On the Randomized Locality of Matching Problems in Regular Graphs

Seri Khoury
INSAIT, Sofia University "St. Kliment Ohridski", Bulgaria

Manish Purohit
Google Research, Mountain View, CA, USA

Aaron Schild
Google Research, San Francisco, CA, USA

Joshua R. Wang
Google Research, San Francisco, CA, USA

Abstract

The main goal in distributed symmetry-breaking is to understand the locality of problems: the radius of the neighborhood that a node must explore to determine its part of a global solution. In this work, we study the locality of matching problems in the family of regular graphs, which is one of the main benchmarks for establishing lower bounds on the locality of symmetry-breaking problems, as well as for obtaining classification results. Our main results are summarized as follows:

1. **Approximate matching:** We develop randomized algorithms to show that $(1+\epsilon)$-approximate matching in regular graphs is truly local, i.e., the locality depends only on ϵ and is independent of all other graph parameters. Furthermore, as long as the degree Δ is not very small (namely, as long as $\Delta \geq \text{poly}(1/\epsilon)$), this dependence is only logarithmic in $1/\epsilon$. This stands in sharp contrast to maximal matching in regular graphs which requires some dependence on the number of nodes n or the degree Δ.
2. **Maximal matching:** Our techniques further allow us to establish a strong separation between the node-averaged complexity and worst-case complexity of maximal matching in regular graphs, by showing that the former is only $O(1)$.

Central to our main technical contribution is a novel martingale-based analysis for the ≈ 40-year-old algorithm by Luby. In particular, our analysis shows that applying one round of Luby's algorithm on the line graph of a Δ-regular graph results in an almost $\Delta/2$-regular graph.

2012 ACM Subject Classification Theory of computation → Distributed algorithms; Mathematics of computing → Probabilistic algorithms; Theory of computation → Graph algorithms analysis

Keywords and phrases regular graphs, maximum matching, augmenting paths, distributed algorithms, Luby's algorithm, martingales

Digital Object Identifier 10.4230/LIPIcs.DISC.2025.40

1 Introduction

Matching problems, such as maximal or maximum matching, have garnered significant attention across several computational models, including the classical sequential model [57, 64, 77, 88, 89]; dynamic networks [10, 33, 34, 59, 62, 85]; streaming algorithms [12, 13, 41, 50, 78, 82]; online algorithms [40, 61, 67, 68, 79]; and distributed computing [1, 2, 27, 32, 38, 45, 46, 48, 55, 56, 63, 66, 74, 75].

In the classical LOCAL model of distributed computing, a network of n nodes collaborates to solve a graph problem through a series of synchronized communication rounds. In each round, a node can send an unbounded-size message to each of its neighbors. The primary goal is to minimize the number of communication rounds required to solve the problem; a measure known as the problem's *locality*. Since each node can only interact with the

nodes in its r-hop neighborhood in r rounds, the problem's locality reflects the radius of the neighborhood that each node must explore to determine its part in the global solution (e.g., whether it is matched and if so to which neighbor).

Typically, the locality of matching problems depends on global graph parameters, such as the number of nodes n or the maximum degree Δ. For instance, a simple algorithm by Israeli and Itai [65] finds a maximal matching in $O(\log n)$ rounds in the LOCAL model, which constitutes a 2-approximate maximum matching in unweighted graphs (or approximate matching, for short). Later, Barenboim et al. [29] showed that this logarithmic dependence on n can be substituted with a logarithmic dependence on Δ by providing an $O(\log \Delta + \text{poly} \log \log n)$-round algorithm. This upper bound also applies to finding a $(1+\epsilon)$-approximate matching (while incurring a multiplicative $\text{poly}(1/\epsilon)$ factor in the number of rounds), as shown by Harris [63].

All of the above results are for randomized algorithms that succeed with high probability.[1] The landscape changes for other computational assumptions. For instance, if the matching is only required to be large in expectation, then recent work has shown that it is possible to shave a $\log \log \Delta$ factor from the round complexity [27, 28, 50, 52]. For deterministic algorithms, however, the primary question is whether the state-of-the-art randomized bounds can be matched.[2] For a more detailed discussion on deterministic algorithms, we refer the reader to the excellent surveys by Suomela [87] and Rozhon [84].

Whether there exist $o(\log n)$-round algorithms that succeed with high probability for maximal or $(1+\epsilon)$-approximate matching in general graphs remains one of the main open questions in the field. On the other hand, the best known lower bound as a function of n is $\Omega(\sqrt{\log n / \log \log n})$ rounds, which was shown by Kuhn, et al. [72], and it applies all the way up to $\text{poly}(\log n)$-approximation.

Regular Graphs. A graph is Δ-regular if all its nodes have degree Δ. The family of regular graphs has received much attention as a natural benchmark for studying the complexity of various fundamental problems (e.g., [4–9, 14–16, 42, 58, 70, 86, 90]). In the LOCAL model, regular graphs serve as the standard benchmark for implementing the *Round Elimination* technique for proving lower bounds [17,21–25,35,36,39], and also for obtaining classifications of locality results (e.g., in paths and cycles [18,37], grids [37,43,60,81], and regular trees [19,20]).[3]

The locality of some problems in regular graphs depends on graph parameters such as the number of nodes n or the degree Δ, while for others, it is independent of any such parameter. Interestingly, as we discuss next, maximal matching falls into the former category, whereas $(1+\epsilon)$-approximate matching is in the latter.

For maximal matching, the approximately 40-year-old $O(\log n)$ upper bound by Israeli and Itai [65] remains the best known even for regular graphs. Moreover, Balliu et al. [21] showed via the Round Elimination technique that maximal matching in regular graphs requires $\Omega(\min\{\Delta, \log \log n / \log \log \log n\})$ rounds for randomized algorithms and $\Omega(\min\{\Delta, \log n / \log \log n\})$ rounds for deterministic ones. Even for low-degree regular graphs (e.g. cycles), any randomized algorithm for maximal matching still requires $\Omega(\log^* n)$ rounds [73, 80].

---

[1] Throughout the paper, we say that an algorithm succeeds with high probability if it succeeds with probability $1 - 1/n^c$ for some arbitrarily large constant $c > 0$.

[2] The current state-of-the-art deterministic algorithms take $\tilde{O}(\log^{5/3} n)$ rounds for maximal matching [54], and $\tilde{O}(\log^{4/3} n \cdot \text{poly}(1/\epsilon))$ rounds for $(1+\epsilon)$-approximate matching [50, 53], where $\tilde{O}(x)$ hides $\text{poly} \log x$ factors.

[3] For more on classification of locality results, as well as ones in general graphs, we refer the reader to the survey by Rozhon [84].

In stark contrast, the story for $(1+\epsilon)$ approximate matching in regular graphs is far less clear. The hard instances for approximate matching developed by Kuhn et al. [72] are very far from being regular. In fact, these instances are also hard for finding an approximate *fractional matching*. On regular graphs, however, fractional matching admits a trivial zero-round solution by simply setting the fractional value for each edge to be $1/\Delta$. This fractional matching can be rounded to an $O(1)$-approximate integral matching in one round by using a simple sampling technique [52].

This raises the question: where does $(1+\epsilon)$-approximate matching fall on the spectrum of locality in regular graphs? Is it closer to approximate fractional matching, or does it require some dependence on n or Δ similar to maximal matching?[4]

1.1 Our Results

Approximate Matching. In this work, our first result is a simple algorithm that finds a $(1+\epsilon)$-approximate matching in regular graphs without any dependence on graph parameters.[5] For constant values of ϵ, the algorithm also works in the more restricted CONGEST model, where the size of the messages is bounded by $O(\log n)$ bits.

▶ **Theorem 1.** *Let n be a positive integer, and let $\epsilon \in (n^{-1/20}, 1/2)$ be an accuracy parameter. There is an $O(\epsilon^{-5}\log(1/\epsilon))$-round LOCAL algorithm that finds a $(1+\epsilon)$-approximate maximum matching in n-node regular graphs, with high probability. The algorithm works in the CONGEST model for constant values of ϵ.*

While Theorem 1 advances our understanding of $(1+\epsilon)$-approximate matching in regular graphs, the polynomial dependence on $1/\epsilon$ is far from desirable. Comparing with the $\tilde{O}(\epsilon^{-3}\log\Delta + \text{polylog}(1/\epsilon, \log\log n))$ round algorithm of Harris [63], the algorithm from Theorem 1 is better only in the regime $\epsilon > 1/\sqrt{\log\Delta}$. In our next result, we present an exponentially faster algorithm as long as $\epsilon > 1/\Delta^{c'}$ for some constant $c' > 0$. In other words, we present an exponentially faster $(1+\epsilon)$-approximation algorithm for graphs that are not extremely sparse, i.e., when $\Delta > \text{poly}(1/\epsilon)$. We note that the constant c in Theorem 2 has not been optimized and can be improved substantially with a more careful analysis.

▶ **Theorem 2** (Main Result I). *Let $c = 10^5$ and let $\epsilon \in (0,1)$ be an accuracy parameter. For Δ-regular graphs with $\Delta > (1/\epsilon)^c$, there is an $O(\log(1/\epsilon))$-round CONGEST algorithm that finds a $(1+\epsilon)$-approximate maximum matching with high probability.*

One may wonder whether the restriction in the theorem about the graph being dense enough is a mere technicality. The following lower bound presented in Theorem 3 shows that this is not the case, allowing us to establish a separation between dense and sparse regular graphs in which dense ones are easier. Note that, for the same problem, the *opposite* separation holds in general graphs (due to the $\Omega(\min\{\log\Delta/\log\log\Delta, \sqrt{\log n/\log\log n}\})$ lower bound by [72], and the upper bounds by [63]).

