An Improved Distributed Algorithm for Maximal Independent Set

Mohsen Ghaffari
MIT
ghaffari@mit.edu

Abstract

The Maximal Independent Set (MIS) problem is one of the basics in the study of *locality* in distributed graph algorithms. This paper presents a very simple randomized algorithm for this problem providing a near-optimal *local complexity*, which incidentally, when combined with some known techniques, also leads to a near-optimal *global complexity*.

Classical MIS algorithms of Luby [STOC'85] and Alon, Babai and Itai [JALG'86] provide the *global complexity* guarantee that, with high probability[1], *all nodes* terminate after $O(\log n)$ rounds. In contrast, our initial focus is on the *local complexity*, and our main contribution is to provide a very simple algorithm guaranteeing that *each* particular node v terminates after $O(\log \mathsf{deg}(v) + \log 1/\varepsilon)$ rounds, with probability at least $1-\varepsilon$. The degree-dependency in this bound is optimal, due to a lower bound of Kuhn, Moscibroda, and Wattenhofer [PODC'04].

Interestingly, this local complexity smoothly transitions to a global complexity: by adding techniques of Barenboim, Elkin, Pettie, and Schneider [FOCS'12; arXiv: 1202.1983v3], we[2] get an MIS algorithm with a high probability global complexity of $O(\log \Delta) + 2^{O(\sqrt{\log \log n})}$, where Δ denotes the maximum degree. This improves over the $O(\log^2 \Delta) + 2^{O(\sqrt{\log \log n})}$ result of Barenboim et al., and gets close to the $\Omega(\min\{\log \Delta, \sqrt{\log n}\})$ lower bound of Kuhn et al.

Corollaries include improved algorithms for MIS in graphs of upper-bounded arboricity, or lower-bounded girth, for Ruling Sets, for MIS in the Local Computation Algorithms (LCA) model, and a faster distributed algorithm for the Lovász Local Lemma.

1 Introduction and Related Work

Locality sits at the heart of distributed computing theory and is studied in the context of problems such as Maximal Independent Set (MIS), Maximal Matching (MM), and Coloring. Over time, MIS has been of special interest as the others reduce to it. The story can be traced back to the surveys of Valiant [Val82] and Cook [Coo83] in the early 80's which mentioned MIS as an interesting problem in non-centralized computation, shortly after followed by (poly-)logarithmic algorithms of Karp and Wigderson [KW84], Luby [Lub85], and Alon, Babai, and Itai [ABI86]. Since then, this problem has been studied extensively. We refer the interested reader to [BEPSv3, Section 1.1], which provides a thorough and up to date review of the state of the art.

In this article, we work with the standard distributed computation model called LOCAL [Pel00]: the network is abstracted as a graph $G = (V, E)$ where $|V| = n$; initially each node only knows its neighbors; communications occur in synchronous rounds, where in each round each node can exchange information only with its graph neighbors.

Note that in the LOCAL model, all that a node can learn in k rounds is the subgraph induced by its k-hop neighborhood (and the random bits of the nodes in it). Because of this, the round complexity of a problem has a purely graph-theoretic meaning: it identifies the radius up to which one needs to look to determine the output of each node, e.g., its color in a coloring. For instance, results of [Lub85, ABI86] imply that looking only at the $O(\log n)$-hop neighborhood suffices, w.h.p.

1.1 Local Complexity

Despite the local nature of the MIS problem, classically the main focus has been on the global complexity, i.e., the time until all nodes terminate. Moreover, somewhat strikingly, essentially all the standard analyses also take a non-local approach by considering the whole graph and showing guarantees on how the algorithm makes a *global progress* towards it *local objectives*. A prominent example is the analysis of [Lub85, ABI86] which show that per round, in expectation, half of the edges of the whole network get removed, hence leading to the *global complexity* guarantee that after $O(\log n)$ rounds, w.h.p., the algorithm terminates everywhere. See [MR10, Section 12.3], [Pel00, Section 8.4], [Lyn96, Section 4.5] for textbook treatments or [Win04] for a simpler analysis.

This issue seemingly suggests a gap in our understanding of *locality*. The starting point in this paper is to question whether this global mentality is necessary.

[1] As standard, we use the phrase *with high probability* (w.h.p.) to indicate that an event has probability at least $1 - 1/n^c$, for an arbitrary constant $c > 1$.

[2] *quasi nanos, gigantium humeris insidentes*

Particularly, can we provide an analysis that only looks at a node and some small neighborhood of it, and provides a guarantee for that node independent of n? To be concrete, our starting question is:

> **Local Complexity Question**: How long does it take till each particular node v terminates, and knows whether it is in the (eventual) MIS or not, with probability at least $1 - \varepsilon$?

Using Δ to denote the maximum degree, one can obtain answers such as $O(\log^2 \Delta + \log 1/\varepsilon)$ rounds for Luby's algorithm, or $O(\log \Delta \log\log \Delta + \log \Delta \log 1/\varepsilon)$ rounds for (a parameter optimized version) of the variant of Luby's used by Barenboim, Elkin, Pettie, and Schneider [BEPSv3] and Chung, Pettie, and Su [CPS14]. However, both of these bounds seem to be off from the right answer; e.g., we cannot get from them the standard $O(\log n)$ high probability global complexity bound. In both, the bound becomes $O(\log^2 n)$ when one sets $\Delta = n^\delta$ for a constant $\delta > 0$ and $\varepsilon = 1/n$.

