

Efficient Computation of Balanced Structures^{*}

David G. Harris^{**}, Ehab Morsy^{***}, Gopal Pandurangan[†], Peter Robinson[‡], and Aravind Srinivasan[§]

Abstract. Basic graph structures such as maximal independent sets (MIS's) have spurred much theoretical research in randomized and distributed algorithms, and have several applications in networking and distributed computing as well. However, the extant (distributed) algorithms for these problems do not necessarily guarantee fault-tolerance or load-balance properties. We propose and study “low-average degree” or “balanced” versions of such structures. Interestingly, in sharp contrast to, say, MIS's, it can be shown that checking whether a structure is balanced, will take substantial time. Nevertheless, we are able to develop good sequential/distributed (randomized) algorithms for such balanced versions. We also complement our algorithms with several lower bounds. Randomization plays a key role in our upper and lower bound results.

Keywords: Graph algorithms, Distributed algorithms, Randomization, Approximation algorithms, Maximal Independent Set, Lower Bounds.

1 Introduction

Graph-theoretic structures such as maximal independent sets (MIS's) and minimal dominating sets (MDS's) are fundamental to graph theory, and their efficient computation is especially useful in the context of distributed computing and networks [12]. MIS, for example, is a basic building block in distributed computing and is useful in basic tasks such as monitoring, scheduling, routing, and clustering [11, 13]; furthermore, the development of fast parallel/distributed algorithms for it has spurred fundamental progress in randomized algorithms and in derandomization [1, 7, 10]. Extensive research has gone into designing fast distributed algorithms for these problems since the early eighties: see [9, 15] and the references therein. We now know that problems such as MIS are quite *local*, i.e., that they admit distributed algorithms that run in a small number of *rounds*: typically (poly-)logarithmic in the network size n (n will denote the number of nodes in the network throughout unless specified otherwise). However, one main drawback of these algorithms

^{*} A preliminary version of this work appears in the *Proc. International Colloquium on Automata, Languages, and Programming* (ICALP), 2013.

^{**} Department of Applied Mathematics, University of Maryland, College Park, MD 20742. davidgharris29@hotmail.com. Supported in part by NSF Award CNS-1010789.

^{***} Division of Mathematical Sciences, Nanyang Technological University, Singapore 637371 and Department of Mathematics, Suez Canal University, Ismailia 22541, Egypt. ehabmorsy@gmail.com.

[†] Division of Mathematical Sciences, Nanyang Technological University, Singapore 637371 and Department of Computer Science, Brown University, Providence, RI 02912. gopalpandurangan@gmail.com. Supported in part by Nanyang Technological University grant M58110000, Singapore Ministry of Education (MOE) Academic Research Fund (AcRF) Tier 2 grant MOE2010-T2-2-082, Singapore MOE AcRF Tier 1 grant MOE2012-T1-001-094, and by a grant from the United States-Israel Binational Science Foundation (BSF).

[‡] Division of Mathematical Sciences, Nanyang Technological University, Singapore 637371. peter.robinson@ntu.edu.sg. Supported in part by Nanyang Technological University grant M58110000 and Singapore Ministry of Education (MOE) Academic Research Fund (AcRF) Tier 2 grant MOE2010-T2-2-082.

[§] Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742. srin@cs.umd.edu. Supported in part by NSF Award CNS-1010789.

is that there is no guarantee on the *quality* of the structure output. For example, the classical MIS algorithms of Alon, Babai & Itai [1] and Luby [10] compute an MIS in $O(\log n)$ rounds with high probability; their focus is not on additional properties of the output MIS. In this paper, we initiate a systematic study of “balanced” versions of these structures, i.e., the *average degree* – in the original graph – of the nodes belonging to the structure is “small”; this study is motivated by both theoretical and practical considerations.

1.1 Problems Addressed

We consider an undirected simple graph $G = (V, E)$ with n nodes and m edges. We denote the average degree of G by $d = d(G) = \frac{2m}{n}$; *this parameter will play a key role in our results*. More generally, given any subset $S \subseteq V$, we define the average degree of S , denoted by d_S , as the total degree (in G) of the vertices of S divided by the number of vertices in S , i.e., $d_S = \frac{\sum_{v \in S} d_v}{|S|}$, where d_v is the degree of node v in G . We assume that G has no isolated vertices for convenience; this assumption can be easily removed.

Recall the following fundamental graph structures:

- a Maximal Independent Set (MIS) is an inclusion-maximal vertex subset $S \subseteq V$ such that no two vertices in S are neighbors;
- a Minimal Dominating Set (MDS) is an inclusion-minimal vertex subset $S \subseteq V$ such that every vertex in G is either in S or is a neighbor of a vertex in S ; and
- a Minimal Vertex Cover (MVC) is an inclusion-minimal vertex subset $S \subseteq V$ such that every edge in G has at least one endpoint in S .

This paper is concerned with the “balanced” versions of these problems:

1. Balanced Maximal Independent Set (BMIS): Given an undirected graph G , a BMIS is an MIS S in G that minimizes d_S . In other words, the BMIS has the minimum average degree (in G) among all MIS's in G .
2. Balanced Minimal Dominating Set (BMDS): Given an undirected graph G , a BMDS is an MDS D in G that minimizes d_D .
3. Balanced Minimal Vertex Cover (BMVC): Given an undirected graph G , a BMVC is an MVC C in G that minimizes d_C .

We note that the *maximum* independent set in a graph G – a well-studied NP-hard problem [5] – is not necessarily a BMIS in G . Consider the graph G that contains a complete graph K_p (assume p is even), and a complete bipartite graph $K_{A,B}$ with $|A| = 2$ and $|B| = 3$. Each vertex in A is connected to a different half of the set of vertices in K_p (i.e., one vertex of A is connected to one half of vertices of K_p and the second vertex of A is connected to the other half of K_p), and each vertex in B is connected to all vertices in K_p . Clearly, B is the maximum independent set in G and has average degree $p + 2$, while A is a BMIS in G since its average degree is $p/2 + 3$. Thus BMIS is quite different compared to the maximum independent set problem: in sharp contrast to the standard MIS, a maximum independent set may be very different from any BMIS.

1.2 Motivations: distributed and complexity-theoretic

One key motivation for our work is understanding the complexity of local computation of globally optimal (or near optimal) fundamental structures. The correctness of structures such as MIS or

MDS can be verified *strictly locally* by a distributed algorithm.¹ In the case of MIS, for example, each node can check the MIS property by communicating only with its neighbors; if there is a violation at least one node will raise an alarm. On the other hand, it is not difficult to show that the correctness of balanced structures such as BMIS cannot be locally verified (in the above sense) as the BMIS refers to a “global” property: nodes have to check the small average degree property, in addition to the MIS property. In fact, it can be shown that, for any D such that $D \in o(n)$, there is a graph of diameter D , where it takes at least $\Omega(D)$ rounds to check whether a given MIS is (a constant approximation of) a BMIS: Consider the graph consisting of a line $L = (u_1, \dots, u_D)$ and a cycle C of $(n - D)$ nodes, and assume that node u_D has an edge to each node in C . One way to select an MIS is to select nodes u_1, u_3, \dots, u_{D-1} and $\Theta(n - D)$ nodes from C : this yields an MIS S with constant average degree. On the other hand, the MIS S' formed by selecting $u_1, u_3, \dots, u_{D-3}, u_D$ has an average degree of $\Theta(n/D) \in \omega(1)$. For node u_1 , however, it takes D time to distinguish between S and S' .

Moreover, we prove that BMIS is an NP-hard problem and hence the optimality of the structure is not easy to check even in a centralized setting. A key issue that we address here is whether one can compute near-optimal local (distributed) solutions to balanced global structures such as BMIS. A main result of this paper is that despite the global nature, we can design efficient distributed algorithms that output high quality balanced structures.

Our work is also a step toward understanding the algorithmic complexity of a few basic balanced problems. While every MIS is an MDS, these two differ significantly in their balanced versions. In particular, we show that there exist graphs for which no MIS is a good BMDS. Hence we need a different approach to compute a good BMDS as compared to a good BMIS. Even for BMIS, we show that while one can (for example) use Luby’s algorithm [10] to efficiently compute an MIS, the same approach fails to compute a good quality BMIS. We present new algorithms for computing such balanced structures.