[4] Note that the amplification technique of [50] cannot be used to amplify the constant-approximation factor to $(1+\epsilon)$ in regular graphs while using $\text{poly}(1/\epsilon)$ phases of amplification. This is because the technique is designed for general graphs, and applying it to a regular graph can result in a non-regular graph after the first step. Consequently, the amplification algorithm would need to use an algorithm for constant-approximate matching in *general graphs*, which requires some dependence on n or Δ.

[5] All our upper bounds apply also to almost regular graphs, where the degrees of all the nodes are within a $(1\pm o(1))$-multiplicative factor from each other. Moreover, the case where $\epsilon \le n^{-1/20}$ can be handled in $\text{poly}(1/\epsilon)$ rounds by transmitting the entire graph to all nodes.

▶ **Theorem 3** (Informal version of hardness result; see full version for details). *For any degree $\Delta \geq 2$ and error $\epsilon = O(\Delta^{-1})$, any LOCAL algorithm that computes a $(1+\epsilon)$-approximate maximum matching in bipartite Δ-regular graphs with at least $n \geq \Omega(\Delta^{-1}\epsilon^{-1})$ nodes requires $\Omega(\Delta^{-1}\epsilon^{-1})$ rounds.*

One intuition behind this separation is that in Δ-regular graphs, the number of nearly optimal solutions to maximum matching scales with Δ (see for instance [11]). Intuitively, this abundance of nearly optimal solutions makes it easier to rapidly identify one using randomness.

At the heart of the proof of our first main result (Theorem 2) is a novel martingales-based analysis of the classical algorithm by Luby [76]. In particular, we use this analysis to prove our *Regular-Graph Preservation Lemma* (Lemma 6), which shows that running a single round of Luby's algorithm on the line graph of a sufficiently dense Δ-regular graph and removing the matched nodes together with their incident edges yields an almost $\approx \Delta/2$-regular graph. Interestingly, this analysis wilil also allow us to attain a good node-averaged complexity for maximal matching.

Node-Averaged Complexity of Maximal Matching. When performing a distributed computation, the algorithm may arrive at some parts of its final output before the rest. That is, some nodes know their local result before the algorithm fully terminates on the last few stragglers. The notion of node-averaged complexity [26, 30, 44, 47] captures this nuance by reasoning about the average over the times at which the nodes finish their computation and commit to their outputs (we formally define the node-averaged complexity in Section 1.2). Node-averaged complexity captures important applications such as optimizing the total energy consumption of the network (see, for instance, [44] and references within).

In sharp contrast to the worst-case complexity of maximal matching in regular graphs, our martingale-based analysis of Luby's algorithm yields a *constant* node-averaged complexity. This also contrasts with a lower bound of Balliu et al. [26] who showed that in general graphs, the node-averaged complexity of maximal matching is $\Omega\left(\min\left\{\frac{\log\Delta}{\log\log\Delta}, \sqrt{\frac{\log n}{\log\log n}}\right\}\right)$.[6]

▶ **Theorem 4** (Main Result II). *The node-averaged complexity of finding a maximal matching in Δ-regular graphs is $O(1)$.*

Theorem 4 implies that, with respect to node-average complexity, maximal matching is easier in regular graphs than in general ones. On the other hand, as discussed earlier, for worst-case complexity, the $O(\log n)$ bound by [65] is still the best known for both regular and general graphs. Whether maximal matching is any easier in regular graphs with respect to worst-case complexity remains an interesting open question.

Outline of the paper. In Section 1.2 we provide some basic definitions and notation. In Section 2 we provide a brief technical overview of our results. Section 3 contains a proof sketch of Theorem 2. The proof of Theorem 4 is deferred to the full version of the paper. The result is obtained by applying $O(\log\log\Delta)$ rounds of Luby, which our results show

[6] We note that the corresponding edge-averaged complexity of maximal matching is known to be $O(1)$, even in general graphs, due to the classical algorithms of [3, 76] (when applied on the line graph) that delete a constant-fraction of the edges in each round [26]. However, when an edge arrives at its part of the output (i.e., whether it is matched or not), it does not necessarily help an incident node to arrive at its part of the output, as several edges are not matched. This nuance is fundamental due to the lower bound by [26] discussed above for the node-averaged complexity of maximal matching in general graphs.

matches all but a $1/(\log \Delta)$-fraction of the nodes with $O(1)$ node-averaged complexity, and applying [51] to match the remaining nodes. The details of the proofs of Theorems 4 and 2, along with all discussion of Theorems 1 and 3, are deferred to the full version.

1.2 Model and Basic Definitions

The LOCAL and CONGEST model. In this work we are interested in the LOCAL and CONGEST models of distributed computing [73, 83]. In both models, there is a synchronized communication network of n computationally unbounded nodes that can communicate via communication rounds; this network both defines the communication pattern and serves as the input graph we want to solve our problem on. In each round, each node can send an unbounded-size (in the LOCAL model) or $O(\log n)$-bit (in the CONGEST model) message to each of its neighbors. The goal is to perform a task (e.g., find a large matching) while minimizing the number of communication rounds. At the end of all communication rounds, each node needs to know which of its neighbors it is matched with, if any.

Basic Graph Notations. For a graph G, we denote by $V(G)$ the set of nodes in G and by $E(G)$ the set of edges. Given a node $u \in V(G)$, we denote by $N_G(u)$ the set of neighbors of u in G, and by $N_G^d(u)$ the set of nodes at distance exactly d from u. When G is clear from the context, we omit the letter G from the notation and use V, E and $N(u)$ for brevity. Throughout the paper, n denotes the number of nodes in the graph. In Δ-regular graphs, all the nodes have the same degree Δ. In this work, we are interested in unweighted and undirected graphs.

Maximum and maximal matching. A matching $\mathcal{M}$ in a graph G is a set of edges in $E(G)$, where no two edges in $\mathcal{M}$ share a node. A maximum matching is a matching of maximum possible size. A $(1+\epsilon)$-approximate matching in G is a matching $\mathcal{M}$ satisfying $OPT \leq (1+\epsilon)|\mathcal{M}|$, where OPT is the size of a maximum matching. A maximal matching is a matching that is not a strict subset of any other matching.

Node-averaged Complexity ([26]). The node-averaged complexity of an algorithm A, $\mathrm{AVG}_V(A)$, is defined as follows.

$$\mathrm{AVG}_V(A) := \max_{G \in \mathcal{G}} \frac{1}{|V(G)|} \cdot \sum_{v \in V(G)} \mathbb{E}\left[T_v^G(A)\right]$$

where $T_v^G(A)$ is the time it takes for v to reach its part of the output when running A. The node-average complexity of a graph problem is defined as the node-average complexity of the best algorithm A^* that minimizes $\mathrm{AVG}_V(A^*)$.

2 Technical Overview

2.1 Warmup: A $\mathrm{poly}(1/\epsilon)$-Round Algorithm for Regular Graphs

To prove Theorem 1, we use a two stage algorithm. We begin with a Δ-regular graph on n vertices with target error parameter $\epsilon > n^{-1/20}$. The first stage (sampling) involves uniformly and independently sampling edges from the graph with the goal of reducing the degree from Δ to $\mathrm{poly}(1/\epsilon)$, plus some post-processing. After this stage, we have an (irregular) graph on at most n vertices with degree at most $d = \mathrm{poly}(1/\epsilon)$ and have used up a constant fraction

of our error parameter ϵ (i.e. this restricted graph still has an almost-perfect matching). Our second stage (matching) involves finding a matching in this restricted graph, and its runtime only depends on the error parameter ϵ and the max degree d.

For the sampling stage, uniformly sampling to degree approximately $\epsilon^{-2} \log n$ would result in a graph that retains a near-perfect matching with high probability (via a Chernoff bound plus a union bound), which then when combined with Harris' algorithm [63] can already give us an $O(\text{poly}(1/\epsilon) \log \log n)$-round algorithm. However, to avoid the dependence on n, we need to find a way to reduce this degree even further. Instead, we sample down to degree $\Theta(\epsilon^{-4})$. The resulting subgraph can be very irregular, but we manage to tease out enough structure to make our argument go through. In particular, some small fraction of vertices may have degree exceeding our target $\Theta(\epsilon^{-4})$ by more than a factor two. We use Chernoff with bounded dependence and the matching polytope to argue that stripping out these problematic high-degree vertices still leaves an almost-perfect matching.

For the matching stage, we find a matching in the constructed low-degree subgraph from the sampling stage. The state-of-the-art algorithms of Harris [63] fit our task, but they have a small runtime dependence on n. Instead, we combine the hypergraph framework from [63] (which was also used in [27, 49, 50]) with some ideas from [52], as follows. In $\text{poly}(1/\epsilon)$ phases, we increase the size of the matching in each phase by $\text{poly}(\epsilon) \cdot n$ edges. The algorithm for each phase finds a large set of disjoint augmenting paths. This is done by first constructing a hypergraph H with the same set of nodes as in the low-degree subgraph, where each hyperedge corresponds to a $1/\epsilon$-length augmenting paths. Then, we find an $O(1/\epsilon)$-approximate fractional matching in the hypergraph by using the algorithms of [31,63,71], and we round this fractional matching by sampling each hyperedge with probability proportional to its fractional value. By using a similar McDiarmid-type argument as in [52], we can show that this rounding produces an integral $O(1/\epsilon)$-approximate matching in the hypergraph H with high probability. Furthermore, observe that the maximum degree of a node in the hypergraph H is $\exp(\text{poly}(1/\epsilon))$. Therefore, the algorithms of [31, 63, 71] for finding a fractional matching in this hypergraph take $O(\text{poly}(1/\epsilon))$ rounds, as desired. We discuss this result formally in the full version.