We present an extremely simple algorithm that overcomes this problem and provides a local complexity of $O(\log \Delta + \log 1/\varepsilon)$. More formally, we prove that:

THEOREM 1.1. *There is a randomized distributed MIS algorithm such that for each node v, the probability that v has not made its decision after the first $O(\log \deg(v) + \log 1/\varepsilon)$ rounds is at most ε. Furthermore, this holds even if the bits of randomness outside the 2-hop neighborhood of v are determined adversarially.*

The perhaps surprising fact that the bound only depends on the degree of node v, even allowing its neighbors to have infinite degree (as $n \to \infty$), demonstrates the *truly local* nature of this algorithm. The logarithmic degree-dependency in the bound is optimal, following a lower bound of Kuhn, Moscibroda and Wattenhofer [KMWv1], e.g., in the following sense: As indicated by [Kuh15], with minor changes in the arguments of [KMWv1], one can prove that there are graphs in which, the time until each node v can know if it is in MIS or not with constant probability is at least $\Omega(\log \Delta)$.

Finally, we note that the fact that *the proof has a locality of 2-hop*—meaning that the analysis only looks at the 2-hop neighborhood and particularly, that the guarantee relies only on the coin tosses within the 2-hop neighborhood of node v—will prove vital as we move to global complexity. This might be interesting for practical purposes as well.

1.2 Global Complexity Notice that Theorem 1.1 easily recovers the standard result that after $O(\log n)$ rounds, w.h.p., all nodes have terminated, but now with a local analysis. In light of the $\Omega(\min\{\log \Delta, \sqrt{\log n}\})$ lower bound of Kuhn et al. [KMWv1], it is interesting to find the best possible upper bound, specially when $\log \Delta = o(\log n)$. The best known bound prior to this work was $O(\log^2 \Delta) + 2^{O(\sqrt{\log\log n})}$ rounds, due to Barenboim et al. [BEPSv3].

The overall plan is based on the following nice and natural intuition, which was used in the MIS results of Alon et al. [ARVX12] and Barenboim et al. [BEPSv3]. We note that this general strategy is often attributed to Beck, as he used it first in his breakthrough algorithmic version of the Lovász Local Lemma [Bec91]. Applied to MIS, the intuition is that, when we run any of the usual randomized MIS algorithms, nodes get removed probabilistically more and more over time. After running this *base algorithm* for a certain number of rounds, a *graph shattering* phenomenon occurs, where what remains of the graph is a number of "small" components. Here, small might be in regard to size, (weak) diameter, the maximum size of some specially defined independent sets, or some other measure. Once the graph is shattered, one switches to a deterministic algorithm to *finish off* the problem in these remaining small components.

Since we are considering graphs with max degree Δ, even ignoring the troubling probabilistic dependencies (which are actually rather important), a simplistic intuition based on *Galton-Watson branching processes* tells us that the graph shattering phenomena starts to show up around the time that the probability ε of each node being left falls below $1/\Delta$[3]. Alon et al. [ARVX12] used an argument of Parnas and Ron [PR07], showing that Luby's algorithm reaches this threshold after $O(\Delta \log \Delta)$ rounds. Barenboim et al. [BEPSv3] used a variant of Luby's, with a small but clever modification, and showed that it reaches the threshold after $O(\log^2 \Delta)$ rounds. As Barenboim et al. [BEPSv3] show, after the shattering, the remaining pieces can be solved deterministically, via the help of known deterministic MIS algorithms (and some other ideas), in $\log \Delta \cdot 2^{O(\sqrt{\log\log n})}$ rounds. Thus, the overall complexity of [BEPSv3] is $O(\log^2 \Delta) + \log \Delta \cdot 2^{O(\sqrt{\log\log n})} = O(\log^2 \Delta) + 2^{O(\sqrt{\log\log n})}$.

To improve this, we use our new MIS algorithm as the base, instead of Luby's,, which as Theorem 1.1 suggests, reaches the shattering threshold after $O(\log \Delta)$ rounds. This is formalized in Section 4. We will also use some minor modifications for the *post-shattering* phase

[3] In truth, the probability threshold is $1/\operatorname{poly}(\Delta)$, because of some unavoidable dependencies. But due to the exponential concentration, the time to reach the $1/\operatorname{poly}(\Delta)$ threshold is within a constant factor of that of the $1/\Delta$ threshold. We will also need to establish some independence. See Section 4.

to reduce its complexity from $\log\Delta \cdot 2^{O(\sqrt{\log\log n})}$ to $2^{O(\sqrt{\log\log n})}$. The overall result thus becomes:

THEOREM 1.2. *There is a randomized distributed MIS algorithm that terminates after $O(\log\Delta)+2^{O(\sqrt{\log\log n})}$ rounds, with probability at least $1-1/n$.*

This improves the best-known bound for MIS and gets close to the $\Omega(\min\{\log\Delta, \sqrt{\log n}\})$ lower bound of Kuhn et al. [KMWv1], which at the very least, shows that the upper bound is provably optimal when $\log\Delta \in [2^{\sqrt{\log\log n}}, \sqrt{\log n}]$. Besides that, the new result matches the lower bound in a stronger and much more instructive sense: as we will discuss in point (C2) below, it perfectly pinpoints why the current lower bound techniques cannot prove a lower bound better than $\Omega(\min\{\log\Delta, \sqrt{\log n}\})$.

1.3 Other Implications Despite its extreme simplicity, the new algorithm turns out to lead to several implications, when combined with some known results and/or techniques:

(C1) Combined with the finish-off phase results of Barenboim et al. [BEPSv3], we get MIS algorithms with complexity $O(\log\Delta)+O(\min\{\lambda^{1+\varepsilon}+\log\lambda\log\log n, \lambda+\lambda^{\varepsilon}\log\log n, \lambda+(\log\log n)^{1+\varepsilon}\})$ for graphs with arboricity λ. Moreover, combined with the low-arboricity to low-degree reduction of Barenboim et al. [BEPSv3], we get an MIS algorithm with complexity $O(\log\lambda+\sqrt{\log n})$. This improves on some results of [BEPSv3, BE10, LW11].