1.3 Practical Motivations

In distributed networks, especially in resource-constrained networks such as ad hoc networks, sensor and mobile networks, it is important to design structures that favor load balancing of tasks among nodes (belonging to the structure). This is crucial in extending the lifetime of the network (see e.g., [17] and the references therein). For example, in a typical application, an MIS (or an MDS) can be used to form clusters with low diameter, with the nodes in the MIS being the “clusterheads” [11]. Each clusterhead is responsible for monitoring the nodes that are adjacent to it. Having an MIS with low degree is useful in a resource/energy-constrained setting since the number of nodes monitored *per* node in the MIS will be low (on average). This can lead to better load balancing, and consequently less resource or energy consumption per node, which is crucial for ad hoc and sensor networks, and help in extending the lifetime of such networks while also leading to better fault-tolerance. For example, in an n -node star graph, the above requirements imply that it is better for the leaf nodes to form the MIS rather than the central node alone. In fact, the average degree of the MIS formed by the leaf nodes – which is 1 – is within a constant factor of the average degree of a star (which is close to 2), whereas the average degree, $n - 1$, of the MIS consisting of the central node alone is much larger.

¹ As is common in distributed verification (e.g., [4]), we require that all nodes output “yes” when given a valid instance; otherwise at least one node must output “no”.

Another proposed application of balanced structures is in the context of dynamic networks where one aims to *maintain* structures such as MISs efficiently even when nodes or links (edges) fail or change with time. Such changes are a feature of *ad hoc* and mobile networks due to mobility or failures. BMISs are a good candidate for maintaining an MIS efficiently in an incremental fashion: since the degrees of nodes in the MIS are balanced, this will lead to less overhead per insertion or deletion.

1.4 Our Results

Randomization is a vital component of our positive and negative results. We first note that the trivial lower bound is d for all balanced problems which follows from the example of a regular graph (where all nodes have the same degree). Hence, in general, the average degree of a balanced structure cannot be guaranteed to be less than d . On the other hand, there exist graphs where the average degree of the BMIS is significantly smaller than d (e.g., consider a graph in which $n/2$ nodes form a complete subgraph, with these nodes connected by a perfect matching to the remaining $n/2$). This leads us to two basic questions: (i) In every given graph G , does there always exist a BMIS whose average degree is at most d ? and (ii) Can question (i) be answered for a specific graph G in polynomial time? We answer both questions in the negative.

The well-known probabilistic proof of Turán’s theorem on independent sets [2, 16, 14] motivates question (i), and sheds light in an interesting way on the BMIS problem. Recall that in this probabilistic approach, we construct an independent set in G as follows: randomly permute the vertices, and construct an independent set I in which we put a vertex v iff no neighbor of v precedes v in the permutation constructed. Note that $P(v \in I) = 1/(d_v + 1)$. Thus, $E[|I|] = \sum_v 1/(d_v + 1)$, which is at least $n/(d + 1)$ by convexity; and, letting T denote the total degree (in G) of I , we have

$$E[T] = \sum_v \frac{d_v}{d_v + 1} < n.$$

Thus, heuristically “ $E \left[\frac{|T|}{|I|} \right] \leq O(d)$ ”; this is also true rigorously at least in the case where all degrees are small, in which case we can show that $|I|$ is concentrated around its mean (e.g., by a second-moment calculation). That is, there is an independent set I with $d_I = O(d)$. Note, however, that I is an independent set, not necessarily an MIS. Nevertheless, this argument appears to suggest that there is an MIS S with $d_S = O(d)$ for all graphs. Our theorems contradict this, show that “ $O(d^2)$ ” is the truth here instead of $O(d)$, and also develop good distributed versions of this result.

We show that unlike MIS, its balanced version, BMIS, is NP-hard. In particular, we show that the following *decision version* of the problem is NP-complete (cf. Theorem 8 in Section 2.3): “Given a graph G , is there an MIS in G with average degree at most d ?” In fact we show that the optimization version BMIS is quite hard to approximate in polynomial time: it cannot be approximated in polynomial time to within a factor of $\Omega(\sqrt{n})$ (cf. Theorem 9 in Appendix 2.4).

Henceforth, we focus on obtaining solutions for BMIS that are good *compared* to the average degree of the graph. We show that we can obtain near-tight solutions that compare well with d . The following are our main results:

Theorem 1. *There is a (centralized) algorithm that selects an MIS of average degree at most $d^2/8 + O(d)$ and runs in $O(nm^3 \log n)$ time with high probability.²*

² “With high probability” (w.h.p.) means with probability $\geq 1 - 1/n^{\Omega(1)}$.

To prove this theorem, we show that Luby’s MIS algorithm [10] returns an MIS with average degree at most $d^2/8 + O(d)$, albeit *with inverse polynomially small* probability. This can be easily turned into a *centralized* algorithm by repeating it a polynomial number of times until the desired bound is obtained; however, this does not yield a fast distributed algorithm. Our average-degree bound here is near-optimal, as we show an almost-matching lower bound (thereby also answering question (i) posed above in the negative):

Theorem 2. *For any real number $\alpha > 1$, there is a graph G with average degree at most α , but in which every MIS has average degree at least $\alpha^2/8 + 3\alpha/4 + 5/8$.*

We next consider *distributed* approximation algorithms for BMIS and show that we can output near-optimal solutions fast, i.e., solutions that are close to the lower bound. We consider the following standard model for our distributed algorithms where the given graph G represents a system of n nodes, with each node having a distinct ID [12]. Each node runs an instance of the distributed algorithm and the computation advances in synchronous *rounds*, where, in each round, nodes can communicate with their neighbors in G by sending messages of size $O(\log n)$. A node initially has only *local knowledge* limited to itself and its neighbors (it may however know n , the network size). We assume that local computation (performed by the node itself) is free as long it is polynomial in the network size. Each node u has local access to a special bit (initially 0) that indicates whether u is part of the output set. Our focus is on the *time complexity*, i.e., the number of rounds of the distributed computation.

We present two distributed algorithms for BMIS in Section 2.2, both running in polylogarithmic rounds; the second algorithm gives a better bound on the average degree at the cost of somewhat increased run time.

Theorem 3. *Consider a graph $G = (V, E)$ with average degree d .*

1. *There is a distributed algorithm that runs in $O(\log n \log \log n)$ rounds and with high probability outputs an MIS with average degree $O(d^2)$.*
2. *There is a distributed algorithm that runs in $\log^{2+o(1)} n$ rounds and with high probability outputs an MIS with average degree $(1 + o(1))(d^2/4 + d)$.*

Note that in general, due to the lower bound of Theorem 2, the bounds provided by algorithms of the above theorem are optimal up to constant factors.

We next present results on BMDS. Since an MIS is also an MDS, an algorithm for MIS can also be used to output an MDS. However, this can lead to a poor approximation guarantee, since there are graphs for which *every* MIS has a very large average degree compared to some MDS. This follows from the graph family used in the proof of Theorem 2: while the average degree of *every* MIS – of any graph in the family – is $\Omega(d^2)$, there exists an MDS with average degree only $O(d)$. Because an MIS is also an MDS, the results of Theorem 3 also hold for BMDS. Our next theorem shows that much better guarantees are possible for BMDS.

Theorem 4. *Any graph G with average degree d has a minimal dominating set with average degree at most $O(\frac{d \log d}{\log \log d})$. Furthermore, there is a sequential randomized algorithm for finding such an MDS in polynomial time w.h.p.*

The next theorem shows that the bound of Theorem 4 is optimal in general up to constant factors:

Theorem 5. For any real number $\alpha > 0$, there are graphs with average degree $\leq \alpha$, but for which any MDS has an average degree of $\Omega(\frac{\alpha \log \alpha}{\log \log \alpha})$.

Finally, we show that there cannot be any bounded approximation algorithm for BMVC:

Theorem 6. For any real number $\alpha > 2$, there are graphs for which the average degree is at most α , but for which the average degree of any MVC is arbitrarily large.

2 Balanced Maximal Independent Set

We first prove Theorem 2 which shows that there are graphs G for which the degree of every MIS is much larger than d . More importantly, the theorem gives a lower bound on the quality of BMIS in general: one cannot guarantee an MIS whose average degree is less than $\frac{d^2}{8} + \Theta(d)$.