2.2 Exponentially Faster Algorithm

In this section, we give a brief technical overview for our main result. On bipartite graphs, our algorithm for proving Theorem 2 simply runs $O(\log(1/\epsilon))$ rounds of Luby's algorithm [76] that finds a large matching in each round. On general graphs, we first run a color coding step to find a large bipartite almost regular subgraph. While the algorithm is very simple, the main challenge is its analysis. In this work, we provide a new martingale-based analysis for Luby's algorithm.

Since our analysis relies on martingale concentration inequalities, it is more convenient to work with a sequential view of Luby's algorithm. Our key lemma (Lemma 6) shows that after applying one round of Luby's algorithm (and removing the matched nodes along with incident edges), the remaining graph remains almost regular, i.e., almost all nodes have very similar degrees. Even with the sequential view, classical martingale concentration inequalities are not sufficient to provide the high probability bounds that we require. We use two techniques, a *shifted martingale trick* and *scaled martingale trick*, in order to provide the requisite bounds. Roughly speaking, we show that the number of matched neighbors of a node behaves similarly to a martingale. Finally, we show that a constant fraction of the nodes are matched in each iteration of Luby's algorithm when the graph is almost regular. Combined with our key lemma, this implies that after $O(\log(1/\epsilon))$ rounds at most ϵ fraction of nodes remain unmatched. We now present a deeper overview of each of the above components in the subsections below.

2.2.1 Sequential view of Luby's algorithm

In the traditional distributed implementation of one round of Luby's algorithm, each edge f picks a uniformly random number r_f and some edge e is chosen into the matching if and only if $r_e < r_{e'}$ for all neighboring edges e'.

We consider the following sequential view - the edges of the graph arrive sequentially in a uniformly random order and an edge e is chosen into the matching if and only if it arrives before any of its neighboring edges.

Assuming that there are no collisions (i.e. each edge chooses a different random number from its neighbors), it can be readily seen that the two algorithms above produce exactly the same distribution over matchings. For CONGEST algorithms, we restrict the range of the random numbers to be integers in $\{1, 2, \ldots, M\}$ for some polynomially large M. In this case, [69] showed that the two algorithms are identical up to a vanishingly small failure probability.

2.2.2 Martingale Techniques

Our analysis of a single round of Luby's algorithm relies on analyzing certain associated martingales and using martingale concentration inequalities.

Shifted Martingale. Consider a collection $X_1, X_2, \ldots, X_t$ of boolean random variables and let $\mathbb{E}[X_i \mid X_1, \ldots, X_{i-1}] = p_i$. We are interested in obtaining high probability bounds on the sum $S_t = \sum_{i=1}^{t} X_i$. For example, consider X_i to be the indicator random variable for the event that the ith edge is chosen into the matching. Now clearly, the random variables $\{X_i\}$ are not independent, so we cannot use standard Chernoff bounds. If we let $S_i = \sum_{j=1}^{i} X_j$, then one could hope to use martingale inequalities to get a concentration result for S_i. The challenge here is that the sequence $S_1, \ldots, S_t$ is not necessarily a martingale. Nevertheless, when the p_i are bounded, then we can still utilize martingale concentration inequalities to show that S_t does not deviate too much from its mean by considering the following *shifted random variables*:

$$Y_i = S_i - \mathbb{E}[S_i \mid Y_0, \ldots, Y_{i-1}] + Y_{i-1}$$

Since the sequence $Y_1, \ldots, Y_t$ is a martingale and further has bounded variance, we can use bounded variance martingale concentration bounds on Y_t to give good concentration on S_t as well. This shifting idea is inspired by [69], in which it is used in the context of approximate-maximum independent set. In this work, we generalize this shifting idea to broader scenarios, which we discuss in the full version.

Scaled Martingale. In some parts of our analysis, the shifted martingale trick doesn't suffice for our purposes. Consider a sequence of random variables $S_1, \ldots, S_t$ such that $\mathbb{E}[S_i \mid S_1, \ldots, S_{i-1}] = (1-p)S_{i-1}$ for some fixed $0 < p < 1$. In this scenario, we expect the difference $S_i - S_{i-1}$ to decrease as i increases. It is challenging for the shifted martingale trick to exploit such dynamics. The main reason is that in order for the shifted martingale to give a concentration result, we need the expected value of $S_i - S_{i-1}$ to be bounded by some fixed number, which wouldn't exploit the property that the difference is decreasing over time. To get a concentration result in such scenarios, we use another trick which we refer to as the *scaling trick*. We can obtain a concentration bound for S_t by considering the

following scaled random variables:

$$F_i = \frac{S_i}{(1-p)^{i-1}}$$

Observe that F_i is a martingale. This is because $\mathbb{E}[F_i \mid F_1, \cdots, F_{i-1}] = \frac{1}{(1-p)^{i-1}}\mathbb{E}[S_i \mid F_1, \cdots, F_{i-1}] = \frac{1}{(1-p)^{i-1}}\mathbb{E}[S_i \mid S_1, \cdots, S_{i-1}] = \frac{S_{i-1}}{(1-p)^{i-2}} = F_{i-1}$. Therefore, we can get a concentration result for S_t by getting a concentration result for F_t. The main intuition behind the scaling trick is that it exploits the decreasing difference between S_i and S_{i-1} over time. This is exactly the reason for dividing S_i by $(1-p)^i$. For instance, since $F_1 = S_1$, if we get that F_t doesn't deviate too far from F_1, it would imply that $S_i = (1-p)^i F_i \approx (1-p)^i F_1 = (1-p)^i S_1$, which is exactly where we're expecting S_i to be at step i.

2.2.3 Local Regular-Graph Preservation Lemma

We analyze a single round of Luby's algorithm using the above martingale based techniques. First, we show a lemma with the following local guarantee.

▶ **Lemma 5** (Local Regular-Graph Preservation Lemma - Informal)**.** *Let G be a bipartite Δ-regular graph and suppose we run one round of Luby's algorithm on G and let u be an arbitrary node in G. Then, with probability at least $1-\exp(-\operatorname{poly}(\Delta))$, $\Delta/2 \pm o(\Delta)$ neighbors of u get matched.*

Recall that $N(u)$ is the set of neighbors of node u and $N^2(u)$ is the set of nodes at distance exactly 2 from u. Let A_u be the set of edges between $N(u)$ and $N^2(u)$. To prove Lemma 5, we show that roughly $\Delta/2$ edges from A_u are chosen into the matching with probability at least $1-\exp(-\operatorname{poly}(\Delta))$. While the claim holds trivially in expectation, it is challenging to obtain high probability bounds due to the dependencies between u's neighbors.

To simplify exposition, let E_u denote the set of edges that are at most 3 hops away from u, i.e. $E_u = E \cap \big(\{u\} \times N(u)) \cup (N(u) \times N^2(u)) \cup (N^2(u) \times N^3(u))\big)$. Clearly edges not in E_u do not affect any of the edges in A_u and hence can be ignored, so we restrict the analysis to assume that only edges from E_u arrive in the sequential view of Luby's algorithm.

Let $\mathcal{M}_u \subset A_u$ be the set of matching edges chosen from A_u and our goal is to get high probability upper and lower bounds on $|\mathcal{M}_u|$. Let $X_i \in \{0,1\}$ be an indicator random variable for the event that the ith arriving edge belongs to $\mathcal{M}_u$. Let $q_i = \mathbb{E}[X_i \mid X_1, \ldots, X_{i-1}]$ be the probability that the ith edge is from A_u and none of its neighboring edges have already arrived. One could try now to utilize the shifted martingale trick described above to obtain concentration bounds on $|\mathcal{M}_u| = \sum_i X_i$. However, the main challenge here is that q_i itself is a random variable. Our goal is to analyze q_i using martingales analysis. Roughly speaking, we use the scaled martingale trick discussed above to get concentration results for q_i for all i, which enables us to apply the shifted martingale trick to get the desired bounds on $|\mathcal{M}_u|$.

Analyzing q_i. To analyze q_i, we need to understand how many edges *survive* after the first $i-1$ edges have already arrived. Intuitively, an edge is still surviving in iteration i if neither it nor any of its neighbors was not sampled in the first $i-1$ iterations. Let E_i be the set of surviving edges in A_u at the beginning of the ith iteration. Then we have, $q_i = \frac{|E_i|}{|E_u|-(i-1)} = \frac{|E_i|}{\Delta(|N^2(u)|+1)-(i-1)}$ since a sampled surviving edge is always added to the matching.

The final key property towards the proof. To get high probability bounds on $|E_i|$, we first prove that $\mathbb{E}[|E_i| \mid E_{i-1}] \approx (1-2/k)|E_{i-1}|$. This is exactly the setting where the scaled martingale trick can help us to get a concentration result for $|E_i|$, which paves the way for proving Lemma 5.

2.2.4 Regular-Graph Preservation Lemma

We use Lemma 5 to show that applying a single round of Luby's algorithm on a regular graph and removing the matched nodes along with incident edges results in a graph that is still almost regular.

