniform algorithm of some non-uniform complexity. What is its uniform complexity? Here we need to have an additional assumption.

Proposition 0.2. *Let $\mathcal{A}$ be a uniform local algorithm of non-uniform complexity $r(n)$ and suppose that $\mathbf{G}$ satisfies that every element of $\mathbf{G}_\bullet$ of radius r_n can be found in a graph on n vertices. Then the uniform complexity of $\mathcal{A}$ is $r(1/\epsilon)$.*

Proof. Let $v \in V(G)$ for some $G \in \mathbf{G}$ does not satisfy

$$\mathbf{P}(R_{\mathcal{A}} > r(n)) < 1/n.$$

By the assumption we may assume that $G \in \mathbf{G}_n$. But then we have that the probability of no output at v is bigger than $1/n$ and, consequently, the probability of correct solution of $\mathcal{A}$ on G is less than $1 - 1/n$. □

As above the assumption is satisfied if $(\mathbf{G}_n)_n$ is a suitable "approximation" of $\mathbf{G}$ and e.g. $r(n) \in o(\log n)$.

Making non-uniform algorithms uniform Note that in the results above we assumed that we started with some uniform algorithm and then compare the uniform and non-uniform complexities under some necessary assumptions. The opposite task, that is, turning non-uniform algorithms to uniform, seems to be much more intricate. For example Theorem 0.3 is a uniformization of **??** and together with our complete understanding of the class $\mathsf{LOCAL}(O(\log^* n))$, we can say that the uniform and non-uniform complexities are equal in this regime. The case of LLL, that is $\mathsf{RLOCAL}(\mathrm{poly}(\log\log n))$ is not understood in general. In particular, it is not clear if the distrbuted LLL algorithm admits uniform version.

Next, we show that Linial's algorithm has a uniform version.

Theorem 0.3 ([HSW17])**.** *Let $\mathbf{G}$ be the class of graphs of degree bounded by $\Delta < \infty$. Then the proper vertex $(\Delta + 1)$-coloring problem is in the class $\mathsf{ULOCAL}(O(\log^* 1/\varepsilon))$, that is to say,*

$$P(\mathcal{R} > O(\log^* 1/\varepsilon)) < \varepsilon.$$

Proof. Let $k \in \mathbb{N}$ and n_k be such that $n_k = \min\{n \in \mathbb{N} : \log^* n = k\}$. We construct an *almost coloring* with $\mathrm{poly}(\Delta) \cup \{\infty\}$ colors with locality k, where by almost coloring we mean that the restriction to first $\mathrm{poly}(\Delta)$ colors is a proper coloring and the last color, ∞, does not have to satisfy anything.

Turn the randomness to labeling from the range n_k as in **??**. Assign to vertices that share the same label with a neighbor the special color ∞. Note that probability of such vertices is at most Δ/n_k. Also note that to run the reduction as in **??** we do not need to have unique identifiers, but just a coloring. In $\log^* n_k$-rounds, i.e., k-rounds, we produce a coloring with $\mathrm{poly}(\Delta)$-colors. Together with the label ∞ we have constructed an almost coloring of the whole graph.

Write c_k for the constructed coloring with range $Z \cup \{\infty\}$, where Z is of size $\mathrm{poly}(\Delta)$, and X_k for the c_k-preimage of Z. Set $(\Delta + 1)$ to be a distinguished set of colors disjoint from Z. We define inductively coloring d_k as follows. Let d_0 be the greedy modification of c_0 on X_0 to a $(\Delta+1)$-coloring. Suppose that d_k has been defined on X_k. Let d_{k+1} be the greedy modification of

$$c_{k+1} \restriction (X_{k+1} \setminus X_k) \cup d_k \restriction X_k$$

on $(X_{k+1} \setminus X_k)$. Note that d_k is a proper vertex $(\Delta + 1)$-coloring on X_k. Moreover, going from d_k to d_{k+1} requires $\mathrm{poly}(\Delta)$ communication rounds on the set $X_{k+1} \setminus X_k$. That is, if $x \in X_k$, then it knows its final color after $\mathrm{poly}(\Delta)(k + 1)$ rounds. In another words

$$\mathrm{P}(\mathcal{R} > O(k)) \leq \Delta/n_k.$$

To finish, we fix $\varepsilon > 0$. Let k be such that $n_{k-1} \leq 1/\varepsilon \leq n_k/\Delta$. Then we have

$$\mathrm{P}(\mathcal{R} > O(1 + \log^* 1/\varepsilon) \leq \mathrm{P}(\mathcal{R} > O(k)) \leq \Delta/n_k \leq \varepsilon$$

and the proof is finished. □

Examples In general, we do not have a clear understanding what should be the relationship between infinite and finite graphs in $\mathbf{G}$. For example when we talk about uniform algorithms, we usually assume that the family $\mathbf{G}$ consists of a single infinite graph and in that case $\mathbf{G}_n$ is empty and the statements above do not say anything. Is it always possible to extend the domain of the algorithm to some class of suitable approximations?

The following results illustrate what issues might occur.

Theorem 0.4. *Let Π be the perfect matching problem. Then on some finite tori Π does not admit any solution, while we have*

$$\Pi \in \mathsf{ULOCAL}(\varepsilon^{1+o(1)})$$

on the infinite grid (for any dimension $d > 1$).

This shows that not every uniform algorithm can be extended to (even a very good) finite approximations. On the other hand, sometimes this is possible.

Theorem 0.5. *Let Π be the proper vertex 3-coloring problem. Then Π is global for non-uniform randomized algorithms, while we have*

$$\Pi \in \mathsf{ULOCAL}(\varepsilon^{1+o(1)})$$

on the infinite grid (for any dimension $d > 1$). This algorithm can be simulated on finite tori, in the sense that some vertices need to see the whole graph.

In the opposite direction, we have the following example on trees.

Theorem 0.6. *Let* Π *be the proper vertex* 3*-coloring problem. Then* $\Pi \in \mathsf{LOCAL}(O(\log n))$ *on finite trees but there is no uniform algorithm for the infinite* Δ*-regular tree.*

Nevertheless, on infinite graphs we get a finer scale to measure complexity somewhere between the LLL class and the global problems, e.g., $\Omega(\log n)$. Note that the following example shows that there is no correspondence between $\mathsf{ULOCAL}(O(\log 1/\varepsilon)$ and $\mathsf{LOCAL}(O(\log n))$.

Theorem 0.7. *Let* Π *be the perfect matching problem in the power* 2 *graph, that is, we add edges that connect vertices of distance exactly* 2*. Then on some finite trees* Π *does not admit any solution, while we have*

$$\Pi \in \mathsf{ULOCAL}(\operatorname{poly}\log 1/\varepsilon)$$

on the infinite Δ*-regular tree, where* $d > 2$.

So far, all these issues have been addressed in case-by-case fashion. For example, on trees we might consider finite trees, large girth graphs or half-edge formalism. Bernshteyn [Ber21] used finite connected subsets of a given Cayley graphs as approximations in the $\log^* n$ regime. On grids we use finite tori. In all cases, the infinite graphs in **G** is a single graph. Maybe, the right generality for **G** are the unimodular random networks, or closed (compact) subsets of rooted trees invariant under moving the root. But it is not clear what should be the finite approximation.

References

[Ber21] Anton Bernshteyn. Probabilistic constructions in continuous combinatorics and a bridge to distributed algorithms. 2021.

[HSW17] Alexander E. Holroyd, Oded Schramm, and David B. Wilson. Finitary coloring. *Ann. Probab.*, 45(5):2867–2898, 2017.