Theorem 2. For any real number $\alpha > 1$, there is a graph G with average degree at most α , but in which every MIS has average degree at least $\alpha^2/8 + 3\alpha/4 + 5/8$.

Proof. Consider the graph consisting of a copies of K_b , as well one copy of $K_{c,c}$, where $b = \lfloor \frac{3+\alpha}{2} \rfloor$ and $c = \lfloor \frac{1}{2} \sqrt{2ab(\alpha - b + 1) + \alpha^2 + \alpha} \rfloor$.

The resulting graph has average degree $\frac{ab(b-1)+2c^2}{ab+2c} \leq \alpha$. Every MIS of this graph contains one vertex from each K_b , as well as one half of the vertices of $K_{c,c}$, for an average degree of $\frac{ab+c^2}{a+c}$. As a tends to infinity, such average degree increasingly approaches $\frac{(3+\alpha-b)b}{2} \geq \alpha^2/8 + 3\alpha/4 + 5/8$.

2.1 An Almost-Optimal Sequential Randomized BMIS Algorithm

In this section, we prove Theorem 1. To do so, we use Luby's algorithm for MIS [10], which can be described as follows. Every vertex v choose a rank ρ_v uniformly and independently from the real interval $[0, 1]$.³ Any vertex whose rank is lower than all its neighbors is then selected for the independent set. Such vertices and their neighbors are removed from the graph, and this process is repeated until an MIS is formed. We show that if the original graph has average degree d , then Luby's algorithm may select an MIS of average degree $\approx d^2/8$, albeit with a polynomially small probability. By checking and repeating the algorithm a polynomial number of times we can obtain an algorithm that outputs such a MIS (w.h.p.). This can be done easily in a sequential setting. Theorem 2 shows that the average degree bound obtained is close to optimal.

Lemma 1. With probability $\Omega(d^{-2}n^{-3})$, Luby's algorithm selects an MIS of average degree $\leq \frac{d^2}{8} + \frac{7d}{4}$. Let $\epsilon > 0$ be any constant. With probability $\Omega(d^{-2}n^{-1})$, Luby's algorithm selects an MIS of average degree $\leq \frac{d^2}{8} + (\frac{7}{4} + \epsilon)d$.

Proof. Let t be the target degree of the MIS we are trying to construct. In this case, either $t = \frac{d^2}{8} + \frac{7d}{4}$ or $t = \frac{d^2}{8} + (\frac{7}{4} + \epsilon)d$. We say a vertex v is *big* if $d_v > t$, otherwise it is *small*. Roughly speaking, our goal is to choose an MIS of degree $\leq t$, and for this end adding small vertices helps us while adding big vertices hurts us.

For each vertex v , let x_v (respectively y_v) denote the number of small (respectively big) neighbors, so $x_v + y_v = d_v$. Now, suppose we run Luby's algorithm completely, and let \tilde{M}, \tilde{N} be the

³ One can also assume that ranks are chosen uniformly from the integer interval $\{1, 2, \dots, n^4\}$, where n is the number of nodes.

random variables that are the number of edges and vertices in the resulting MIS I . Define the random variable $S_t = \tilde{M} - t\tilde{N}$. The MIS I will have degree $\leq t$ iff $S_t \leq 0$. We have

$$\begin{aligned} E[S_t] &= \sum_v (d_v - t)P(v \in I) \leq \sum_{\text{small } v} (d_v - t)P(v \text{ selected in 1st round}) + \sum_{\text{big } v \in I} y_v \\ &\quad + \sum_{\text{big } v} \max(0, (x_v - t))P(v \text{ available after 1st round}). \end{aligned}$$

Let us consider these terms in turn. The probability that a small vertex v is selected in the first round is $\frac{1}{d_v+1}$.

Note that any edge whose endpoints are both big vertices can appear only once in the MIS, hence $\sum_{\text{big } v \in I} y_v \leq \sum_{\text{big } v} y_v/2$.

To upper-bound $(x_v - t)P(v \text{ available after 1st round})$, consider all the small neighbors of v . Suppose that, for a small $u \in N(v)$, the rank of vertex u is smaller than all its neighbors (including v), as well as all the other small neighbors of v . This occurs with probability $\frac{1}{d_u+x_v}$. If this event occurs, then vertex u is selected and vertex v is unavailable for I . Furthermore, for each of the small neighbors of v , the corresponding events are mutually exclusive. Hence we get $P(v \text{ available after 1st round}) \leq 1 - \sum_{\text{small } u \in N(v)} \frac{1}{d_u+x_v} \leq 1 - \sum_{\text{small } u \in N(v)} \frac{1}{t+x_v} \leq \frac{t}{t+x_v}$. Hence, we obtain the following bound on $E[S_t]$:

$$\begin{aligned} E[S_t] &\leq \sum_{\text{small } v} \frac{d_v - t}{d_v + 1} + \sum_{\text{big } v} \left(y_v/2 + \max(0, (x_v - t) \frac{t}{t + x_v}) \right) \\ &\leq \sum_{\text{small } v} \frac{d_v - t}{d_v + 1} + \sum_{\text{big } v} (y_v/2 + x_v/2) \leq \sum_{\text{small } v} \frac{d_v - t}{d_v + 1} + \sum_{\text{big } v} d_v/2 \end{aligned}$$

Now suppose there are n_S small vertices and n_B large vertices, with average degrees d_S and d_B respectively. We have $n_S + n_B = n$ and $n_S d_S + n_B d_B = dn$. By concavity we have

$$E[S_t] \leq n \frac{d_B d_S (d + 1 - d_S) - 2t(d_B - d) + d(d_B - 2d_S)}{2(d_S + 1)(d_B - d_S)}.$$

Routine calculus shows that this achieves its maximum value at $d_B = t$ and $d_S = \frac{\sqrt{2t\sqrt{t^2+2t+1}-3t}}{t-2}$, yielding

$$E[S_t] \leq n \frac{2\sqrt{2t}(d-t) + t(d-1) + 2d}{2(t+1)}.$$

For $t = d^2/8 + 7d/4$, the resulting expression is negative for all $d \geq 1$. For $t = d^2/8 + (7/4 + \epsilon)d$, the resulting expression is bounded uniformly below $-\Omega(1)$ for all $d \geq 1$.

We have shown that the random variable S_t has mean $E[S_t] \leq 0$ (respectively $E[S_t] \leq -\Omega(1)$). Furthermore, S_t is always a rational number with denominator at most $8n^2$, and $S_t \geq -d^2n$. This implies that, with probability at least $\Omega(d^{-2}n^{-3})$, we must have $S_t \leq 0$. When this occurs, resulting MIS has average degree $\leq t = \frac{d^2}{8} + \frac{7d}{4}$ as desired. A similar argument applies to $t = \frac{d^2}{8} + (7/4 + \epsilon)d$. \square

Theorem 1. *There is a (centralized) algorithm that selects an MIS of average degree at most $d^2/8 + O(d)$ and runs in $O(nm^3 \log n)$ time with high probability.*

Proof. By repeating Luby's algorithm for $O(n^3 d^2)$ iterations we find an MIS with average degree $\leq \frac{d^2}{8} + \frac{7d}{4}$ w.h.p. As each iteration can be implemented in $O(m \log n)$ time, the total work expended is $O(nm^3 \log n)$ w.h.p. If we are willing to accept an MIS of slightly higher degree, i.e., $\leq \frac{d^2}{8} + (\frac{7}{4} + \epsilon)d$, this can be reduced to $O(\frac{m^3}{n} \log n)$.

2.2 Distributed Algorithms for BMIS

This section is devoted for designing different distributed algorithms for BMIS. In particular we will prove Theorem 3 (Parts 1, and 2). The proposed algorithms do not require any global information of the original graph other than n .

Theorem 3. *Consider a graph $G = (V, E)$ with average degree d .*

1. *There is a distributed algorithm that runs in $O(\log n \log \log n)$ rounds and with high probability outputs an MIS with average degree $O(d^2)$.*
2. *There is a distributed algorithm that runs in $\log^{2+o(1)} n$ rounds and with high probability outputs an MIS with average degree $(1 + o(1))(d^2/4 + d)$.*

Proof of Part 1 of Theorem 3. We propose a distributed algorithm that constructs an MIS I of G such that the following two properties hold with high probability: (a) I has average degree at most $O(d^2)$, and (b) I is constructed within $O(\log n \times \log \log n)$ rounds.