(C2) The new results highlight the barrier of the current lower bound techniques. In the known locality-based lower bound arguments, including that of [KMWv1], to establish a T-round lower bound, it is necessary that within T rounds, each node sees only a tree. That is, each T-hops neighborhood must induce a tree, which implies that the girth must be at least $2T+1$. Since any g-girth graph has arboricity $\lambda \leq O(n^{\frac{2}{g-2}})$, from (C1), we get an $O(\sqrt{\log n})$-round MIS algorithm when $g=\Omega(\sqrt{\log n})$. More precisely, for any graph with girth $g=\Omega(\min\{\log\Delta, \sqrt{\log n}\})$, we get an $O(\min\{\log\Delta+2^{O(\sqrt{\log\log n})}, \sqrt{\log n}\})$-round algorithm. Hence, the $\Omega(\min\{\log\Delta, \sqrt{\log n}\})$ lower bound of [KMWv1] is essentially the best-possible when the topology seen by each node within the allowed time must be a tree. This means, to prove a better lower bound, one has to part with these *"tree local-views"* topologies. However, that gives rise to intricate challenges and actually, to the best of our knowledge, there is no distributed locality-based lower bound, in fact for any (local) problem, that does not rely on *tree local-views*.

(C3) We get an $O(\sqrt{\log n})$-round MIS algorithm for Erdös-Rényi random graphs $G(n,p)$. This is because, if $p=\Omega(\frac{2^{\sqrt{\log n}}}{n})$, then w.h.p. the graph has diameter $O(\sqrt{\log n})$ hops (see e.g. [CL01]) and when $p=O(\frac{2^{\sqrt{\log n}}}{n})$, with high probability, $\Delta=O(2^{\sqrt{\log n}})$ and thus, the algorithm of Theorem 1.2 runs in at most $O(\sqrt{\log n})$ rounds.

(C4) Combined with a recursive sparsification method of Bisht et al. [BKP14], we get a $(2,\beta)$-ruling-set algorithm with complexity $O(\beta\log^{1/\beta}\Delta)+2^{O(\sqrt{\log\log n})}$, improving on the complexities of [BEPSv3] and [BKP14]. An (α,β)-ruling set S is a set where each two nodes in S are at distance at least α, and each node $v\in V\setminus S$ has a node in S within its β-hops. So, a $(2,1)$-ruling-set is simply an MIS. The term $O(\beta\log^{1/\beta}\Delta)$ is arguably best-possible for the current method, which roughly speaking works by computing the ruling set iteratively using β successive reductions of the degree.

(C5) In the Local Computation Algorithms (LCA) model of Rubinfeld et al. [RTVX11] and Alon et al. [ARVX12], we get improved bounds for computing MIS. Namely, the best-known time and space complexity improve from, respectively, $2^{O(\log^3\Delta)}\log^3 n$ and $2^{O(\log^3\Delta)}\log^2 n$ bounds of [LRY15] to $2^{O(\log^2\Delta)}\log^3 n$ and $2^{O(\log^2\Delta)}\log^2 n$.

(C6) We get a Weak-MIS algorithm with complexity $O(\log\Delta)$, which thus improves the round complexity of the distributed algorithmic version of the Lovász Local Lemma presented by Chung, Pettie, and Su [CPS14] from $O(\log_{\frac{1}{ep(\Delta+1)}} n\cdot\log^2\Delta)$ to $O(\log_{\frac{1}{ep(\Delta+1)}} n\cdot\log\Delta)$. Roughly speaking, a Weak-MIS computation should produce an independent set S such that for each node v, with probability at least $1-1/\operatorname{poly}(\Delta)$, v is either in S or has a neighbor in S.

(C7) We get an $O(\log\Delta+\log\log\log n)$-round MIS algorithm for the CONGESTED-CLIQUE model where per round, each node can send $O(\log n)$-bits to each of the other nodes (even those not adjacent to it): After running the MIS algorithm of Theorem 1.1 for $O(\log\Delta)$ rounds, w.h.p., if $\Delta\geq n^{0.1}$, we are already done, and otherwise, as Lemma 4.2 shows, all leftover components have size $o(n^{0.5})$. In the latter case, using the algorithm of [HPP+15], we can make all nodes know the *leader* of their component in $O(\log\log\log n)$ rounds, and using Lenzen's

routing [Len13], we can make each leader learn the topology of its whole component, solve the related MIS problem locally, and send back the answers, all in $O(1)$ rounds.

2 Warm Up: Luby's Algorithm

Here, we briefly review Luby's algorithm [Lub85, ABI86]. The main purpose is to point out the challenge in (tightly) analyzing the local complexity of this algorithm, which the algorithm of the next section tries to bypass. The reader can choose to skip this section.

> **Luby's Algorithm** "*In each* `round`*, each node picks a random number*[4] *uniformly from* $[0, 1]$*; strict local minimas join the MIS, and get removed from the graph along with their neighbors.*"

Each round of the algorithm can be implemented in 2 communication rounds on G, one for exchanging the random numbers and the other for informing neighbors of newly joined MIS nodes. Ignoring this 2 factor, in the sequel, a *round* means one round of the algorithm.

2.1 Local Analysis: Take 1 To analyze the algorithm's local complexity, the natural idea is to say that over time, each local neighborhood gets "simplified", e.g., that the degrees shrink with time. The following observation is the base tool in this argument, the proof of which is left as a simple excercise.