▶ **Lemma 6** (Regular-Graph Preservation Lemma - Informal). *Let G be a bipartite Δ-regular graph and let G' be the graph obtained by running one round of Luby's algorithm on G and deleting matched nodes and their incident edges. Then all but $o(1)$ fraction of nodes in G' have their degree in the range $\Delta/2 \pm o(\Delta)$ with high probability.*

When Δ is large enough ($\Delta \geq \operatorname{poly}\log n$), Lemma 5 followed by a union bound suffices to show that all nodes have their degree in the range $\Delta/2 \pm o(\Delta)$ with high probability. On the other hand, when Δ is small, then by Lemma 5, the expected number of nodes that do *not* have the requisite degree is only $n \cdot \exp(-\operatorname{poly}(\Delta))$. Further, the degree of a node after a round of Luby's algorithm only depends on $O(\operatorname{poly}(\Delta))$ other nodes. So, we can use a Chernoff-Hoeffding with bounded dependence inequality to argue that all but a $\exp(-\operatorname{poly}(\Delta))$-fraction of nodes have the required degree.

2.2.5 Proof Sketch of Theorem 2

We first argue that running one round of Luby's algorithm on an almost regular graph matches a constant fraction of the nodes with high probability. We note that while this claim is easy to see in expectation, obtaining a high probability bound requires the use of our shifted martingale technique, particularly when the graph becomes only almost regular (instead of fully regular, as in the first iteration). Combined with Lemma 6 that states that the resulting graph remains almost regular, we get that after $O(\log 1/\epsilon)$ rounds, at most ϵ-fraction of nodes remain unmatched.

2.2.6 Extension to Node-Averaged Complexity

We now sketch our result for the node-average complexity of MM given the Regular-Graph Preservation Lemma. We can combine Lemma 6 with the observation that in an (almost) regular graph, every node has a constant probability of being matched (and hence is only expected to survive a constant number of rounds under Luby). Since node-averaged complexity cares about the expected time to match a node, our initial rounds of Luby are consistent with our $O(1)$ node-averaged complexity goal. However, we cannot continue this trick until all nodes are matched, since eventually the graph will become too irregular for us to proceed. Instead, we only use Luby until a $O(1/\log\Delta)$ fraction of nodes remain; after so few nodes remain, it is safe for us to finish by repeatedly calling a previous black box [27] that takes $O(\log\Delta/\log\log\Delta)$ rounds to match a constant fraction of nodes but which can handle general (i.e. irregular) graphs. Similar to our previous argument, nodes are only expected to survive a constant number of black box invocations, but since there are so few nodes left our node-averaged complexity is still $O(1)$. Unlike the Luby phase, it is safe to continue this phase until all nodes are matched since it no longer matters how regular the graph is.

2.3 Overview of Lower Bounds

To prove lower bounds against LOCAL algorithms, we use some ideas from a lower bound construction of Ben-Basat, Kawarabayashi, and Schwartzman [32] that gave $\Omega(1/\epsilon)$ lower bounds in the LOCAL model for $(1+\epsilon)$ maximum matching as well as other approximate graph problems. The critical idea in that proof is that when an r-round LOCAL algorithm is deciding what to do with a node v (e.g. who to match it with), it can only use the local structure of the graph around v; in particular, it can only see the r-hop neighborhood around v. This means that we could cut out this r-hop neighborhood from the graph, put it back in differently, and the algorithm would have to make the same decision on v (or for randomized algorithms, the same distribution on decisions). The proof revolves around designing these r-hop neighborhoods as (symmetrical) gadgets, then showing that a constant number of gadgets will (with constant probability) induce an unmatched node between them. The BKS proof uses a simple path as their gadget which involves $O(r)$ nodes. Hence the algorithm can be shown to have an overall error rate of one error per $O(r)$ nodes, so the critical round threshold to allow for $(1+\epsilon)$-multiplicative approximations is $\Omega(1/\epsilon)$ rounds.

Relative to their result, the main upgrade we want to make is that the counterexample graph(s) should be Δ-regular. It is relatively straightforward to take BKS path gadgets and stick them into a large cycle, recovering their $\Omega(1/\epsilon)$ round lower bound for 2-regular bipartite graphs. The main technical hurdle we overcome is generalizing to higher degree. We know from our upper bounds that there must be some degradation as the degree Δ increases, so the main question is, how much efficiency do we need to lose to burn our excess degree? We design gadgets with $O(\Delta r)$ nodes and asymptotically maintain the original error rate of one error per constant number of gadgets (the cycle proof argues about the outcome of two gadgets inducing a mistake, but for the general case we reason about the outcome of five gadgets), yielding $\Omega(1/(\Delta\epsilon))$ round lower bounds. We discuss this formally in the full version.

3 An $O(\log 1/\epsilon)$-round Algorithm

In this section, we show that there is an $O(\log(1/\epsilon))$-round algorithm to find a $(1+\epsilon)$-approximate maximum matching in Δ-regular graphs when $\Delta \geq (\frac{1}{\epsilon})^c$ for a large constant c. Due to space constraints, we defer the pseudocode for the algorithm and full proofs to the full version. We note that the proof makes no attempt to optimize the constant c and we expect that it can be reduced significantly by a more careful analysis.

▶ **Theorem 2** (Main Result I). *Let $c = 10^5$ and let $\epsilon \in (0,1)$ be an accuracy parameter. For Δ-regular graphs with $\Delta > (1/\epsilon)^c$, there is an $O(\log(1/\epsilon))$-round CONGEST algorithm that finds a $(1+\epsilon)$-approximate maximum matching with high probability.*

A Roadmap for the Proof of Theorem 2. The algorithm for proving Theorem 2 first runs a simple color coding step to find a bipartite almost regular subgraph, and then runs Luby's algorithm for $O(\log(1/\epsilon))$ rounds. Since our analysis for Luby's algorithm uses several martingale inequalities, it is cleaner to work with the sequential view of Luby that we present in Section 3.1. In Section 3.2 we show that a single round of Luby's algorithm in an almost regular graph matches a constant fraction of the nodes, with high probability.[7] In Section 3.3, we state our key Regular-Graph Preservation Lemma that shows that running one round

[7] In fact, we prove a more general claim where it suffices that a constant fraction of the edges are balanced.

of Luby's algorithm on an almost regular graph and deleting the matched nodes yields an almost regular graph. A local version of that lemma, which bounds the probability that the degree of a particular node u almost exactly halves after each round is the most technical part of the proof and is given in the full version. Finally, in Section 3.4, we put these components together to finish the proof.

3.1 Sequential View of Luby's Algorithm

In the traditional distributed implementation of one round of Luby's algorithm, each edge e picks a uniformly random integer r_e in the set $\{1, 2, \ldots, M\}$ for an appropriately chosen polynomially large M. An edge e is chosen to be in the matching if $r_e < r_{e'}$ for all neighboring edges e'.

However, it is convenient for our analysis to work with a sequential view of Luby's algorithm. In the sequential view, the edges are sequentially sampled independently without replacement. When an edge e is sampled, it is added to the matching only if none of its neighboring edges had been sampled earlier. Crucially, unlike the greedy matching algorithm, a sampled edge e is *blocked* by a neighboring edge e' that was previously sampled even if e' itself is not in the matching. [69] showed that the two algorithms are equivalent by showing that they produce the same distribution over matchings with high probability[8]. We formalize this in the following proposition.

▶ **Proposition 7** (Proposition 3 in [69]). *For any unweighted graph G with m edges and constant $c' > 0$, let $\mathcal{D}_0$ and $\mathcal{D}_1$ be the distributions over matchings produced by the traditional and sequential views of Luby respectively. The total variation distance between $\mathcal{D}_0$ and $\mathcal{D}_1$ is at most $1/m^{c'}$; in particular*

$$\sum_{matchings\ \mathcal{M}_0} |\mathbb{P}_{\mathcal{M}\sim\mathcal{D}_0}[\mathcal{M} = \mathcal{M}_0] - \mathbb{P}_{\mathcal{M}\sim\mathcal{D}_1}[\mathcal{M} = \mathcal{M}_0]| \leq 1/m^{c'}$$

Following Proposition 7, in this paper we focus on analyzing the sequential view of Luby whenever we want to make claims about a single round of Luby's algorithm. This incurs an additional tiny $1/\operatorname{poly}(n)$ failure probability, which we can tolerate.

3.2 Analyzing One Round of Luby's Algorithm on Almost Regular Graphs

In this section we show that a single round of Luby's algorithm on almost regular graphs matches a constant fraction of the nodes with high probability. In fact, we prove this claim for a more general family of graphs in the following theorem.

▶ **Theorem 8.** *Let G be an undirected graph with n nodes and m edges, and let $d = 2m/n$ be the average degree. Let $E^{low} = \{(u, v) \in E(G) \mid deg(u) \leq 2d, deg(v) \leq 2d\}$ be the set of edges induced by nodes with degree at most $2d$. If $|E^{low}| \geq m/2$, then one round of Luby's algorithm finds a matching in G of size at least $n/288$ with high probability.*

At a high level, the proof of this theorem does the following. Consider the sequential view of Luby's algorithm. When an edge $e = \{u, v\}$ is sampled, it blocks at most $deg(u) + deg(v)$ other edges from joining the matching in the future. However, it can only block at most

[8] They stated the claim in the context of independent sets.