This algorithm does not require any global information of the original graph, other than the network size n . The algorithm depends on a parameter ϕ , which is held constant. For any constant $c > 0$, one can choose ϕ appropriately so that the distributed algorithms succeeds with probability $1 - n^{-c}$; the parameter ϕ will only affect the running time by a constant factor. This is the strongest form in which an algorithm can be said to succeed with high probability.

The algorithm has three phases, which are intended to address the cases where $d \leq O(\sqrt{n/\log n})$, $\Theta(\sqrt{n/\log n}) \leq d \leq \Theta(\sqrt{n})$, and $d \geq \Omega(\sqrt{n})$ respectively. The first phase runs Luby's algorithm for MIS on the vertices with degree $\leq \sqrt{n/\log n}$. The next phase gradually extends the resulting independent set by finding MIS's of the subgraphs consisting of successively larger degrees. Finally, using Luby's algorithm, this is extended to an MIS of G itself. It is easy to see that this leads to an MIS of G , and the resulting algorithm runs in $O(\log n \times \log \log n)$ rounds. We will also show that if we run only Phases I and III of this algorithm, then we can obtain an MIS of average degree $O(d^2 \log d)$ in time $O(\log n)$.

We introduce the following definition which will be used throughout the proof. For any real number s , we let G_s denote the subgraph of G induced on the vertices of degree $\leq s$. This notation is used in describing Algorithm 1.

The following basic principle will be used in a variety of places in this proof:

Proposition 1. *Suppose a graph G has n vertices and average degree d . Suppose $s > 1$. Then the subgraph G_{sd} contains at least $n(1 - 1/s)$ vertices.*

Proof. Note that $\sum_v d_v = nd$. Suppose that G_{sd} has less than $n(1 - 1/s)$ vertices. Then there are more than n/s vertices with degree larger than sd . They contribute more than $sd \cdot n/s$ to the sum $\sum_v d_v$, which is a contradiction. \square

We now show that this algorithm has good behavior in the first two parameter regimes. The third regime $d = \Omega(\sqrt{n})$ is trivial.

First Phase – Luby’s algorithm on $G_{\phi\sqrt{n/\log n}}$:(Recall that G_x denotes the subgraph of vertices of degree $\leq x$.)

1. Each vertex v in $G_{\phi\sqrt{n/\log n}}$ marks itself independently with probability $1/(2d_v)$.
2. If two adjacent nodes are marked, unmark the one with higher degree (breaking ties arbitrarily).
3. Add any marked nodes to the independent set I .

Second Phase – Extending the MIS:

1. For $i = 0, 1, \dots, \frac{1}{2} \lceil \log_2 \log n - \log_2 \phi \rceil$, repeat the following:
2. Let $x_i = 2^i \phi \sqrt{n/\log n}$. Run Luby’s MIS algorithm for $\phi \log n$ iterations to extend the current independent set I to an MIS of the graph G_{x_i} .

Third Phase:

1. Using Luby’s algorithm, extend the current independent set I to an MIS of G .

Algorithm 1: Distributed Algorithm for Approximating BMIS.

Lemma 2 (First Phase). *Suppose $d \leq \frac{\phi}{2} \sqrt{n/\log n}$. Then with probability $1 - n^{-\Omega(1/\phi)}$, the independent set produced at the end of the first phase, contains $\Omega(n/d)$ vertices. In particular, the final MIS produced has average degree $O(d^2)$.*

Proof. Let n', d' denote the number of vertices and average degree of the graph $G_{\phi\sqrt{n/\log n}}$. Note that $d' \leq d$. By Proposition 1 we have $n' \geq n/2$.

For each vertex $v \in G_{\phi\sqrt{n/\log n}}$ let X_v be the random variable indicating that v was marked, and X'_v the random variable indicating that v was accepted into I (i.e. it did not conflict with a higher-degree vertex). Let Y denote the size of this independent set, i.e., $Y = \sum_{v \in G_{\phi\sqrt{n/\log n}}} X'_v$. As shown in [10], we have $E[Y] = \Omega(n'/d') = \Omega(n/d)$. Now, we want to show that Y is concentrated around its mean. As described in [6], this sum can be viewed as a “read- k family”; each variable X'_v is a Boolean function of the underlying independent variables X , and each variable X_v affects at most $k = O(\phi\sqrt{n/\log n})$ of the Boolean functions. Hence this sum obeys a similar concentration bound to the Chernoff bound, albeit with the exponent divided by k . In particular, the probability that Y deviates below a constant factor from its mean is given by $P(Y \leq (1-x)E[Y]) \leq e^{-\frac{E[Y]x^2}{2k}} \leq e^{-\Omega(\frac{\log n}{\phi^2})} = n^{-\Omega(1/\phi^2)}$. Hence, with high probability, the total number of vertices returned from the first phase is $\Omega(n/d)$ as desired. \square

We next show that the second phase has the appropriate behavior:

Lemma 3 (Second Phase). *Suppose $\frac{\phi}{2} \sqrt{n/\log n} < d \leq \frac{1}{2} \sqrt{n}$. Then with probability $1 - n^{-\Omega(1/\phi)}$, the independent set produced at the end of the second phase, contains $\Omega(n/d)$ vertices. In particular, the final MIS produced has average degree $O(d^2)$.*

Proof. Let n', d' represent the number of vertices and average degree of $G_{\sqrt{n}}$. By Proposition 1 we must have $n' = \Omega(n)$ and $d' \leq d$.

If $d' \leq \frac{\phi}{2} \sqrt{n/\log n}$, then by Proposition 1 there would be $\Omega(n)$ vertices of G with degree $\leq \frac{\phi}{2} \sqrt{n/\log n}$. By Lemma 2, phase 1 would then produce an independent set with $\Omega(n/\sqrt{n/\log n}) = \Omega(n/d)$ vertices.

So suppose $d' > \frac{\phi}{2} \sqrt{n/\log n}$. Now, as i increases, x_i is multiplied by a factor of 2 as it increases from $\phi\sqrt{n/\log n}$ to \sqrt{n} . In particular, there is some value of i which has $2d' \leq x_i \leq 4d'$. At this point, the standard analysis shows that $\phi \log n$ iterations of Luby’s algorithm produces, with probability $1 - n^{-\Omega(1/\phi)}$, an MIS of the graph G_{x_i} . By Proposition 1, G_{x_i} contains $\Omega(n')$ vertices.

- 1: Let $\phi > 1$ be a fixed parameter. Initialize $I = \emptyset$.
- 2: **for** $i = 0, \dots, \lceil \log_\phi n \rceil$ **do**
- 3: Using any distributed MIS algorithm, extend I to an MIS of the graph G_{ϕ^i} .
- 4: Return the final MIS I .

Algorithm 2: Greedy Distributed Approximation Algorithm for BMIS.

Furthermore, *any* MIS of G' must contain $\Omega(n'/d')$ vertices; the reason for this is that the maximum degree of any vertex in G_{x_i} is $O(d')$, and it is necessary to select $\Omega(n'/d')$ simply to ensure that every vertex is covered by the MIS.

Now, at stage i we produce an MIS of G_{x_i} which contains $\Omega(n/d)$ vertices. This is eventually extended to an MIS of G with $\Omega(n/d)$ vertices. \square

Proof of Part 2 of Theorem 3. The greedy algorithm for BMIS is very simple. We label the vertices in order of increasing degree (breaking ties arbitrarily). Each vertex is added to the IS (Initially, $IS = \emptyset$), unless it was adjacent to a earlier vertex already selected. This is a simple deterministic algorithm which requires time $O(m)$.

Theorem 7. *The greedy algorithm produces an MIS of degree at most $\frac{d^2}{4} + d$.*

Proof. Order the vertices in order of increasing degree $d_1 \leq d_2 \leq \dots \leq d_n$. Define the indicator variable x_v to be 1 if $v \in I$ and 0 otherwise, where I is the MIS produced. For any pair of vertices u and v with $d_u \geq d_v$, we also define the indicator y_{vu} to be 1 if $v \in I$ and there is an edge from v to u . (It may seem strange to include the variable y_{vv} , as we always have $y_{vv} = 0$ in the intended solution, but this will be crucial in our proof, which is based on LP relaxation.)