LEMMA 2.1. *Consider a node u, let $d(u)$ be its degree and d_{max} be the maximum degree among its neighbors, at a particular round. The probability that u gets removed in this round is at least $\frac{d(u)+1}{d(u)+d_{max}}$.*

From this lemma, we get that in $O(1)$ rounds from the start, with probability at least $1/2$, either u is removed or its degree falls below $\Delta/2$. We would like to continue this argument and say that in every $O(1)$ rounds, u's degree shrinks by another 2 factor, thus getting a bound of $O(\log \Delta)$. However, this is not straightforward as u's degree drops might get delayed because of delays in the degree drops of u's neighbors. In fact the issue is rather sever as the degree drops of different nodes can be positively correlated.

One can try to handle these dependencies in a pessimistic way, using union bound. An argument based on this approach can be given which shows a local complexity of $O(\log^{2.5} \Delta + \log \Delta \log 1/\varepsilon)$. The author has also found a stronger (and much more complex) analysis that improves the bound to $O(\log^2 \Delta + \log 1/\varepsilon)$. However, even with this bound which has the desirable ε-dependency, the $O(\log^2 \Delta)$ term is undesirable.

[4] One can easily see that a precision of $O(\log \Delta)$ bits suffices.

2.2 Local Analysis: Take 2 Here, we briefly explain (a variant of) the modification of Luby's algorithm that Barenboim et al. [BEPSv3] use. The key is the following clever idea: they *manually* circumvent the problem of nodes having a lag in their degree drops, that is, they *kick out* nodes that their degree drops is lagging significantly out of the algorithm.

Formally, they divide time into phases of $\Theta(\log \log \Delta + \log 1/\varepsilon)$ rounds and require that by the end of phase k, each node has degree at most $\Delta/2^k$. At the end of each phase, each node that has a degree higher than the allowed threshold is *kicked out*. The algorithm is run for $\log \Delta$ phases. From Lemma 2.1, we can see that the probability that a node that has survived up to phase $i-1$ gets *kicked out* in phase i is at most $2^{-\Theta(\log \log \Delta + \log 1/\varepsilon)} = \frac{\varepsilon}{\log \Delta}$. Hence, the probability that a given node v gets kicked out in one of the $\log \Delta$ phases is at most ε. This means, by the end of $\Theta(\log \Delta \log \log \Delta + \log \Delta \log 1/\varepsilon)$ rounds, with probability $1-\varepsilon$, node v is not kicked out and thus has joined or has a neighbor in the MIS.

This $\Theta(\log \Delta \log \log \Delta + \log \Delta \log 1/\varepsilon)$ local complexity has an improved Δ-dependency. However, its ε-dependency is undesirable, due to the $\log \Delta$ factor. Note that this is exactly the reason that the *shattering threshold* in the result of [BEPSv3] is $O(\log^2 \Delta)$ rounds.

3 Our Algorithm and Its Local Complexity

Here we present a very simple and clean algorithm that guarantees for each node v that after $O(\log \Delta + \log 1/\varepsilon)$ rounds, with probability at least $1-\varepsilon$, node v has terminated and it knows whether it is in the (eventual) MIS or it has a neighbor in the (eventual) MIS.

The Intuition: Recall that the difficulty in locally analyzing Luby's algorithm is that the degree-dropping progresses of a node v can be delayed by those of its neighbors, which in turn can be delayed by their own neighbors, and so on. To bypass this, we try to completely disentangle the "progress" of v from that of nodes that are far away, e.g., those at distance above 2.

The intuitive base of the algorithm is as follows: There are two scenarios in which a node v has a good chance of being removed: either (1) v is trying to join the MIS and it does not have too many competing neighbors, in which case v has a chance to join the MIS, or (2) a large number of neighbors of v are trying to join the MIS each of which does not have too much competition, in which case it is likely that one of them joins the MIS and thus v gets removed. These two cases also depend only on v's 2-neighborhood. Our key idea is to create an essentially deterministic *dynamic* which has these two scenarios as its more stable points and makes

each node v spend a significant amount of time in these two scenarios, unless it has been removed already.

The Algorithm: In each round t, each node v has a *desire-level* $p_t(v)$ for joining MIS, which initially is set to $p_0(v) = 1/2$. We call the total sum of the desire-levels of neighbors of v it's *effective-degree* $d_t(v)$, i.e., $d_t(v) = \sum_{u \in N(v)} p_t(u)$. The desire-levels change over time as follows:

$$p_{t+1}(v) = \begin{cases} p_t(v)/2, & \text{if } d_t(v) \geq 2 \\ \min\{2p_t(v), 1/2\}, & \text{if } d_t(v) < 2. \end{cases}$$

The desire-levels are used as follows: In each round, node v gets *marked* with probability $p_t(v)$ and if no neighbor of v is marked, v joins the MIS and gets removed along with its neighbors[5].

Each round of the algorithm can be implemented in 2 communication rounds, one for exchanging the desire-levels and marks, and the other for informing neighbors of newly joined MIS nodes. In fact 2-bit messages suffice (for desire-levels, it's enough to report per round whether $p_t(v)$ decreased or not). In the sequel, each round means a round of the algorithm.

The Analysis: The correctness is clear as the set of nodes that join the MIS is an independent set and the algorithm terminates at a node only if the node is either in MIS or adjacent to a node in MIS. We next argue that each node v is likely to terminate quickly.