$2(2d) = 4d$ edges from E^{low}, because of the fact that each endpoint of an edge in E^{low} has degree at most $2d$. Thus, in the first Cn sampling steps of sequential Luby for some sufficiently small constant $C < 1$, at most $(4d)(Cn) = 4Cdn = 8Cm$ edges are blocked, meaning that the probability that any particular edge sampled during one of these steps is in E^{low} and has not been blocked is at least $(|E^{low}| - (8Cm))/(m - Cn) > 1/6$ as long as $C < 1/16$. Thus, the found matching has size at least $(Cn)(1/6)$ in expectation. Using martingale concentration inequalities, we can argue that the size of the matching found is at least $(1/2)(Cn)(1/6) > n/288$ with high probability, as desired. We give a formal version of this proof in the full version.

3.3 Regular-Graph Preservation

We first define some notation to facilitate the rest of the discussion. Intuitively, we say a node (α, Δ)-regular if all nodes in its two hop neighborhood have the same degree.

▶ **Definition 9** ((α, Δ)-regular node). *Let Δ be an integer and $\alpha \in [0, 1]$ be a real number. Given a graph G, we say that a node u is (α, Δ)-regular in G if for any $v \in \{u\} \cup N(u) \cup N^2(u)$, we have $\Delta(1 - \alpha) \leq deg(v) \leq \Delta(1 + \alpha)$.*

We can now present our main technical lemma that states that if u is a (α, Δ)-regular node, then its degree becomes almost exactly $\Delta/2$ after running one round of Luby's algorithm, with failure probability exponentially small in Δ. The proof of Lemma 10 is the most technical part of the paper and is deferred to the full version.

▶ **Lemma 10** (Local Regular-Graph Preservation Lemma). *Let $\Delta \geq 2^{10}$ be an integer, and $\alpha \in [0, 1/10]$ be a real number. Let G be a bipartite graph and u be an (α, Δ)-regular node in it. Let $deg'(u)$ be the number of unmatched neighbors of u after running sequential Luby on G. With probability at least $1 - \exp(-\Delta^{1/16})$, it holds that:*

$$\frac{\Delta}{2}\left(1 - (10\alpha + \Delta^{-1/600})\right) \leq deg'(u) \leq \frac{\Delta}{2}\left(1 + (10\alpha + \Delta^{-1/600})\right)$$

As long as Δ is large enough ($\approx \log n$), Lemma 10 followed by a union bound suffices to show that an almost regular graph remains almost regular with their degrees halved after one round of Luby's algorithm. However, for smaller Δ, we can no longer rely on a simple union bound. In the following lemma, we use the Chernoff-Hoeffding concentration inequality with bounded dependence to show that even for small Δ, *almost all* nodes halve their degree.

▶ **Lemma 11** (Regular-Graph Preservation Lemma). *Let $n \geq K \geq \Delta \geq 2^{10}$ be integers, and let $\alpha \in [0, 1/10]$, $\delta \in [0, 1/100]$ be real numbers. Let G be a bipartite graph with n nodes and max degree K and let G' be the graph obtained by running sequential Luby on G and removing all matched nodes together with their incident edges.*

If at least $(1 - \delta)$-fraction of nodes in G are (α, Δ)-regular, then at least $(1 - \delta')$-fraction of nodes in G' are $(\alpha', \Delta/2)$-regular with probability at least $1 - \exp\left(-n/(K^{10}\exp\left(\Delta^{1/99}\right))\right)$, where $\delta' = K^2 \cdot (\delta + 2\exp\left(-\Delta^{1/100}\right)$ and $\alpha' = 10\alpha + \Delta^{-1/600}$.

Proof. We say that a node v is *good* if its degree in G' is in the range $\frac{\Delta}{2}(1 \pm \alpha')$. Any node that's not good is called *bad*. Let R be the set of nodes that are (α, Δ)-regular in G. By Lemma 10, each (α, Δ)-regular node in G that remains unmatched is good with probability at least $1 - \exp\left(-\Delta^{1/16}\right)$. Hence, the expected number of nodes from R that are bad is at most $|R| \cdot \exp\left(-\Delta^{1/16}\right) \leq n$. To obtain a high probability bound, we use Chernoff-Hoeffding inequality with bounded dependence as follows.

For each node $u \in R$, let $B_u \in \{0,1\}$ be an indicator random variable that indicates whether u is bad. We have $\sum_{u\in R} \mathbb{E}[X_u] \leq |R| \cdot \exp\left(-\Delta^{1/16}\right) \leq n \cdot \exp\left(-\Delta^{1/16}\right)$. Since the maximum degree in G is K, each X_u depends on fewer than K^{10} nodes . Hence we apply Chernoff with bounded dependence (see full version for theorem statement) to get:

$$\begin{aligned}
\mathbb{P}\left[\sum_{u\in R} X_u > 2n\exp(-\Delta^{1/100})\right] &\leq \mathbb{P}\left[\sum_{u\in R} X_u > 2|R|\exp(-\Delta^{1/16})\right] \\
&\leq \mathbb{P}\left[\sum_{u\in R} X_u > \sum_{u\in R}\mathbb{E}[X_u] + |R|\exp(-\Delta^{1/16})\right] \\
&\leq \exp\left(-\frac{2(|R|\cdot\exp(-\Delta^{1/16}))^2}{|R|\cdot K^{10}}\right) \\
&= \exp\left(-\frac{2|R|}{K^{10}\exp\left(2\Delta^{1/16}\right)}\right) \\
&\leq \exp\left(-\frac{2|R|}{K^{10}\exp\left(\Delta^{1/99}\right)}\right) \leq \exp\left(-\frac{n}{K^{10}\exp\left(\Delta^{1/99}\right)}\right)
\end{aligned}$$

where the last line used $\Delta > 2^{10}$ and $|R| \geq (1-\delta)n \geq n/2$. We say that a node v poisons a node u if v prevents u from being $(\alpha', \Delta/2)$-regular, i.e., v poisons u if v is *bad* and is at distance at most 2 from u. We note that each bad node poisons at most K^2 nodes. Let B be the set of bad nodes, i.e., we have $|B| = \sum_{u\in R} X_u$ and let $\tilde{B} = V(G) \setminus R$ be the set of nodes that are not (α, Δ) regular in G. For nodes in $\tilde{B}$, we have no guarantee on the probability of whether they become good, so we always assume that they poison K^2 nodes. In total, the number of nodes that are not $(\alpha', \Delta/2)$-regular in G' is at most $(|\tilde{B}| + |B|)K^2$ which is at most:

$$\begin{aligned}
(|\tilde{B}| + 2n\cdot\exp\left(-\Delta^{1/100}\right))\cdot K^2 &\leq (\delta n + 2n\cdot\exp\left(-\Delta^{1/100}\right))\cdot K^2 \\
&= n\cdot K^2\cdot(\delta + 2\exp\left(-\Delta^{1/100}\right))
\end{aligned}$$

with probability at least $1 - \exp\left(-n/(K^{10}\exp\left(\Delta^{1/99}\right))\right)$, as desired. ◀

Lemma 11 shows that after each round of Luby's algorithm, the remaining graph still satisfies the requirement that most vertices remain almost regular, albeit with worsening parameters. The following technical claim shows that these parameters remain small enough after $O(\log(1/\epsilon)$ rounds. The proof uses basic arithmetic and is deferred to the appendix.

▷ Claim 12. Let Δ be an integer, $c = 1/10^5$, and ϵ be a real number satisfying $1 > \epsilon \geq 1/\Delta^c$. Furthermore let:

1. $\alpha_0 = \Delta^{-1/600}$ and $\alpha_i = 10\alpha_{i-1} + \Delta^{-1/600}$ for $i \geq 1$,
2. $\delta_0 = \exp(-\Delta^{1/200})$, and $\delta_i = \Delta^2(\delta_{i-1} + 2\exp(-(\Delta/2^i)^{1/100}))$ for $i \geq 1$.

It holds that $\alpha_i \leq 1/10$ and $\delta_i \leq \exp(-\Delta^{1/300})$ for $1 \leq i \leq 10\log(1/\epsilon)$.

We now extend Lemma 11 to a multi-round argument, showing that most of nodes' degrees after running Luby's algorithm for i rounds become $\approx \Delta/2^i$.

▶ **Lemma 13** (Multi-Round Regular-Graph Preservation). *Let $(\Delta, \epsilon, (\alpha_i), (\delta_i))$ be as defined in Claim 12. Let G be a bipartite n-node graph, where all but $\delta = \delta_0 = \exp\left(-\Delta^{1/200}\right)$-fraction of the nodes are (α_0, Δ)-regular. For $i \leq 10\log(1/\epsilon)$, let G_i be the graph obtained after running Luby's algorithm for i rounds on G, where in each round we remove the matched nodes together with their incident edges. If $|V(G_i)| \geq \epsilon n$, then with high probability, at least $(1-\delta_i)$-fraction of the nodes in G_i are $(\alpha_i, \Delta/2^i)$-regular.*

Proof. The idea is to use a recursive argument where we apply Lemma 10 with a simple union bound in the dense case, and Lemma 11 in the sparse case. For convenience of notation, let $\Delta_i = \Delta/2^i$. Since we apply Lemmas 10 and 11 recursively for i rounds, we need to ensure that $\alpha_i \leq 1/10$ and $\delta_i \leq 1/100$, which follows from Claim 12. The proof is split into two cases, a dense case where $\Delta \geq \log^{50} n$, and a sparse case. We start with the dense case.

