Approximating the Domatic Number

Uriel Feige* Magnús M. Halldórsson† Guy Kortsarz‡ Aravind Srinivasan§

Abstract

The domatic number problem is that of partitioning the vertices of an undirected graph into the maximum number of disjoint dominating sets. Let n denote the number of vertices in a graph, δ the minimum degree, and Δ the maximum degree. Trivially, the domatic number is at most $(\delta+1)$. We show that every graph has a domatic partition with $(1-o(1))(\delta+1)/\ln n$ dominating sets, and moreover, that such a domatic partition can be found in polynomial time. This implies a $(1+o(1))\ln n$ approximation algorithm for domatic number. We also show this to be essentially best possible. Namely, extending the approximation hardness of set cover by combining multi-prover protocols with zero-knowledge techniques, we show that for every $\epsilon > 0$, a $(1-\epsilon)\ln n$-approximation implies that $NP \subseteq DTIME(n^{O(\log\log n)})$. This makes domatic number the first natural maximization problem (known to the authors) that is provably approximable to within polylogarithmic factors but no better.

We also show that every graph has a domatic partition with $(1-o(1))(\delta+1)/\ln\Delta$ dominating sets, where the "$o(1)$" term goes to zero as Δ increases. This can be turned into an efficient algorithm that produces a domatic partition of $\Omega(\delta/\ln\Delta)$ sets.

1 Introduction

A *dominating set* in a graph is a set of vertices such that every vertex in the graph is either in the set or has a neighbor in the set. A *domatic partition* is a partition of the vertices so that each part is a dominating set of the graph. The *domatic number* of a graph is the maximum number of dominating sets in a domatic partition of the graph, or equivalently, the maximum number of disjoint dominating sets.

The domatic partition problem is one of the classical NP-hard problems. It is also one of the few graph problems in Garey and Johnson [16] whose approximability status on general graphs has until now been a blank page, with no published upper or lower bounds [8]. The purpose of this paper is to mend that situation and derive the optimal approximability within a lower order term.

The domatic partition problem arises in various situations of locating facilities in a network. Assume that a node in a network can access only resources located at neighboring nodes (or at itself). Then if there is an essential type of resource that must be accessible from every node (a hospital, a printer, a file, etc.), copies of the resource need to be distributed over a dominating set of the network. If there are several essential types of resources, each one of them occupies a dominating set. If each node has bounded capacity, there is a limit to the number of resources

*Weizmann Institute, Rehovot, Israel. feige@wisdom.weizmann.ac.il

†University of Iceland, Reykjavík, Iceland. mmh@hi.is. Work done in part at Kyoto University, Japan.

‡Open University, Ramat Aviv, Israel. guyk@shaked.openu.ac.il

§Bell Laboratories, Lucent Technologies, 600–700 Mountain Avenue, Murray Hill, NJ 07974-0636, USA. srin@research.bell-labs.com

that can be supported. In particular, if each node can only serve a single resource, the maximum number of resources supportable equals the domatic number of the graph [15]. We can show how the general case of larger, possibly non-uniform, capacities can be reduced to the unit case.

It is easily seen that the domatic number of a graph G, $D(G)$, is at most $\delta(G)+1$, where δ is the minimum degree. Also, if G contains an isolated vertex, it is easy to see that $D(G) = 1$, and the entire vertex set, for instance, forms a dominating set; if G does not contain any isolated vertex, then $D(G) \geq 2$, since any maximal independent set and its complement are dominating sets. Thus, any nontrivial approximation algorithm has performance ratio at most $(\delta + 1)/2$. Fujita [14] has studied several greedy algorithms and shown that their performance ratio is no better than $(\delta + 1)/2$, for values of δ up to $O(\sqrt{n})$. The only other lower bound on $D(G)$ given in a recent encyclopædic treatment of domination problems [19, 18] is $D(G) \geq \lceil n/(n - \delta(G)) \rceil$ [39], where n is the number of vertices. This lower bound is relevant only in very dense graphs, and degenerates to $D(G) \geq 2$ when $\delta(G) \leq n/2$.

Numerous results are known for special classes of graphs. A graph G is said to be *domatically full* if $D(G) = \delta(G) + 1$, the maximum possible. Determining if a d-regular graph is domatically full is NP-complete, for any $d \geq 3$ [34, 23]. Farber [10] showed nonconstructively that *strongly chordal* graphs are domatically full. This class contains the classes of interval graphs and path graphs. Rao and Rangan [37] then gave a linear time algorithm for interval graphs, and Peng and Chang [32] for strongly chordal graphs. Farber's theorem turned out to be a special case of a result of Berge [5] for *balanced* hypergraphs, and Kaplan and Shamir [21] presented a simple algorithm. They also showed split graphs and bipartite graphs to be NP-hard. Efficient algorithms are known for partial k-trees, using generic methods [2]. Bonucelli [6] showed that circular-arc graphs are NP-hard, while Marathe et al. [28] gave a 4-approximation algorithm.