THEOREM 3.1. *For each node v, the probability that v has not made its decision within the first $\beta(\log \mathsf{deg} + \log 1/\varepsilon)$ rounds, for a large enough constant β and where deg denotes v's degree at the start of the algorithm, is at most ε. This holds even if the outcome of the coin tosses outside $N_2^+(v)$ are determined adversarially.*

Let us say that a node u is *low-degree* at time t if $d_t(u) < 2$, and *high-degree* otherwise. Considering the intuition discussed above, we define two types of *golden rounds* for a node v: (1) rounds in which $d_t(v) < 2$ and $p_t(v) = 1/2$, (2) rounds in which $d_v(t) \geq 1$ and at least $d_t(v)/10$ of it is contributed by low-degree neighbors. These are called golden rounds because, as we will see, in the first type, v has a constant chance of joining the MIS and in the second type there is a constant chance that one of those low-degree neighbors of v joins the MIS and thus v gets removed. For the sake of analysis, let us imagine that node v keeps track of the number of golden rounds of each type it has been in.

LEMMA 3.1. *By round $\beta(\log \mathsf{deg} + \log 1/\varepsilon)$, either v has joined or has a neighbor in the MIS, or at least one of its golden round counts reached $\frac{\beta}{13}(\log \mathsf{deg} + \log 1/\varepsilon)$.*

Proof. We focus only on the first $\beta(\log \mathsf{deg} + \log 1/\varepsilon)$ rounds. Let g_1 and g_2 be, respectively, the number of golden rounds for v of types 1 and 2, during this period. We assume that by the end of round $\beta(\log \mathsf{deg} + \log 1/\varepsilon)$, node v is not removed and $g_1 \leq \frac{\beta}{13}(\log \mathsf{deg} + \log 1/\varepsilon)$, and we conclude that, then it must have been the case that $g_2 > \frac{\beta}{13}(\log \mathsf{deg} + \log 1/\varepsilon)$.

Let h be the number of rounds during which $d_t(v) \geq 2$. Notice that the changes in $p_t(v)$ are governed by the condition $d_t(v) \geq 2$ and the rounds with $d_t(v) \geq 2$ are exactly the ones in which $p_t(v)$ decreases by a 2 factor. Hence, there are (exactly) h rounds in which $p_t(v)$ decreases by a 2 factor. Furthermore, the number of 2 factor increases in $p_t(v)$ can be at most equal to the number of 2 factor decreases in it, as $p_t(v)$ is capped to $1/2$. Hence, the total number of rounds in which $p_t(v)$ increases or decreases (by a 2 factor) is at most $2h$. Therefore, there are at least $\beta(\log \mathsf{deg} + \log 1/\varepsilon) - 2h$ rounds in which $p_t(v) = 1/2$. Now out of these rounds, at most h of them can be when $d_t(v) \geq 2$. Hence, $\beta(\log \mathsf{deg} + \log 1/\varepsilon) - 3h \leq g_1$. As we have assumed $g_1 \leq \frac{\beta}{13}(\log \mathsf{deg} + \log 1/\varepsilon)$, we get that $\beta(\log \mathsf{deg} + \log 1/\varepsilon) - 3h \leq \frac{\beta}{13}(\log \mathsf{deg} + \log 1/\varepsilon)$. We thus get $h \geq \frac{4\beta}{13}(\log \mathsf{deg} + \log 1/\varepsilon)$.

Let us consider the changes in the effective-degree $d_t(v)$ of v over time. If $d_t(v) \geq 1$ and this is not a golden round of type-2, then we have

$$d_{t+1}(v) \leq 2\frac{1}{10}d_v(t) + \frac{1}{2}\frac{9}{10}d_t(v) < \frac{2}{3}d_t(v).$$

There are g_2 golden rounds of type-2. Except for these, whenever $d_t(v) \geq 1$, the effective-degree $d_t(v)$ shrinks by at least a $2/3$ factor. In those exceptions, it increases by at most a 2 factor. Each of these exception rounds cancels the effect of at most 2 shrinkage rounds, as $(2/3)^2 \times 2 < 1$. Thus, ignoring the total of at most $3g_2$ rounds lost due to type-2 golden rounds and their cancellation effects, every other round with $d_t(v) \geq 2$ pushes the effective-degree down by a $2/3$ factor[6]. This cannot (continue to) happen more than $\log_{3/2} \mathsf{deg}$ times

[5] There is a version of Luby's algorithm which also uses a similar marking process. However, at each round, letting $deg(v)$ denote the number of the neighbors of v remaining at that time, Luby's sets the marking probability of each node v to be $\frac{1}{deg(v)+1}$. In our algorithm, we change the probability dynamically/flexibly over time, trying to push towards the two desirable scenarios mentioned in the intuition, and this simple dynamic is the key ingredient of the new algorithm.

[6] Notice the switch to $d_t(v) \geq 2$, instead of $d_t(v) > 1$. We need to allow a small slack here, as done by switching to threshold

as that would lead the effective degree to exit the $d_t(v) \geq 2$ region. Hence, the number of rounds in which $d_t(v) \geq 2$ is at most $\log_{3/2} \deg + 3g_2$. That is, $h \leq \log_{3/2} \deg + 3g_2$. Since $h \geq \frac{4\beta}{13}(\log \deg + \log 1/\varepsilon)$, we get $g_2 > \frac{\beta}{13}(\log \deg + \log 1/\varepsilon)$.

LEMMA 3.2. *In each type-1 golden round, with probability at least* $1/200$*, v joins the MIS. Moreover, in each type-2 golden round, with probability at least* $1/200$*, a neighbor of v joins the MIS. Hence, the probability that v has not been removed (due to joining or having a neighbor in MIS) during the first $\beta(\log \deg + \log 1/\varepsilon)$ rounds is at most ε. These statements hold even if the coin tosses outside $N_2^+(v)$ are determined adversarially.*

Proof. In each type-1 golden round, node v gets marked with probability $1/2$. The probability that no neighbor of v is marked is $\prod_{u \in N(v)}(1-p_t(u)) \geq 4^{-\sum_{u \in N(v)} p_t(v)} = 4^{-d_t(v)} > 4^{-2} = 1/16$. Hence, v joins the MIS with probability at least $1/32 > 1/200$.