Dense case where $\Delta \geq \log^{50} n$. By Lemma 10 and a simple union bound argument, in the case where $\Delta \geq \log^{50} n$, *all* the nodes are $(\alpha_i, \Delta/2^i)$-regular after running Luby's algorithm for i rounds with high probability. This is because the failure probability of Lemma 10 is at most $\exp(-\Delta_i^{1/16})$ at round i, and $\Delta_i \geq \Delta^{0.99}$ for any i (since $\epsilon \geq 1/\Delta^{1/c}$ and $i \leq 10\log(1/\epsilon)$).

Sparse case where $\Delta \leq \log^{50} n$. The idea is to apply Lemma 11 recursively for i rounds and the proof follows by induction.

Induction base case i=1. By Proposition 7, traditional and sequential Luby produce the same distributions over matchings in the graph G, up to $1/\operatorname{poly}(n)$ total variation distance. Hence, for $i = 1$ the claim follows directly from Lemma 11, since the failure probability of Lemma 11 is only $\exp\left(-n/\Delta^{10}\exp\left(\Delta^{1/99}\right)\right) < 1/n^{1000}$, for $\Delta \leq \log^{50} n$.

Induction step. Assume that the claim is true for i, i.e., assume that all but $|V(G_i)|\cdot\delta_i$ nodes in G_i are $(\alpha_i, \Delta/2^i)$-regular with high probability. We show that the claim is true for $i+1$. Again, by Proposition 7, traditional Luby and sequential Luby produce the same distributions over matchings in the graph G_i, up to a $1/|E(G_i)|^c \leq 1/(|V(G_i)|)^c \leq 1/(\epsilon n)^c \leq (1/n^{0.99})^c$ total variation distance, for an arbitrarily large constant c. Since the max degree in G_i is at most Δ, by Lemma 11 it holds that all but $|V(G_{i+1})|\cdot\delta_{i+1}$ nodes in G_{i+1} are $(\alpha_{i+1}, \Delta/2^{i+1})$-regular in G_{i+1} with probability at least

$$1 - \exp\left(-\frac{|V(G_i)|}{\Delta^{10}\exp\left(\Delta_i^{1/99}\right)}\right) \geq 1 - \exp\left(-\frac{\epsilon n}{\Delta^{10}\exp\left(\Delta^{1/99}\right)}\right) \geq 1 - \frac{1}{n^{1000}}$$

where the first inequality holds since $\Delta_i \leq \Delta$ and $|V(G_i)| \geq \epsilon n$, and the second inequality holds since $\epsilon \geq 1/\Delta^{1/100} \geq 1/n^{1/100}$ and $\Delta \leq \log^{50} n$. Hence, by a union bound on all i's, we get that as long as $V(G_i)$ has at least ϵn nodes, it holds that all but $|V(G_i)|\cdot\delta_i$ nodes are (α_i, Δ_i)-regular in G_i with probability at least $1 - 1/n^{100}$. ◀

3.4 Putting it together

Lemma 13 shows that the graph remains almost regular after $i \approx \log(1/\epsilon)$ repeated applications of Luby's algorithm, and Theorem 8 showed that each round of Luby's on an almost regular graph matches a constant fraction of nodes. We can now combine these two results to show that as long as the remaining graph is large enough, each application of Luby's algorithm matches a constant fraction of nodes.

We first present a corollary of Theorem 8 to formalize that the almost regular graphs implied by Lemma 13 do indeed satisfy the requirements of Theorem 8. We defer the proof to the appendix for brevity.

▶ **Corollary 14.** *Let $\Delta \geq C$, where C is a large enough constant, and let G' be a graph with n' nodes and max degree Δ. Suppose at least $(1 - \exp(-\Delta^{1/300}))$-fraction of the nodes are (α', Δ')-regular for $\alpha' \leq 1/10$. It holds that Luby's algorithm matches $n'/288$ nodes in G' with high probability.*

We are now ready to prove Theorem 2.

Proof of Theorem 2. We first provide a proof that assumes that the graph in bipartite, and then we lift this assumption by a simple sampling argument. We simply run Luby's algorithm in G for $i = 10\log(1/\epsilon)$ rounds. By Lemma 13, all but an $\exp(-\Delta^{1/300})$-fraction of the nodes in the graph are (α_i, Δ_i)-regular at round i with high probability. Hence, by Corollary 14, in each of these rounds we match at least a $(1/288)$-fraction of the nodes. Therefore, after $O(\log(1/\epsilon))$, all but ϵ-fraction of the nodes are matched, as desired.

Overcoming bipartiteness. Lemma 13 assumes that the input graph is bipartite. To overcome this, we apply a simple color-coding trick. Before running Luby's algorithm, each node picks a uniformly random color independently in $\{0, 1\}$. This naturally defines a bipartite subgraph graph G' by ignoring all monocolored edges. Observe that for a given node $u \in G$, the degree of u in G' is $(\Delta/2)(1 \pm \Delta^{-0.4})$ with probability at least $1 - \exp(-\Delta^{0.2}/6)$ by a standard Chernoff-Hoeffding bound. Hence, if $\Delta \geq \log^{50} n$, then all the nodes degrees in G' are $\Delta(1 \pm \Delta^{-0.4})$ with high probability. Therefore, all the nodes are also $(\Delta^{-0.4}, \Delta)$-regular in that case. Otherwise, if $\Delta < \log^{50} n$, then we can use a bounded dependence Chernoff-Hoeffding type argument, as follows. By a union bound, the probability that a node u isn't $(\Delta^{-0.4}, \Delta)$-regular is at most $\Delta^2 \cdot \exp(-\Delta^{0.2}/6) \leq \exp(-\Delta^{0.1})$. Hence, the expected number of nodes that aren't $(\Delta^{-0.4}, \Delta)$-regular is at most $n \cdot \exp(-\Delta^{0.1})$. Furthermore, the event of whether a node is $(\Delta^{-0.4}, \Delta)$-regular depends only on $< \Delta^{10}$ other nodes. Hence, by applying Chernoff with bounded dependence (see full version for details), we get that with high probability, all but $n \cdot \exp(-\Delta^{1/200})$ nodes are $(\Delta^{-0.4}, \Delta)$-regular in G'. Hence, since $\Delta^{-0.4} \leq \Delta^{-1/600}$, it suffices to apply Lemma 13 on G'. This concludes the proof. ◀

References

1 Mohamad Ahmadi and Fabian Kuhn. Distributed maximum matching verification in CONGEST. In *DISC*, volume 179 of *LIPIcs*, pages 37:1–37:18, 2020. doi:10.4230/LIPICS.DISC.2020.37.

2 Mohamad Ahmadi, Fabian Kuhn, and Rotem Oshman. Distributed approximate maximum matching in the CONGEST model. In *DISC*, volume 121 of *LIPIcs*, pages 6:1–6:17, 2018. doi:10.4230/LIPICS.DISC.2018.6.

3 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. doi:10.1016/0196-6774(86)90019-2.

4 Noga Alon, Sonny Ben-Shimon, and Michael Krivelevich. A note on regular ramsey graphs. *J. Graph Theory*, 64(3):244–249, 2010. doi:10.1002/JGT.20453.

5 Noga Alon, Amin Coja-Oghlan, Hiêp Hàn, Mihyun Kang, Vojtech Rödl, and Mathias Schacht. Quasi-randomness and algorithmic regularity for graphs with general degree distributions. *SIAM J. Comput.*, 39(6):2336–2362, 2010. doi:10.1137/070709529.

6 Noga Alon, Shmuel Friedland, and Gil Kalai. Every 4-regular graph plus an edge contains a 3-regular subgraph. *J. Comb. Theory B*, 37(1):92–93, 1984. doi:10.1016/0095-8956(84)90048-0.

7 Noga Alon, Shmuel Friedland, and Gil Kalai. Regular subgraphs of almost regular graphs. *J. Comb. Theory B*, 37(1):79–91, 1984. doi:10.1016/0095-8956(84)90047-9.

8 Noga Alon and Guy Moshkovitz. Limitations on regularity lemmas for clustering graphs. *Adv. Appl. Math.*, 124:102135, 2021. doi:10.1016/J.AAM.2020.102135.

9 Noga Alon and Pawel Pralat. Modular orientations of random and quasi-random regular graphs. *Comb. Probab. Comput.*, 20(3):321–329, 2011. doi:10.1017/S0963548310000544.

10 Moab Arar, Shiri Chechik, Sarel Cohen, Cliff Stein, and David Wajc. Dynamic matching: Reducing integral algorithms to approximately-maximal fractional algorithms. In *ICALP*, volume 107 of *LIPIcs*, pages 7:1–7:16, 2018. doi:10.4230/LIPICS.ICALP.2018.7.

11 Armen S. Asratian and Nikolai N. Kuzjurin. On the number of nearly perfect matchings in almost regular uniform hypergraphs. *Discret. Math.*, 207(1-3):1–8, 1999.

12 Sepehr Assadi, Soheil Behnezhad, Sanjeev Khanna, and Huan Li. On regularity lemma and barriers in streaming and dynamic matching. In *STOC*, pages 131–144, 2023. doi:10.1145/3564246.3585110.

13 Sepehr Assadi and Janani Sundaresan. Hidden permutations to the rescue: Multi-pass streaming lower bounds for approximate matchings. In *FOCS*, pages 909–932, 2023. doi:10.1109/FOCS57990.2023.00058.