As the greedy algorithm selects v iff no earlier vertex was adjacent to it, we have $x_v = 1$ if and only if $y_{1v} = y_{2v} = \dots = y_{v-1,v} = 0$. In particular, x_v satisfies the linear constraint $x_v \geq 1 - y_{1v} - y_{2v} - \dots - y_{vv}$. The variables x, y also clearly satisfy the linear constraints $\forall v: 0 \leq x_v \leq 1$, $\forall v \leq u: 0 \leq y_{vu}$, and $\forall v: \sum_u y_{vu} \leq d_v x_v$. which we refer to as the *core constraints*. The final MIS contains $\sum_v x_v$ vertices and $\sum_v d_v x_v$ edges, and hence the average degree of the resulting MIS is $d_I = \sum_v d_v x_v / \sum_v x_v$.

We wish to find an upper bound on the ratio $R = \frac{\sum_v d_v x_v}{\sum_v x_v}$. The variables x, y satisfy many other linear and non-linear constraints, and in particular are forced to be integral. However, we will show that the core constraints are sufficient to bound R . The way we will prove this is to explicitly construct a solution x, y which satisfies the core constraints and maximizes R subject to them, and then show that the resulting x, y still satisfies $R \leq \frac{d^2}{4} + d$.

Let x, y be real vectors which maximizes R among all real vectors satisfying the core constraints, and among all such vectors, which minimize $\sum_{u>v} y_{vu}$. Suppose $y_{vu} > 0$ for some $u > v$. If $x_u = 1$, then we simply decrement y_{vu} by ϵ . The constraint $x_u \geq 1 - y_{1u} - \dots - y_{uu}$ clearly remains satisfied as $x_u = 1$, and all other constraints are unaffected. The objective function is also unchanged. However, this reduces $\sum_{u>v} y_{vu}$, contradicting maximality of x, y .

Suppose $y_{vu} > 0$ for some $u > v$, and $x_u < 1$ strictly. Note that $y_{vu} \leq d_v x_v$, so we must have $x_v > 0$ strictly. For some sufficiently small ϵ , we change x, y as follows: $y'_{vu} = y_{vu} - \epsilon$, $y'_{vv} = y_{vv} + \frac{\epsilon}{d_v + 1}$, $x'_v = x_v - \frac{\epsilon}{d_v + 1}$, $x'_u = x_u + \frac{\epsilon}{d_u + 1}$, and $y'_{uu} = y_{uu} + \frac{\epsilon d_u}{d_u + 1}$. All other values remain unchanged. We claim that the constraints on x, y are still preserved. Furthermore, the numerator of R increases and the denominator decreases; hence $R' \geq R$. This contradicts the maximality of x, y .

In summary, we can assume $y_{vu} = 0$ for all $u > v$. In this case, the core constraints on v become simply $1 - y_{vv} \leq x_v \leq 1$ and $y_{vv} \leq d_v x_v$.

It is a simple exercise to maximize R subject to these constraints (every vertex operates completely independently). The maximum is achieved by a solution which has the form, for some $t > 0$, of $x_v = \frac{1}{d_v+1}$ for $d_v \leq t$, and $x_v = 1$ for $d_v > t$. In this case, the objective function $R(x)$ satisfies

$$R \leq \frac{\sum_{d_v \leq t} \frac{d_v}{d_v+1} + \sum_{d_v > t} d_v}{\sum_{d_v \leq t} \frac{1}{d_v+1} + \sum_{d_v > t} 1}$$

Let d_S, d_B denote the average degrees of the vertices of degree $\leq t, > t$ respectively, and let n_S, n_B represent the number of such vertices. Then by concavity, we have

$$R \leq \frac{n_S \frac{d_S}{d_S+1} + n_B d_B}{\frac{n_S}{d_S+1} + n_B} \leq \frac{d(d_B - d_S) + d_B d_S (d - d_S)}{d_S (d - d_S) + (d_B - d_S)}$$

Routine calculus shows that this achieves its maximum value at $d_B = \infty$ and $d_S = d/2$, yielding $R \leq d^2/4 + d$ as claimed. \square

This greedy algorithm can be converted, with only a little loss, to a distributed algorithm as shown in Algorithm 2. This algorithm is basically the sequential greedy algorithm, except we are quantizing the degrees to multiples of some parameter ϕ . Allowing $\phi \rightarrow 1$ sufficiently slowly, we obtain an algorithm which requires $\log^{2+o(1)} n$ rounds and returns an MIS of average degree at most $(1 + o(1))(d^2/4 + d)$ w.h.p. As we have seen in Theorem 2, this is within a factor of 2 of the lowest degree possible.

The following lemma shows that the above analysis of the greedy algorithm is essentially tight.

Lemma 4. *For all real numbers $\alpha > 0$, there are graphs of average degree $\leq \alpha$, and for which the greedy algorithm produces an MIS of average degree at least $\alpha^2/4 + \alpha - 1$.*

Proof. Define the following graph G which contains three groups of vertices A, B, C . We have $|A| = a, |B| = b - 1$, and $|C| = ab$. Each A -vertex connects to b C -vertices. Each C -vertex connects to all $b - 1$ B -vertices.

The graph G contains $a + ab + b - 1$ vertices and ab^2 edges, and has an average degree of $d = \frac{2ab^2}{a+ab+b-1}$.

Now, the vertices in A, C have degree b , while the vertices in B have degree ab . Suppose that the greedy algorithm selects the A -vertices (they are tied with the C -vertices). It then selects all B -vertices, and hence, the resulting MIS has $a + b - 1$ vertices and ab^2 edges. Now set $b = \left\lfloor \frac{(a+1)\alpha + \sqrt{(a+1)^2\alpha^2 + 8(a-1)a\alpha}}{4a} \right\rfloor$. As a tends to infinity, we have $d \leq \alpha$, while the degree of the resulting MIS approaches a value that is at least $\alpha^2/4 + \alpha - 1$. \square

2.3 NP-Completeness of the Decision Version of BMIS

We show NP-completeness of the decision version of BMIS (cf. Sec. 1.1) by reducing a variant of the 3-SAT problem to the BMIS problem and vice versa.

A Boolean formula in conjunctive normal form is called a (k, s) -formula if every clause contains exactly k distinct variables and every variable occurs in at most s clauses. A (k, s) -formula is called

a $(k, =s)$ -formula if every variable occurs in exactly s clauses. Let (k, s) -SAT (resp., $(k, =s)$ -SAT) denote the satisfiability problem restricted to (k, s) -formula (resp., $(k, =s)$ -formula). Kratochvil et al. [8] proved that the $(k, =s)$ -SAT problem is NP-complete for every $k \geq 3$ and $s \geq 4$. We now show how to reduce an instance of the $(3, =4)$ -SAT problem to an instance of BMIS.

In the $(3, =4)$ -SAT problem, the input is a set of clauses, each of them with 3 variables and each variable occurs in exactly 4 clauses and the aim is to find a satisfying truth assignment to the whole $(3, =4)$ -formula. To form a satisfying truth assignment we must pick one literal from each clause and give it the value TRUE. But our choices must be consistent, namely, if we choose a variable x in one clause, we cannot choose the negation of x in another. Any consistent choice of literals, one from each clause, specifies a truth assignment. Recall that for the BMIS problem, we are given a graph G and we want to know whether G contains MIS with average degree at most that of the graph. We relate the above two problems as follows.

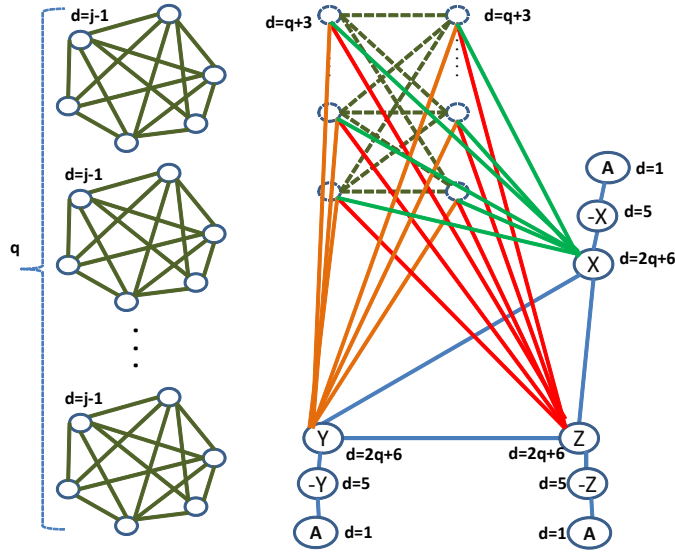


Fig. 1. Graph construction corresponding to clause $(x \vee y \vee z)$.