Now consider a type-2 golden round. Suppose we walk over the set L of low-degree neighbors of v one by one and expose their randomness until we reach a node that is marked. We will find a marked node with probability at least

$$1 - \prod_{u \in L}(1 - p_u(t)) \geq 1 - e^{-\sum_{u \in L} p_u(t)}$$
$$\geq 1 - e^{-d_t(v)/10} \geq 1 - e^{-1/10} > 0.08.$$

When we reach the first low-degree neighbor u that is marked, the probability that no neighbor of u is marked is at least $\prod_{w \in N(u)}(1 - p_t(w)) \geq 4^{-\sum_{w \in N(u)} p_t(w)} \geq 4^{-d_t(u)} > 1/16$. Hence, with probability at least $0.08/16 = 1/200$, one of v's neighbors joins the MIS.

We now know that in each golden round, v gets removed with probability at least $1/200$, due to joining MIS or having a neighbor join the MIS. Thus, using Lemma 3.1, we get that the probability that v does not get removed is at most $(1 - 1/200)^{\frac{\beta}{13}(\log \deg + \log 1/\varepsilon)} \leq \varepsilon$.

4 Improved Global Complexity

Here, we explain how combining the algorithm of the previous section with some known techniques leads to a randomized MIS algorithm with a high probability global complexity of $O(\log \Delta) + 2^{O(\sqrt{\log \log n})}$ rounds.

As explained in Section 1.2, the starting point is to run the algorithm of the previous section for $\Theta(\log \Delta)$ rounds. Thanks to the local complexity of this base algorithm, as we will show, we reach the *shattering threshold* after $O(\log \Delta)$ rounds. The 2-hop *randomness locality* of Theorem 3.1, the fact that it only relies on the randomness bits within 2-hop neighborhood, plays a vital role in establishing this shattering phenomena. The precise statement of the shattering property achieved is given in Lemma 4.2, but we first need to establish a helping lemma:

LEMMA 4.1. *Let $c > 0$ be an arbitrary constant. For any 5-independent set of nodes S—that is, a set in which the pairwise distances are at least 5—the probability that all nodes of S remain undecided after $\Theta(c \log \Delta)$ rounds of the algorithm of Section 3 is at most $\Delta^{-c|S|}$.*

Proof. We walk over the nodes of S one by one: when considering node $v \in S$, we know from that Theorem 3.1 that the probability that v stays undecided after $\Theta(c \log \Delta)$ rounds is at most Δ^{-c}, and more importantly, this only relies on the coin tosses within distance 2 of v. Because of the 5-independence of set S, the coin tosses we rely on for different nodes of S are non-overlapping and hence, the probability that the whole set S stays undecided is at most $\Delta^{-c|S|}$.

From this lemma, we can get the following *shattering* guarantee. Since the proof is similar to that of [BEPSv3, Lemma 3.3], or those of [Bec91, Main Lemma], [ARVX12, Lemma 4.6], and [LRY15, Theorem 3], we only provide a brief sketch.

Let us fix some notation. Let G^{x^-} denote the graph where we put edges between each two G-nodes that have G-distance at most x. Also, for a given vertex set S, let $G[S]$ be the subgraph of G induced by nodes in S.

LEMMA 4.2. *Let c be a large enough constant and B be the set of nodes remaining undecided after $\Theta(c \log \Delta)$ rounds of the MIS algorithm of the previous section on a graph G. Then, with probability at least $1 - 1/n^c$, we have the following two properties:*

- *(P1) There is no (G^{4^-})-independent (G^{9^-})-connected subset $S \subseteq B$ s.t. $|S| \geq \log_\Delta n$.*
- *(P2) Each connected component of $G[B]$ has each at most $O(\log_\Delta n \cdot \Delta^4)$ nodes.*

Proof. [Proof Sketch] Let $H = G^{9^-} \setminus G^{4^-}$, i.e., the result of removing G^{4-} edges from G^{9-}. For (P1), note that the existence of any such set S would mean $H[B]$

$d_t(v) \geq 2$, in order to avoid the possible zigzag behaviors on the boundary. This is because, the above argument does not bound the number of 2-factor increases in $d_t(v)$ that start when $d_t(v) \in (1/2, 1)$ but these would lead $d_t(v)$ to go above 1. This can continue to happen even for an unlimited time if $d_t(v)$ keeps zigzagging around 1 (unless we give further arguments of the same flavor showing that this is not possible). However, for $d_t(v)$ to go/stay above 2, it takes increases that start when $d_t(v) > 1$, and the number of these is upper bounded to g_2.

contains a $(\log_\Delta n)$-node tree subgraph. There are at most $4^{\log_\Delta n}$ different $(\log_\Delta n)$-node tree topologies and for each of them, less than $n\Delta^{9\log_\Delta n}$ ways to embed it in H. For each of these trees, by Lemma 4.1, the probability that all of its nodes stay is at most $\Delta^{-2c(\log_\Delta n)}$. By a union bound over all trees, we conclude that with probability $1 - n(4\Delta^9)^{\log_\Delta n}\Delta^{-2c(\log_\Delta n)} \geq 1 - 1/n^c$, no such such set S exists. For (P2), note that if $G[B]$ has a component with more than $\Theta(\log_\Delta n \cdot \Delta^4)$ nodes, then we can find a set S violating (P1): greedily add nodes to the candidate S one-by-one, and each time discard all nodes within 4-hops of the newly added node, which are at most $O(\Delta^4)$ many.