14 László Babai. On the complexity of canonical labeling of strongly regular graphs. *SIAM J. Comput.*, 9(1):212–216, 1980. doi:10.1137/0209018.

15 László Babai. On the automorphism groups of strongly regular graphs I. In *ITCS*, pages 359–368. ACM, 2014. doi:10.1145/2554797.2554830.

16 László Babai, Xi Chen, Xiaorui Sun, Shang-Hua Teng, and John Wilmes. Faster canonical forms for strongly regular graphs. In *FOCS*, pages 157–166. IEEE Computer Society, 2013. doi:10.1109/FOCS.2013.25.

17 Alkida Balliu, Thomas Boudier, Sebastian Brandt, and Dennis Olivetti. Tight lower bounds in the supported LOCAL model. In *PODC*, pages 95–105. ACM, 2024. doi:10.1145/3662158.3662798.

18 Alkida Balliu, Sebastian Brandt, Yi-Jun Chang, Dennis Olivetti, Mikaël Rabie, and Jukka Suomela. The distributed complexity of locally checkable problems on paths is decidable. In Peter Robinson and Faith Ellen, editors, *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, PODC 2019, Toronto, ON, Canada, July 29 - August 2, 2019*, pages 262–271. ACM, 2019. doi:10.1145/3293611.3331606.

19 Alkida Balliu, Sebastian Brandt, Yi-Jun Chang, Dennis Olivetti, Jan Studený, and Jukka Suomela. Efficient classification of locally checkable problems in regular trees. In Christian Scheideler, editor, *36th International Symposium on Distributed Computing, DISC 2022, October 25-27, 2022, Augusta, Georgia, USA*, volume 246 of *LIPIcs*, pages 8:1–8:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICS.DISC.2022.8.

20 Alkida Balliu, Sebastian Brandt, Yi-Jun Chang, Dennis Olivetti, Jan Studený, Jukka Suomela, and Aleksandr Tereshchenko. Locally checkable problems in rooted trees. *Distributed Comput.*, 36(3):277–311, 2023. doi:10.1007/S00446-022-00435-9.

21 Alkida Balliu, Sebastian Brandt, Juho Hirvonen, Dennis Olivetti, Mikaël Rabie, and Jukka Suomela. Lower bounds for maximal matchings and maximal independent sets. *J. ACM*, 68(5):39:1–39:30, 2021. doi:10.1145/3461458.

22 Alkida Balliu, Sebastian Brandt, Fabian Kuhn, and Dennis Olivetti. Improved distributed lower bounds for MIS and bounded (out-)degree dominating sets in trees. In *PODC*, pages 283–293. ACM, 2021. doi:10.1145/3465084.3467901.

23 Alkida Balliu, Sebastian Brandt, Fabian Kuhn, and Dennis Olivetti. Distributed Δ-coloring plays hide-and-seek. In *STOC*, pages 464–477. ACM, 2022. doi:10.1145/3519935.3520027.

24 Alkida Balliu, Sebastian Brandt, Fabian Kuhn, and Dennis Olivetti. Distributed maximal matching and maximal independent set on hypergraphs. In *SODA*, pages 2632–2676. SIAM, 2023. doi:10.1137/1.9781611977554.CH100.

25 Alkida Balliu, Sebastian Brandt, and Dennis Olivetti. Distributed lower bounds for ruling sets. *SIAM J. Comput.*, 51(1):70–115, 2022. doi:10.1137/20M1381770.

26 Alkida Balliu, Mohsen Ghaffari, Fabian Kuhn, and Dennis Olivetti. Node and edge averaged complexities of local graph problems. *Distributed Comput.*, 36(4):451–473, 2023. doi:10.1007/S00446-023-00453-1.

27 Reuven Bar-Yehuda, Keren Censor-Hillel, Mohsen Ghaffari, and Gregory Schwartzman. Distributed approximation of maximum independent set and maximum matching. In *PODC*, pages 165–174, 2017. doi:10.1145/3087801.3087806.

28 Reuven Bar-Yehuda, Keren Censor-Hillel, and Gregory Schwartzman. A distributed $(2+\epsilon)$-approximation for vertex cover in $O(\log\delta/\epsilon \log\log\delta)$ rounds. In *PODC*, pages 3–8. ACM, 2016. doi:10.1145/2933057.2933086.

29 Leonid Barenboim, Michael Elkin, Seth Pettie, and Johannes Schneider. The locality of distributed symmetry breaking. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 321–330. IEEE Computer Society, 2012. doi:10.1109/FOCS.2012.60.

30 Leonid Barenboim and Yaniv Tzur. Distributed symmetry-breaking with improved vertex-averaged complexity. *CoRR*, abs/1805.09861, 2018. arXiv:1805.09861.

31 Ran Ben-Basat, Guy Even, Ken-ichi Kawarabayashi, and Gregory Schwartzman. Optimal distributed covering algorithms. *Distributed Comput.*, 36(1):45–55, 2023. doi:10.1007/S00446-021-00391-W.

32 Ran Ben-Basat, Ken-ichi Kawarabayashi, and Gregory Schwartzman. Parameterized distributed algorithms. In *DISC*, 2019.

33 Sayan Bhattacharya, Peter Kiss, and Thatchaphol Saranurak. Dynamic $(1+\epsilon)$-approximate matching size in truly sublinear update time. In *FOCS*, pages 1563–1588, 2023.

34 Sayan Bhattacharya, Peter Kiss, Thatchaphol Saranurak, and David Wajc. Dynamic matching with better-than-2 approximation in polylogarithmic update time. In *SODA*, pages 100–128, 2023. doi:10.1137/1.9781611977554.CH5.

35 Sebastian Brandt. An automatic speedup theorem for distributed problems. In *PODC*, pages 379–388. ACM, 2019. doi:10.1145/3293611.3331611.

36 Sebastian Brandt, Orr Fischer, Juho Hirvonen, Barbara Keller, Tuomo Lempiäinen, Joel Rybicki, Jukka Suomela, and Jara Uitto. A lower bound for the distributed lovász local lemma. In *STOC*, pages 479–488. ACM, 2016. doi:10.1145/2897518.2897570.

37 Sebastian Brandt, Juho Hirvonen, Janne H. Korhonen, Tuomo Lempiäinen, Patric R. J. Östergård, Christopher Purcell, Joel Rybicki, Jukka Suomela, and Przemyslaw Uznanski. LCL problems on grids. In Elad Michael Schiller and Alexander A. Schwarzmann, editors, *Proceedings of the ACM Symposium on Principles of Distributed Computing, PODC 2017, Washington, DC, USA, July 25-27, 2017*, pages 101–110. ACM, 2017. doi:10.1145/3087801.3087833.

38 Sebastian Brandt, Yannic Maus, Ananth Narayanan, Florian Schager, and Jara Uitto. On the locality of hall's theorem. In Yossi Azar and Debmalya Panigrahi, editors, *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2025, New Orleans, LA, USA, January 12-15, 2025*, pages 4198–4226. SIAM, 2025. doi:10.1137/1.9781611978322.143.

39 Sebastian Brandt and Dennis Olivetti. Truly tight-in-Δ bounds for bipartite maximal matching and variants. In Yuval Emek and Christian Cachin, editors, *PODC*, pages 69–78. ACM, 2020. doi:10.1145/3382734.3405745.

40 Niv Buchbinder, Joseph (Seffi) Naor, and David Wajc. Lossless online rounding for online bipartite matching (despite its impossibility). In *SODA*, pages 2030–2068, 2023. doi:10.1137/1.9781611977554.CH78.

41 Marc Bury, Elena Grigorescu, Andrew McGregor, Morteza Monemizadeh, Chris Schwiegelshohn, Sofya Vorotnikova, and Samson Zhou. Structural results on matching estimation with applications to streaming. *Algorithmica*, 81(1):367–392, 2019. doi:10.1007/S00453-018-0449-Y.

42 Teena Carroll, David J. Galvin, and Prasad Tetali. Matchings and independent sets of a fixed size in regular graphs. *J. Comb. Theory A*, 116(7):1219–1227, 2009. doi:10.1016/J.JCTA.2008.12.008.

43 Yi-Jun Chang, Tsvi Kopelowitz, and Seth Pettie. An exponential separation between randomized and deterministic complexity in the LOCAL model. *SIAM J. Comput.*, 48(1):122–143, 2019. doi:10.1137/17M1117537.

44 Soumyottam Chatterjee, Robert Gmyr, and Gopal Pandurangan. Sleeping is efficient: MIS in $O(1)$-rounds node-averaged awake complexity. In Yuval Emek and Christian Cachin, editors, *PODC*, pages 99–108. ACM, 2020. doi:10.1145/3382734.3405718.

45 Andrzej Czygrinow and Michal Hanckowiak. Distributed algorithm for better approximation of the maximum matching. In *COCOON*, volume 2697 of *Lecture Notes in Computer Science*, pages 242–251, 2003. doi:10.1007/3-540-45071-8_26.

46 Guy Even, Moti Medina, and Dana Ron. Distributed maximum matching in bounded degree graphs. In *ICDCN*, pages 18:1–18:10, 2015. doi:10.1145/2684464.2684469.

47 Laurent Feuilloley. How long it takes for an ordinary node with an ordinary ID to output? In Shantanu Das and Sébastien Tixeuil, editors, *SIROCCO*, volume 10641 of *Lecture Notes in Computer Science*, pages 263–282. Springer, 2017. doi:10.1007/978-3-319-72050-0_16.