BMIS Graph Construction Given a $(3, =4)$ -formula $F = A_1 \wedge A_2 \wedge \dots \wedge A_k$, we construct the following graph G_F , which we simply denote as G if F is clear from the context. For each clause, say $A_i = (x \vee y \vee z)$, in F , we construct the following corresponding *clause component*. Construct a triangle with vertices labeled x , y , and z , and for each *triangle vertex*, add an additional neighbor labeled the negation of the corresponding vertex, and for each of these negations, add another neighbor (of degree 1). Construct a single copy of $K_{q,q}$ and q copies of K_j , and connect each of the triangle vertices with all vertices in $K_{q,q}$ (see Fig. 1). Finally, we connect each labeled vertex in each clause component with its negation vertex in all other clause components, and hence, any MIS in G cannot contain vertices labeled both x and $-x$ for all variables x in F . This construction is shown in Figure 1. We will see in the proof of Theorem 8 how to obtain a mapping between truth assignments of the literals in F and maximal independent sets on G_F .

Clearly, this construction takes polynomial time and each of the triangle vertices has degree $2q + 6$, whereas each of the negation vertices has degree 5, and each neighbor of these negation vertices has degree 1.

Lemma 5. *Let F be a $(3, =4)$ -formula and consider the corresponding graph G with average degree d . Consider any MIS S in G . Then $d_S \leq d$ iff S contains one triangle vertex from every clause component of G .*

Proof. The average degree of one component in G is $(qj(j-1) + 2q(q+3) + 3(2q+6) + 18)/(qj + 2q + 3 + 6)$. The average degree of an MIS (in one component) containing one of the triangle vertices and the three degree 1 vertices is $(q(j-1) + (2q+6) + 3)/(q+1+3)$. On the other hand, the average degree of any MIS that does not contain one of the triangle vertices is at least $(q(j-1) + q(q+3) + 3)/(q+q+3)$. Therefore, the ratio of the average degree of any MIS in one component that does not contain one of the triangle vertices to the average degree of this component tends to $(j+2)/4$ as q tends to infinity.

Now consider the whole graph G . Clearly, the average degree of G is the average degree of each component scaled appropriately by the number of clauses in F , denoted by k . Now, assume that i of the k clauses are not satisfiable (hence $k-i$ clauses are satisfiable). Then, the average degree of any MIS in G is at least

$$\frac{(k-i)(q(j-1) + (2q+6) + 3) + i(q(j-1) + q(q+3) + 3)}{(k-i)(q+1+3) + i(q+q+3)}.$$

Therefore, the ratio of the average degree of any MIS in G to the average degree of G tends to $i(j+2)/(2(k+i))$ as q tends to infinity. That is, by choosing j large enough with respect to k , the above ratio will be greater than 1.

Theorem 8. *The following decision problem is NP-complete: Given a graph G , is there an MIS in G with average degree at most that of the graph?*

Proof. We proceed by giving reductions from $(3, =4)$ -SAT to BMIS and vice versa.

$(3, =4)$ -SAT \rightarrow BMIS: We need to show that if a $(3, =4)$ -formula F has a satisfying assignment, then there exists an MIS S in G with $d_S \leq d$.

Assume that the $(3, =4)$ -formula F has a satisfying assignment. For each clause in F , we pick any literal whose value under a satisfying assignment is **TRUE** (there must be at least one such literal), and add the corresponding vertex in the component corresponding to that clause to S . We then add, from each component, all degree one vertices and one vertex from each complete graph K_q to S . Clearly, S yields an independent set that is maximal. Moreover, since S contains one of the triangle vertices in each component, Lemma 5 tells us that the average degree of S is at most that of G .

BMIS \rightarrow $(3, =4)$ -SAT: We want to show that if there exists an MIS S in G with an average degree of at most that of the graph G , then the corresponding $(3, =4)$ -formula F has a satisfying assignment.

By Lemma 5, if S has $d_S \leq d$, then S includes, for each clause component of G , one of the triangle vertices, all degree one vertices, and one vertex from each complete graph K_q .

To obtain a satisfying assignment of F , we assign x a value of **TRUE** if S contains a vertex labeled x , and a value of **FALSE** if S contains a vertex labeled $-x$ (if S contains neither, we assign an arbitrary truth value to x). Clearly this yields a truth assignment that satisfies all clauses in F .

- 1: Mark each vertex of degree $> 2d$ independently with prob. $\frac{\log t}{t}$ where $t = \frac{2d \log d}{\log \log d}$.
- 2: Mark every vertex of degree $\leq 2d$.
- 3: If any vertex v is not marked, and none of the neighbors of v are marked, then mark v .
- 4: Let M denote the set of marked vertices at this point. M forms a dominating set of G , but is not necessarily minimal. Using any algorithm, select a minimal dominating set $M' \subseteq M$.
- 5: Check if $d_{M'} \leq t$. If so, return M' . Otherwise, return FAIL.

Algorithm 3: Approximation Algorithm for BMDS.

2.4 Hardness of Approximating BMIS

Theorem 9. *The BMIS problem on a graph with n vertices cannot be approximated to within a factor of $(3 + \sqrt{8n - 7})/12$ in polynomial time.*

Proof. Given a graph H with p vertices, we construct a graph G with $n = 2p^2 + p + 1$ vertices as follows. We first construct a complete bipartite graph $K_{A,B}$ such that $|A| = p^2 + 1$ and $|B| = p^2$, and then attach H to $K_{A,B}$ by connecting each vertex in H with p vertices in A such that each vertex in A has at most one neighbor in H . Assume that the size of the maximum independent set of H is at least $p^{1-\epsilon}$, where $\epsilon = O(1/\sqrt{\log \log p})$ (see for example the graph in [3]). It is well known that the maximum independent set problem on graphs with p vertices cannot be approximated within $p^{1-\epsilon}$ in polynomial time [3]. This implies that there is no approximation algorithm that guarantees an approximated solution for the maximum independent set problem on H of more than p^ϵ vertices (since the size of the maximum independent set is at most p). We now convert the graph H to a p -regular graph by adding an appropriate number of self loops to each vertex in H . We observe that the above inapproximability result on the maximum independent set problem on H still holds for the resulting p -regular graph H . Thus, each vertex of the graph H has degree $2p$, one vertex in A has degree p^2 (the vertex with no neighbors in H), and each of the remaining vertices of $A \cup B$ has degree $p^2 + 1$. Clearly, the optimal solution for BMIS on G includes the maximum independent set on H , and has average degree at most $(2p^{2-\epsilon} + (p^2 - p^{2-\epsilon})(p^2 + 1) + p^2)/(p^{1-\epsilon} + p^2 - p^{2-\epsilon} + 1)$. On the other hand, the average degree of the approximated polynomial time solution of BMIS is at least $(2p^{1+\epsilon} + (p^2 + 1 - p^{1+\epsilon})(p^2 + 1))/(p^\epsilon + p^2 + 1 - p^{1+\epsilon})$. Note that $\epsilon = O(1/\sqrt{\log \log p})$ tends to 0 for sufficiently large p , and hence the average degrees of the optimal BMIS and the approximated BMIS tend to $3p^2/(p + 1)$ and $p^2 + p/(p^2 - p + 2)$, respectively. Therefore, the approximated BMIS is at least $(p + 1)/3 + (p + 1)/3p(p^2 - p + 2) \geq (p + 1)/3$ times the optimal, which proves the theorem.