From (P2) of Lemma 4.2, it follows that running the deterministic MIS algorithm of Panconesi and Srinivasan [PS92], which works in $2^{O(\sqrt{\log n'})}$ rounds in graphs of size n', in each of the remaining components finishes our MIS problem in $2^{O(\sqrt{\log\Delta + \log\log n})}$ rounds. However, the appearance of the $\log\Delta$ in the exponent is undesirable, as we seek a complexity of $O(\log\Delta) + 2^{O(\sqrt{\log\log n})}$. To remedy this, we use an idea similar to [BEPSv3], which tries to leverage (P1):

In a rough sense, the (P1) property of Lemma 4.2 gives that if we "contract nodes that are closer than 5-hops" (to be made precise), the left over components would have size at most $\log_\Delta n$, which would thus avoid the undesirable $\log\Delta$ term in the exponent. We next formalize this intuition.

The *finish-off* algorithm is as follows: For the sake of analysis and our discussions, we will consider each connected component C of the remaining nodes separately; the algorithm itself runs in parallel for all the components. First, run the base MIS algorithm for an extra $\Theta(\log\Delta)$ rounds. Consider each component C and let B_C be the left over nodes of C after these rounds. Now, compute a $(5,h)$-ruling set R_C of the set B_C of the remaining nodes, for an $h = \Theta(\log\log n)$, and with regards to the distances in C. Recall that a $(5,h)$-ruling set R_C means each two nodes of R_C have C-distance at least 5 while for each node in B_C, there is at least one node in R_C within its C-distance of h-hops. This $(5,h)$-ruling set R_C can be computed in $O(\log\log n)$ rounds using the algorithm[7] of Schneider, Elkin and Wattenhofer [SEW13]. See also [BEPSv3, Table 4]. Form clusters around R_C-nodes by letting each node $v \in B_C$ join the cluster of the nearest R_C-node, with regards to distances in C, and breaking ties arbitrarily. Then, contract each cluster to a new node. Thus, we get a new graph G'_C on these new nodes, where in reality, each of these nodes has radius $h = O(\log\log n)$. Thus, a communication round on G'_C can be simulated by $O(h)$ communication rounds on C.

From (P1) of Lemma 4.2, we can get that G'_C has at most $\log_\Delta n$ nodes, w.h.p. We here provide a short sketch for the reasoning of that, see [BEPSv3, Page 19, Steps 3 and 4] for a more precise description. Even though R_C might be disconnected in G^{9-}, by greedily adding more nodes of C to it, one by one, we can make it connected in G^{9-} while keeping it 5-independent. We note that this is done only for the analysis. Since by (P1) of Lemma 4.2, the end result should have size at most $\log_\Delta n$, with high probability, we conclude G'_C has at most $\log_\Delta n$ nodes, with high probability.

We can now compute an MIS of C, via almost the standard deterministic way of using network decompositions. We run the network decomposition algorithm of Panconesi and Srinivasan [PS92] on G'_C. This takes $2^{O(\sqrt{\log\log_\Delta n})}$ rounds and gives G'_C-clusters of radius at most $2^{O(\sqrt{\log\log_\Delta n})}$, colored with $2^{O(\sqrt{\log\log_\Delta n})}$ colors such that adjacent clusters do not have the same color. We will walk over the colors one by one and compute the MIS of the clusters of that color, given the solutions of the previous colors. Each time, we can (mentally) expand each of these G'_C clusters to all the B_C-nodes of the related cluster, which means these B_C-clusters have radius at most $\log\log n \cdot 2^{O(\sqrt{\log\log_\Delta n})}$ in C. While solving the problem of color-j clusters, we make a node in each of these clusters gather the whole topology of its cluster and also the adjacent MIS nodes of the previous colors. Then, this cluster-center solves the MIS problem locally, and reports it back. Since each cluster has radius $\log\log n \cdot 2^{O(\sqrt{\log\log_\Delta n})}$, this takes $\log\log n \cdot 2^{O(\sqrt{\log\log_\Delta n})}$ rounds per color. Thus, over all the colors, the complexity becomes $2^{O(\sqrt{\log\log_\Delta n})} \cdot \log\log n \cdot 2^{O(\sqrt{\log\log_\Delta n})} = 2^{O(\sqrt{\log\log n})}$ rounds. Including the $O(\log\log n)$ ruling-set computation rounds and the $O(\log\Delta)$ pre-shattering rounds, this gives the promised global complexity of $O(\log\Delta) + 2^{O(\sqrt{\log\log n})}$, hence proving Theorem 1.2.

5 Open Questions

The gap between the upper and lower bounds, which now shows up when $\log\Delta = \omega(\sqrt{\log n})$, is perhaps the most interesting open question. We saw in (C2) of Section 1.3 that if the lower-bound is the one that should be improved, we need to go away from "tree local-views" topologies. Another longstanding open problem

[7] This is different than what Barenboim et al. did. They could afford to use the more standard ruling set algorithm, particularly computing a $(5, 32\log\Delta + O(1))$-ruling set for their purposes, because the fact that this $32\log\Delta$ ends up multiplying the complexity of their finish-off phase did not change (the asymptotics of) their overall complexity.

is to find a $\text{poly}(\log n)$ deterministic MIS algorithm. Combined with our results, that can potentially get us to an $O(\log \Delta) + \text{poly}(\log \log n)$ randomized algorithm.

Acknowledgment: I would like to thank Eli Gafni, Bernhard Haeupler, Stephan Holzer, Fabian Kuhn, Nancy Lynch, and Seth Pettie for valuable discussions. I am also grateful to Fabian Kuhn, Nancy Lynch, and the reviewers of SODA'16 for carefully reading the paper and many helpful comments. The point (C2) in Section 1.3 was brought to my attention by Fabian Kuhn. The idea of highlighting the *local complexity* as an alternative complexity measure for local problems is rooted in conversations with Stephan Holzer and Nancy Lynch, and in Eli Gafni's insistence[8] that *"the (true) complexity of a local problem should not depend on n"*.