Let Δ denote the maximum degree of a given graph. Our main result is a tight bound on the approximability of the domatic number problem in general graphs. In particular, we give:

(a) An algorithm that finds a domatic partition of size $(1 - o(1))(\delta + 1)/\ln n$, where the "$o(1)$" term goes to zero as n increases.

(b) An algorithm that finds a domatic partition of size $\Omega(\delta/\ln\Delta)$.

(c) A nonconstructive argument showing that the domatic number is at least $(1 - o(1))(\delta + 1)/\ln\Delta$, where the "$o(1)$" term goes to zero as Δ increases. This shows that the *value* of the domatic number can be approximated within a factor of nearly $\ln\Delta$.

(d) A construction showing that for every $\epsilon > 0$, no polynomial-time algorithm can approximate the problem within a $(1-\epsilon)\ln n$ factor, unless NP has slightly superpolynomial-time algorithms ($NP \subseteq DTIME(n^{\log\log n})$). It holds also in the special cases of split graphs and bipartite graphs. It also yields a $(1 - o(1))\ln\Delta$–hardness.

The $(1+o(1))\ln n$–approximation algorithm is a simple randomized assignment (though care is needed not to lose a factor of two in the analysis), and is derandomized using the method of conditional probabilities. The results of (b) and (c) use the Lovász Local Lemma (LLL) [9] as their basic tool. Suitable application of the LLL to our randomized assignment algorithm above shows that the domatic number is at least $(1/3 - o(1))(\delta + 1)/\ln\Delta$; we then refine this using a "slow partitioning" scheme, leading to our result that the domatic number is at least $(1 - o(1))(\delta + 1)/\ln\Delta$. The $O(\ln\Delta)$–approximation algorithm is a constructive version of the LLL, following an approach of Beck [4].

The hardness construction builds on the proof of Feige [11] of similar hardness for the set cover and dominating set problems. In fact, the construction here generalizes the result of [11] in that it shows that it is hard to distinguish between the following two cases: when the minimum dominating set is large (and thus the domatic number small), or when there are many small disjoint dominating sets. This parallels the situation with the archetypical *minimum* partitioning problem, graph coloring, where Feige and Kilian [13] showed that it is hard to distinguish between the case when the maximum independent set is small, and when the chromatic number is small. The construction of the current paper, in fact, draws additionally on the zero-knowledge techniques used in [13].

It is instructive to view our results in a larger context – that of the study of approximation algorithms in general. It has been empirically observed and further supported by classification of constraint satisfaction problems [22] that there seem to be no "natural" maximization problems approximable within polylogarithmic factors but no better. Our results provide (to the best of our knowledge) the first maximization problem with such an approximation behavior, as the domatic number is a maximization problem approximable within logarithmic factors but no better.

Our algorithmic results give absolute ratios, namely bounds in terms of some basic parameters of the graph (minimum degree, number of vertices), rather than in terms of the size of the optimal solution. These are in fact the first nontrivial lower bounds on the size of an optimal domatic partition for arbitrary δ, Δ such that $\delta \geq \ln \Delta$:

$$D(G) \geq (1 - o(1)) \cdot \frac{\delta + 1}{\ln \Delta}. \tag{1}$$

As shown in Appendix A, this bound is best possible up to lower order terms, for a large range of values of $\delta = \delta(n)$ and $\Delta = \Delta(n)$.

In the past, most absolute ratios have been obtained by fairly simple greedy algorithms. Our algorithms are derandomizations of simple randomized algorithms, but their derandomized versions are not particularly natural, and natural greedy algorithms for the problem attain much worse results. It is also interesting that the hardness result gives a "gap location at 1": namely, it is equally hard to approximately partition graphs that are domatically full. We can generalize the problem to partitioning hypergraphs into a maximum number of dominating sets. It corresponds to the maximum number of disjoint transversals of the dual hypergraph. The hardness results apply without change because graphs are special cases of hypergraphs.

The rest of the paper is organized as follows. Section 2 starts with an algorithm that finds a domatic partition into roughly $\delta/\ln n$ sets, followed by an algorithm that yields a partition with $\Omega(\delta/\ln \Delta)$ sets. It then proves (1). The section ends with a short proof that the performance ratio of the natural greedy algorithm is $O(\sqrt{n \log n})$. The hardness reduction is given in Section 3.