48 Manuela Fischer. Improved deterministic distributed matching via rounding. *Distributed Comput.*, 33(3-4):279–291, 2020. doi:10.1007/S00446-018-0344-4.

49 Manuela Fischer, Mohsen Ghaffari, and Fabian Kuhn. Deterministic distributed edge-coloring via hypergraph maximal matching. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 180–191. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.25.

50 Manuela Fischer, Slobodan Mitrovic, and Jara Uitto. Deterministic $(1+\epsilon)$-approximate maximum matching with poly$(1/\epsilon)$ passes in the semi-streaming model and beyond. In *STOC*, pages 248–260, 2022. doi:10.1145/3519935.3520039.

51 Mohsen Ghaffari. An improved distributed algorithm for maximal independent set. In *SODA*, pages 270–277. SIAM, 2016. doi:10.1137/1.9781611974331.ch20.

52 Mohsen Ghaffari, Themis Gouleakis, Christian Konrad, Slobodan Mitrovic, and Ronitt Rubinfeld. Improved massively parallel computation algorithms for MIS, matching, and vertex cover. In *PODC*, pages 129–138. ACM, 2018. doi:10.1145/3212734.3212743.

53 Mohsen Ghaffari and Christoph Grunau. Faster deterministic distributed MIS and approximate matching. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 1777–1790. ACM, 2023. doi:10.1145/3564246.3585243.

54 Mohsen Ghaffari and Christoph Grunau. Near-optimal deterministic network decomposition and ruling set, and improved MIS. In *65th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2024, Chicago, IL, USA, October 27-30, 2024*, pages 2148–2179. IEEE, 2024. doi:10.1109/FOCS61266.2024.00007.

55 Mohsen Ghaffari, David G. Harris, and Fabian Kuhn. On derandomizing local distributed algorithms. In *FOCS*, pages 662–673, 2018. doi:10.1109/FOCS.2018.00069.

56 Mohsen Ghaffari, Fabian Kuhn, Yannic Maus, and Jara Uitto. Deterministic distributed edge-coloring with fewer colors. In *STOC*, pages 418–430, 2018. doi:10.1145/3188745.3188906.

57 Ashish Goel, Michael Kapralov, and Sanjeev Khanna. Perfect matchings in O(n log n) time in regular bipartite graphs. *SIAM Journal on Computing*, 42(3):1392–1404, 2013. doi:10.1137/100812513.

58 Louis Golowich. A new berry-esseen theorem for expander walks. In *STOC*, pages 10–22. ACM, 2023. doi:10.1145/3564246.3585141.

59 Fabrizio Grandoni, Stefano Leonardi, Piotr Sankowski, Chris Schwiegelshohn, and Shay Solomon. $(1 + \epsilon)$-approximate incremental matching in constant deterministic amortized time. In *SODA*, pages 1886–1898, 2019. doi:10.1137/1.9781611975482.114.

60 Christoph Grunau, Václav Rozhon, and Sebastian Brandt. The landscape of distributed complexities on trees and beyond. In Alessia Milani and Philipp Woelfel, editors, *PODC '22: ACM Symposium on Principles of Distributed Computing, Salerno, Italy, July 25 - 29, 2022*, pages 37–47. ACM, 2022. doi:10.1145/3519270.3538452.

61 Anupam Gupta, Guru Guruganesh, Binghui Peng, and David Wajc. Stochastic online metric matching. In *ICALP*, volume 132 of *LIPIcs*, pages 67:1–67:14, 2019. doi:10.4230/LIPICS.ICALP.2019.67.

62 Manoj Gupta and Richard Peng. Fully dynamic (1+ ϵ)-approximate matchings. In *FOCS*, pages 548–557, 2013. doi:10.1109/FOCS.2013.65.

63 David G. Harris. Distributed local approximation algorithms for maximum matching in graphs and hypergraphs. In *FOCS*, pages 700–724, 2019. doi:10.1109/FOCS.2019.00048.

64 John E. Hopcroft and Richard M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM J. Comput.*, 2(4):225–231, 1973. doi:10.1137/0202019.

65 Amos Israeli and Alon Itai. A fast and simple randomized parallel algorithm for maximal matching. *Inf. Process. Lett.*, 22(2):77–80, 1986. doi:10.1016/0020-0190(86)90144-4.

66 Taisuke Izumi, Naoki Kitamura, and Yutaro Yamaguchi. A nearly linear-time distributed algorithm for exact maximum matching. In David P. Woodruff, editor, *Proceedings of the 2024 ACM-SIAM Symposium on Discrete Algorithms, SODA 2024, Alexandria, VA, USA, January 7-10, 2024*, pages 4062–4082. SIAM, 2024. doi:10.1137/1.9781611977912.141.

67 Chinmay Karande, Aranyak Mehta, and Pushkar Tripathi. Online bipartite matching with unknown distributions. In *STOC*, pages 587–596, 2011. doi:10.1145/1993636.1993715.

68 Richard M. Karp, Umesh V. Vazirani, and Vijay V. Vazirani. An optimal algorithm for on-line bipartite matching. In *STOC*, pages 352–358, 1990. doi:10.1145/100216.100262.

69 Ken-ichi Kawarabayashi, Seri Khoury, Aaron Schild, and Gregory Schwartzman. Improved distributed approximations for maximum independent set. In *DISC*, volume 179 of *LIPIcs*, pages 35:1–35:16, 2020. doi:10.4230/LIPICS.DISC.2020.35.

70 Ludek Kucera. Canonical labeling of regular graphs in linear average time. In *FOCS*, pages 271–279. IEEE Computer Society, 1987. doi:10.1109/SFCS.1987.11.

71 Fabian Kuhn, Thomas Moscibroda, and Roger Wattenhofer. The price of being near-sighted. In *SODA*, pages 980–989. ACM Press, 2006. URL: http://dl.acm.org/citation.cfm?id=1109557.1109666.

72 Fabian Kuhn, Thomas Moscibroda, and Roger Wattenhofer. Local computation: Lower and upper bounds. *J. ACM*, 63(2):17:1–17:44, 2016. doi:10.1145/2742012.

73 Nathan Linial. Locality in distributed graph algorithms. *SIAM J. Comput.*, 21(1):193–201, 1992. doi:10.1137/0221015.

74 Zvi Lotker, Boaz Patt-Shamir, and Seth Pettie. Improved distributed approximate matching. *J. ACM*, 62(5):38:1–38:17, 2015. doi:10.1145/2786753.

75 Zvi Lotker, Boaz Patt-Shamir, and Adi Rosén. Distributed approximate matching. *SIAM J. Comput.*, 39(2):445–460, 2009. doi:10.1137/080714403.

76 Michael Luby. A simple parallel algorithm for the maximal independent set problem. *SIAM J. Comput.*, 15(4):1036–1053, 1986. doi:10.1137/0215074.

77 Aleksander Madry. Navigating central path with electrical flows: From flows to matchings, and back. In *FOCS*, pages 253–262, 2013. doi:10.1109/FOCS.2013.35.

78 Andrew McGregor. Finding graph matchings in data streams. In *APPROX*, volume 3624 of *Lecture Notes in Computer Science*, pages 170–181, 2005. doi:10.1007/11538462_15.

79 Aranyak Mehta. Online matching and ad allocation. *Found. Trends Theor. Comput. Sci.*, 8(4):265–368, 2013. doi:10.1561/0400000057.

80 Moni Naor. A lower bound on probabilistic algorithms for distributive ring coloring. *SIAM J. Discret. Math.*, 4(3):409–412, 1991. doi:10.1137/0404036.

81 Moni Naor and Larry J. Stockmeyer. What can be computed locally? *SIAM J. Comput.*, 24(6):1259–1277, 1995. doi:10.1137/S0097539793254571.

82 Ami Paz and Gregory Schwartzman. A (2+ϵ)-approximation for maximum weight matching in the semi-streaming model. *ACM Trans. Algorithms*, 15(2):18:1–18:15, 2019. doi:10.1145/3274668.

83 David Peleg. *Distributed computing: a locality-sensitive approach*. Society for Industrial and Applied Mathematics, USA, 2000.

84 Václav Rozhon. Invitation to local algorithms. *CoRR*, abs/2406.19430, 2024. doi:10.48550/arXiv.2406.19430.

85 Shay Solomon. Fully dynamic maximal matching in constant update time. In *FOCS*, pages 325–334, 2016. doi:10.1109/FOCS.2016.43.

86 Daniel A. Spielman. Faster isomorphism testing of strongly regular graphs. In *STOC*, pages 576–584. ACM, 1996. doi:10.1145/237814.238006.

87 Jukka Suomela. Survey of local algorithms. *ACM Comput. Surv.*, 45(2):24:1–24:40, 2013. doi:10.1145/2431211.2431223.

88 Jan van den Brand, Yin Tat Lee, Danupon Nanongkai, Richard Peng, Thatchaphol Saranurak, Aaron Sidford, Zhao Song, and Di Wang. Bipartite matching in nearly-linear time on moderately dense graphs. In *FOCS*, pages 919–930, 2020. doi:10.1109/FOCS46700.2020.00090.

89 Raphael Yuster. Maximum matching in regular and almost regular graphs. *Algorithmica*, 66(1):87–92, 2013. doi:10.1007/S00453-012-9625-7.

90 Or Zamir. Algorithmic applications of hypergraph and partition containers. In *STOC*, pages 985–998. ACM, 2023. doi:10.1145/3564246.3585163.