References

[ABI86] Noga Alon, László Babai, and Alon Itai. A fast and simple randomized parallel algorithm for the maximal independent set problem. *Journal of algorithms*, 7(4):567–583, 1986.

[ARVX12] Noga Alon, Ronitt Rubinfeld, Shai Vardi, and Ning Xie. Space-efficient local computation algorithms. In *Pro. of ACM-SIAM Symp. on Disc. Alg. (SODA)*, pages 1132–1139, 2012.

[BE10] Leonid Barenboim and Michael Elkin. Sublogarithmic distributed mis algorithm for sparse graphs using nash-williams decomposition. *Distributed Computing*, 22(5-6):363–379, 2010.

[Bec91] József Beck. An algorithmic approach to the Lovász local lemma. I. *Random Structures & Algorithms*, 2(4):343–365, 1991.

[BEPSv3] Leonid Barenboim, Michael Elkin, Seth Pettie, and Johannes Schneider. The locality of distributed symmetry breaking. In *Foundations of Computer Science (FOCS) 2012*, pages 321–330. IEEE, 2012, also coRR abs/1202.1983v3.

[BKP14] Tushar Bisht, Kishore Kothapalli, and Sriram Pemmaraju. Brief announcement: Super-fast t-ruling sets. In *the Proc. of the Int'l Symp. on Princ. of Dist. Comp. (PODC)*, pages 379–381. ACM, 2014.

[CL01] Fan Chung and Linyuan Lu. The diameter of random sparse graphs. *Advances in Applied Math*, 26(4):257–279, 2001.

[Coo83] Stephen A Cook. An overview of computational complexity. *Comm. of the ACM*, 26(6):400–408, 1983.

[CPS14] Kai-Min Chung, Seth Pettie, and Hsin-Hao Su. Distributed algorithms for the lovász local lemma and graph coloring. In *the Proc. of the Int'l Symp. on Princ. of Dist. Comp. (PODC)*, pages 134–143, 2014.

[8] This is a paraphrased version of his comment during a lecture on Linial's $\Omega(\log^* n)$ lower bound, in the Fall 2014 Distributed Graph Algorithms (DGA) course at MIT.

[HPP+15] James Hegeman, Gopal Pandurangan, Sriram Pemmaraju, Viveck Sardeshmuck, and Michele Scquizatto. Toward optimal bounds in the congested clique: Graph connectivity and MST. In *the Proc. of the Int'l Symp. on Princ. of Dist. Comp. (PODC)*, page to appear, 2015.

[KMWv1] Fabian Kuhn, Thomas Moscibroda, and Roger Wattenhofer. What cannot be computed locally! In *the Proc. of the Int'l Symp. on Princ. of Dist. Comp. (PODC)*, pages 300–309. ACM, 2004, also coRR abs/1011.5470v1.

[Kuh15] Fabian Kuhn. Personal Communication, 2015.

[KW84] Richard M Karp and Avi Wigderson. A fast parallel algorithm for the maximal independent set problem. In *Proc. of the Symp. on Theory of Comp. (STOC)*, pages 266–272. ACM, 1984.

[Len13] Christoph Lenzen. Optimal deterministic routing and sorting on the congested clique. In *the Proc. of the Int'l Symp. on Princ. of Dist. Comp. (PODC)*, pages 42–50, 2013.

[LRY15] Reut Levi, Ronitt Rubinfeld, and Anak Yodpinyanee. Local computation algorithms for graphs of non-constant degrees. *preprint arXiv:1502.04022*, 2015.

[Lub85] Michael Luby. A simple parallel algorithm for the maximal independent set problem. In *Proc. of the Symp. on Theory of Comp. (STOC)*, pages 1–10, 1985.

[LW11] Christoph Lenzen and Roger Wattenhofer. MIS on Trees. In *the Proc. of the Int'l Symp. on Princ. of Dist. Comp. (PODC)*, pages 41–48. ACM, 2011.

[Lyn96] Nancy A Lynch. *Distributed algorithms*. Morgan Kaufmann, 1996.

[MR10] Rajeev Motwani and Prabhakar Raghavan. *Randomized algorithms*. Chapman & Hall/CRC, 2010.

[Pel00] David Peleg. *Distributed Computing: A Locality-sensitive Approach*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.

[PR07] Michal Parnas and Dana Ron. Approximating the minimum vertex cover in sublinear time and a connection to distributed algorithms. *Theoretical Computer Science*, 381(1):183–196, 2007.

[PS92] Alessandro Panconesi and Aravind Srinivasan. Improved distributed algorithms for coloring and network decomposition problems. In *Proc. of the Symp. on Theory of Comp. (STOC)*, pages 581–592. ACM, 1992.

[RTVX11] Ronitt Rubinfeld, Gil Tamir, Shai Vardi, and Ning Xie. Fast local computation algorithms. In *Symp. on Innov. in Theo. of Comp. Sci.*, pages 223–238, 2011.

[SEW13] Johannes Schneider, Michael Elkin, and Roger Wattenhofer. Symmetry breaking depending on the chromatic number or the neighborhood growth. *Theoretical Computer Science*, 509:40–50, 2013.

[Val82] Leslie G Valiant. Parallel computation. In *Proc. 7th IBM Symp. on Math. Found. of Comp. Sci.*, 1982.

[Win04] Aaron Windsor. A simple proof that finding a maximal independent set in a graph is in NC. *Information processing letters*, 92(4):185–187, 2004.