3 Balanced Minimal Dominating Set

For arbitrary graphs, we turn our attention to designing algorithms for finding approximate solutions to BMDS. Since any MIS in a given graph G is also an MDS in G , all algorithms designed for BMIS also return an BMDS in G of the same average degree. Thus, we have the same bounds (and distributed algorithms) corresponding to those in Section 2. However, for BMDS, better bounds are possible. Given a graph with average degree d , we will present a polynomial-time algorithm that finds an MDS of average degree $O(\frac{d \log d}{\log \log d})$. We will also construct a family of graphs G for which every MDS has average degree $\Omega(\frac{d \log d}{\log \log d})$.

Theorem 4. *Any graph G with average degree d has a minimal dominating set with average degree at most $O(\frac{d \log d}{\log \log d})$. Furthermore, there is a sequential randomized algorithm for finding such an MDS in polynomial time w.h.p.*

Proof. Let d_v denote the degree of vertex v in G for notational convenience. For a target degree t , and any set of vertices V_0 , we define $S_t^{V_0} = \sum_{v \in V_0} (d_v - t)$. Our goal is to find an MDS X with $S_t^X \leq 0$, for some $t = O(d \log d / \log \log d)$.

Let $x = 2d$ and divide the vertices into three classes: A , the set of vertices of degree $\leq x$; B , the set of vertices of degree $> x$, which have at least one neighbor in A ; and C , the set of vertices of degree $> x$, all of whose neighbors are in B or C . Mark each vertex in $B \cup C$ with probability $p = \frac{\log t}{t}$. Next, form the set $Y \subseteq B \cup C$, by inserting all marked vertices in $B \cup C$ and vertices in C with no marked neighbors. Clearly Y dominates C , and $A \cup Y$ dominates G . Now, select two subsets $A' \subseteq A$ and $Y' \subseteq Y$ such that $X = A' \cup Y'$ is an MDS of G .

We first examine $S_t^{Y'}$. Any vertex of G with degree $\leq t$ contributes at most 0 to $S_t^{Y'}$. Therefore, suppose v has degree $> t$. If $v \in B$, it is selected for Y with probability at most $\frac{\log t}{t}$. If $v \in C$, all its neighbors are marked with probability $\frac{\log t}{t}$, so it is selected for Y with probability at most $\frac{\log t}{t} + (1 - \frac{\log t}{t})^t \leq 2\frac{\log t}{t}$. Hence the expected contribution of such vertex to $S_t^{Y'}$ is at most $2d_v \frac{\log t}{t}$. Summing over all such vertices, we have $E[S_t^{Y'}] \leq 2|B \cup C|d_{B \cup C} \frac{\log t}{t}$, where $d_{B \cup C}$ denotes the average degree of vertices in $B \cup C$.

Now, some of the vertices in A are dominated by B -vertices of Y' . Let A_0 be the set of vertices *not dominated* by Y' . These vertices can only be dominated by vertices of A' , so we must have $|A'|(|d_{A'} + 1|) \geq |A_0|$. Subject to the conditions $|A'|(|d_{A'} + 1|) \geq |A_0|$ and $d_{A'} \leq x$, we have $S_t^{A'} \leq \frac{|A_0|(x-t)}{x+1}$. The ultimate MDS may contain vertices in $A - A_0$ as well; however, as $x \leq t$, these will have a negative contribution to S_t , and hence they will only help us in showing an upper bound on S_t .

Consider the expected size of A_0 . A vertex $v \in A$ lies in A_0 if none of its neighbors are marked (this is not a necessary condition), and vertices are marked independently with probability p . Hence $E[|A_0|] \geq \sum_{v \in A} (1-p)^{d_v} \geq |A|(1-p)^{d_A}$. Putting all this together, we have that the final MDS $X = A' \cup Y'$ satisfies $E[S_t^X] \leq 2p|B \cup C|d_{B \cup C} + |A| \frac{x-t}{x+1} (1-p)^{d_A}$. For d sufficiently large, p approaches zero, so that that $(1-p)^{d_A} \leq e^{-2pd_A}$.

We know that $|A| + |B| + |C| = n$, and $|A|d_A + (|B| + |C|)d_{B \cup C} \leq nd$. Eliminating $|A|, |B|, |C|$ we have

$$E[S_t^X] \leq n \left(\frac{2d_{B \cup C}(d - d_A) \log t}{t(d_{B \cup C} - d_A)} - \frac{(d_{B \cup C} - d)(t - 2d)t^{-\frac{d_A}{t}}}{(2d + 1)(d_{B \cup C} - d_A)} \right)$$

Routine calculus shows that, for t sufficiently large, this achieves its maximum value at $d_{B \cup C} \rightarrow \infty$ and $d_A = \frac{t \log(\frac{t-2d}{4d+2})}{\log t}$, yielding

$$E[S_t^X] \leq \frac{2d \log t}{t} - 2 \log \left(\frac{t - 2d}{4d + 2} \right) - 2.$$

For $t = 2d \log d / \log \log d$, the RHS approaches $-\infty$ as $d \rightarrow \infty$. This implies that $E[S_t^X] \leq 0$, so there is a positive probability of selecting an MDS of average degree $\leq t$.

Note that for $t = 2d \log d / \log \log d$, we have $S_t^X \geq -dn \log^{O(1)} n$ and $E[S_t^X] \leq -\Omega(1)$. By Markov's inequality, the random variable S_t^X is negative with probability $\geq d^{-1}n^{-1} \log^{-O(1)} n$. Hence, after $dn \log^{O(1)} n$ iterations of this sampling process, we find an MDS of degree $O(\frac{d \log d}{\log \log d})$ with high probability. The total work expended is $m^2 \log^{O(1)} n$. This is summarized as Algorithm 3.

We next prove Theorem 5, which shows that this bound $O(\frac{d \log d}{\log \log d})$ is optimal up to constant factors.

Theorem 5. *For any real number $\alpha > 0$, there are graphs with average degree $\leq \alpha$, but for which any MDS has an average degree of $\Omega(\frac{\alpha \log \alpha}{\log \log \alpha})$.*

Proof. We will construct a graph of average degree $d = O(\alpha)$, all of whose MDS's have degree $\Omega(\frac{\alpha \log \alpha}{\log \log \alpha})$. To simplify the proof, we will ignore rounding issues. As all the quantities tend to infinity with α , such rounding issues are negligible for α sufficiently large.

Define $k = \log_2(\alpha \log \alpha / \log \log \alpha)$. We define a random process which constructs a graph with three types of vertices, which we denote A, B, C (these play the same role as in the proof of Theorem 4). The vertices in A, B are organized into clusters of related vertices. For class A , there are $l = (\frac{\log \alpha}{\log \log \alpha})^2$ clusters of size α . For class B , there are r clusters of size $\frac{\alpha \log \alpha}{r \log \log \alpha}$, for some $r = \Theta(k)$ (the constant will be specified later).

There are $2^k - 1$ vertices in class C . These are not organized into clusters but are considered individually. We index these vertices by the non-zero k -dimensional binary vectors over the finite field $GF(2)$. That is, C corresponds to $C = GF(2)^k - \mathbf{0}$.

We add the following edges to the graph (some of these edges are deterministic, some are random):

1. From each A -vertex to the other vertices in the same A -cluster.
2. From each B -vertex to all the other B -vertices, even those outside its cluster.
3. For each B -cluster b , we choose a random non-zero binary vector v_b in $GF(2)^k$. For each vertex in C , indexed by vector w , we construct an edge from all the B -vertices in the cluster b to the vertex w iff $v_b \cdot w = 1$. The dot product here is taken over the field $GF(2)$.
4. For each A -cluster a , we select $\frac{r \log \log \alpha}{\log \alpha}$ of the B -clusters uniformly at random, with replacement.

We add an edge from every vertex in the A -cluster a to every vertex in the selected B -clusters. This graph has degree $O(\alpha)$. The following lemmas characterize the behavior of this graph and its minimal dominating sets:

Lemma 6. *Any MDS of G contains at most one vertex from each B -cluster. If the MDS contains i such B -vertices, then it contains at least $2^{k-i} - 1$ C -vertices.*

Proof. Let v_1, \dots, v_i be the binary vectors associated with the selected B -vertices. Then the set of k -dimensional binary vectors which are perpendicular to v_1, \dots, v_i has dimension at least $k - i$. So there are at least $2^{k-i} - 1$ non-zero vectors perpendicular to v_1, \dots, v_i . The corresponding C -vectors have no edges to the selected B -vertices, and no edges to any other type of vertex. In order for them to be dominated, they must themselves be part of the MDS. \square

Lemma 7. *It is possible to select the parameter $r = \Theta(k)$ such that, with high probability, all C vertices have degree $\Omega(\frac{\alpha \log \alpha}{\log \log \alpha})$.*

Proof. Let us fix a particular C -vertex, associated to binary vector $w \in GF(2)^k$. This vector w is perpendicular to any randomly selected vector $v \neq \mathbf{0}$ with probability $1/2 - 2^{-k}$. Hence the expected number of B -clusters connected to it is $r(1/2 - 2^{-k})$. By Chernoff's bound, the probability that it connects to fewer than $r/4$ clusters is $\exp(-\Omega(r))$.

For r a sufficiently large constant multiple of k , this probability is much less than 2^{-k} . By the union bound, this implies that the probability that any C vertex connects to $< r/4$ B -clusters is negligible. So with high probability, every C -vertex connects to $\Omega(r)$ B -clusters. As every B -cluster has $\frac{\alpha \log \alpha}{r \log \log \alpha}$ vertices, this implies that the C -vertices have degree $\Omega(\frac{\alpha \log \alpha}{\log \log \alpha})$. \square

Lemma 8. *With high probability, the graph G satisfies the following property: For all sets X of B -clusters where X contains at least $(3/4)k$ distinct B -clusters, all but $O(\log \alpha)$ of the A -clusters are connected to some vertex in X .*

Proof. Suppose we fix a set X which contains $i \geq 3/4k$ distinct B -clusters. Any given A -cluster connects to $\alpha/(2^k/r)$ B -clusters chosen uniformly a random, so the probability that this A -cluster is disjoint to X is at most $(1 - i/r)^{\alpha/(2^k/r)} \leq \exp(-\alpha i/2^k)$. Hence the expected number of such A -clusters is at most $l \exp(-\alpha i/2^k)$. For $i \geq 3/4k$, this is $o(\log \alpha)$. Hence by Chernoff's bound, the probability that the number of disconnected A -clusters exceeds $\phi \log \alpha$ is at most $\exp(-\phi' \log \alpha)$, where ϕ' increases with ϕ .

The total number of such sets X is at most $2^r = \exp(O(k))$. So, by the union-bound, the probability that any such X has this event occurring is at most $\exp(O(k) - \phi' \log \alpha)$. For ϕ a sufficiently large constant, this probability is negligible. \square

Lemma 9. *Suppose the graph G has all the properties of Lemmas 6, 7, 8. Then every MDS of G has degree $\Omega(\frac{\alpha \log \alpha}{\log \log \alpha})$.*

Proof. Suppose we have an MDS of G which contains i distinct B -clusters. There are two cases. First, suppose $i \leq 3/4k$. In this case, the MDS contains at least $2^{k-i} \geq 2^{k/4}$ C -clusters. The MDS contains at most one vertex from each of the A -clusters. Hence, the average degree of the A, C vertices in this MDS is $\Omega(\frac{l\alpha + 2^{k/4} 2^k}{l + 2^{k/4}}) = \Omega(\frac{\alpha \log \alpha}{\log \log \alpha})$. As any B -vertex has also degree $\Omega(\frac{\alpha \log \alpha}{\log \log \alpha})$, this implies that the total average degree of the MDS is $\Omega(\frac{\alpha \log \alpha}{\log \log \alpha})$.

Next, suppose $i \geq 3/4k$. The total number of $B + C$ vertices is at least $i + 2^{k-i} = \Omega(k)$. All but $O(\log \alpha)$ of the A -clusters are already dominated by B vertices; these are the only A -clusters which can join the MDS. As a vertex in A cluster connects to only the other vertices in that same cluster, the total number of A -vertices in the MDS is at most $O(\log \alpha)$. Hence the degree of the MDS is at least $\Omega(\frac{\alpha \log \alpha + k 2^k}{\log \alpha + k}) = \Omega(\frac{\alpha \log \alpha}{\log \log \alpha})$. This completes the proof of Theorem 5. \square

With high probability, every MDS of G has degree $\Omega(\frac{\alpha \log \alpha}{\log \log \alpha})$.

4 Balanced Minimal Vertex Cover

Theorem 6. *For any real number $\alpha > 2$, there are graphs for which the average degree is at most α , but for which the average degree of any MVC is arbitrarily large.*

Proof. The following example shows that the ratio of the average degree of any MVC in the underlying graph to that of the graph itself can become arbitrarily large.

Consider the graph G that contains a single copy H of a complete graph K_p such that each vertex of H is connected to q neighbors, each of them of degree 1. Then we have $d = \frac{p-1+2q}{1+q}$.

On the other hand, any VC in G contains at least $p - 1$ vertices from H . In particular, the minimum-average-degree MVC in G contains exactly $p - 1$ vertices from H and the q neighbors of the remaining vertex of H , so has average degree at least $\frac{(p-1)(p-1+q)+q}{p-1+q}$. Now, let $p = \lfloor (\alpha - 2)q \rfloor$ and let $p, q \rightarrow \infty$. The resulting graphs have $d \leq \alpha$, while the MVC has its degree approach ∞ .

(Note that if we allow G to contain isolated vertices, then this theorem becomes a triviality: we can simply add arbitrarily many isolated vertices to a graph G .)

5 Conclusion

We have initiated the study – graph-theoretic, algorithmic, randomized, and distributed – of the balanced versions of some fundamental graph-theoretic structures. As discussed in Section 1, the study of balanced structures can be useful in providing fault-tolerant, load-balanced MISs and MDSs. We have developed reasonably-close upper and lower bounds for many of these problems. Furthermore, for the BMIS problem, we have presented fast (local) distributed algorithms that achieves an approximation close to the best possible in general; a key problem that is left open is whether one can do the same for the BMDS problem. We view our results also as a step toward understanding the complexity of local computation of these structures whose optimality itself cannot be verified locally.

References

1. N. Alon, L. Babai, and A. Itai. A fast and simple randomized parallel algorithm for the maximal independent set problem. *J. Algorithms*, 7(4):567–583, 1986.
2. N. Alon and J.H. Spencer. *Probabilistic Method*. Wiley, 2008.
3. L. Engebretsen and J. Holmerin. Clique is hard to approximate within $n^{1-o(1)}$. *Lecture Notes in Computer Science*, 1853:2–12, 2000.
4. Pierre Fraigniaud, Amos Korman, and David Peleg. Local distributed decision. In *FOCS*, pages 708–717, 2011.
5. M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
6. D. Gavinsky, S. Lovett, M. Saks, and S. Srinivasan. A tail bound for read-k families of functions. *arXiv preprint arXiv:1205.1478*, 2012.
7. Richard M. Karp and Avi Wigderson. A fast parallel algorithm for the maximal independent set problem. *J. ACM*, 32(4):762–773, 1985.
8. J. Kratochvil, P. Savicky, and Z. Tuza. One more occurrence of variables makes satisfiability jump from trivial to NP-complete. *SICOMP*, pages 203–210, 1993.
9. F. Kuhn, T. Moscibroda, and R. Wattenhofer. Local computation: Lower and upper bounds. *CoRR*, abs/1011.5470, 2010.
10. M. Luby. A simple parallel algorithm for the maximal independent set problem. *SIAM J. Comput.*, 15(4):1036–1053, 1986.
11. T. Moscibroda. Clustering. *Book Chapter in Algorithms for Sensor and Ad Hoc Networks*, pages 37–60, 2007.
12. D. Peleg. *Distributed Computing: A Locality-Sensitive Approach*. SIAM, 2000.
13. R. Rajaraman. Topology control and routing in ad hoc networks: a survey. *SIGACT News*, 33(2):60–73, 2002.
14. J. Spencer. Turan’s theorem for k -graphs. *Discrete Mathematics*, 2(2):183–186, 1972.
15. J. Suomela. Survey of local algorithms. *ACM Computing Surveys*, to appear.
16. P. Turan. On an extremal problem in graph theory. *Mat. Fiz. Lapok*, 48:436–452, 1941.
17. H. Zhang and H. Shen. Balancing energy consumption to maximize network lifetime in data-gathering sensor networks. *IEEE TPDS*, 20(10):1526–1539, 2009.