2 Approximation algorithms and existential results

This section is devoted to positive results for domatic partitions. Section 2.1 shows algorithmically that $D(G) \geq (1 - o(1))(\delta + 1)/\ln n$, and also shows that $D(G) \geq (1/3 - o(1))(\delta + 1)/\ln \Delta$. (The two usages of "$o(1)$" here respectively correspond to $n \to \infty$ and $\Delta \to \infty$.) This second result is made algorithmic in Section 2.2, with a loss in the constant factor; the existentially tight result that $D(G) \geq (1 - o(1))(\delta + 1)/\ln \Delta$ is then shown in Section 2.3. Section 2.4 is devoted to a short analysis of the natural greedy algorithm for domatic partition.

Notation. Let $N(v)$ denote the set of neighbors of a vertex v in the given graph G, and let $N^+(v) = \{v\} \cup N(v)$. Let $d(v) = |N(v)|$ denote the degree of v, and let $d^+(v) = |N^+(v)| = 1 + d(v)$. A *partial coloring* of G is an arbitrary coloring of an arbitrary subset of the vertices. Given a current partial coloring, define a Boolean variable $A_{v,c}$ to be true if there is no vertex of color c in $N^+(v)$, and to be false otherwise. Define $[\ell]$ to be the set $\{1, 2, \ldots, \ell\}$.

For an event X, $\mathcal{P}[X]$ denotes its probability and $\mathbf{E}[X]$ its expectation. Finally, let e denote the base of the natural logarithm.

2.1 Logarithmic bounds

Theorem 1 *Any graph admits a (polynomial-time constructible) domatic partition of size* $(\delta + 1)(1 - O(\log \log n / \log n))/ \ln n)$.

Proof. Independently give each vertex one of $\ell = (\delta + 1)/\ln(n \ln n)$ colors at random. For any vertex-color pair (v, c), $\mathcal{P}[A_{v,c}] = (1 - 1/\ell)^{d^+(v)} \leq e^{-d^+(v)/\ell} \leq 1/(n \ln n)$. The expected number of bad events $A_{v,c}$ is then at most $\ell / \ln n$, and the expected number of colors that form dominating sets is at least

$$\ell\,(1 - 1/\ln n) = \frac{\delta + 1}{\ln n}\left(1 - \frac{\ln \ln n + 1}{\ln(n \ln n)}\right). \tag{2}$$

The color-classes that do not form dominating sets can all be merged into any one color class that is a dominating set; thus, we get a domatic partition whose expected number of sets is at least as large as the r.h.s. of (2).

This randomized argument can be derandomized using the method of conditional probabilities (cf. [1]). Number the vertices arbitrarily as $v_1, v_2, \ldots, v_n$, and color the vertices in this order (never recoloring a vertex) as follows. Color v_1 arbitrarily. Suppose the first $j \geq 1$ vertices have been colored with respective colors $c_1, c_2, \ldots, c_j$; vertex v_{j+1} is colored as follows. Let $d_{j+1}(v) = |N^+(v) \cap \{v_{j+1}, v_{j+2}, \ldots, v_n\}|$. Then, the conditional probability of the bad event $A_{v,c}$ is given by

$$\mathcal{P}[A_{v,c} | c_1, c_2, \ldots c_j] = \begin{cases} 0 & \text{if } \exists v_z \leq j \text{ such that } v_z \in N^+(v) \text{ and } c_z = c; \\ (1 - 1/\ell)^{d_{j+1}(v)} & \text{otherwise.} \end{cases}$$

The weight of the current coloring is given by

$$g(c_1, c_2, \ldots, c_j) \doteq \sum_{v \in V} \sum_{c} \mathcal{P}[A_{v,c} | c_1, c_2, \ldots, c_j];$$

this is precisely the expected number of pairs (v, c) pairs for which $A_{v,c}$ will hold after coloring all vertices, given the current coloring $c_1, \ldots, c_j$. In each step $j + 1$, we choose a color for v_{j+1} so that the weight of the coloring does not increase. (Such a color exists, since $g(c_1, c_2, \ldots, c_j)$ is a convex combination of the values $\{g(c_1, c_2, \ldots, c_j, c_{j+1}) : \ c_{j+1} \in [\ell]\}$.) Then, the total number of colors that are not dominating sets is at most the weight of the final coloring, which we ensure is at most the expected number at the outset, or $\ell / \ln n$. $\square$

We now refine this argument using the LLL to get better bounds when $\Delta \leq n^{1/3}$. We state the symmetric, simpler version of the LLL.

Lemma 2 (LLL [9]) *Let $p < 1$, and let $\mathcal{E}_i$, $1 \le i \le k$ be k events such that $\mathcal{P}[\mathcal{E}_i] \le p$ for all i. Suppose there is an integer d such that $e \cdot p \cdot (d+1) \le 1$, and each event is independent of all but at most d other events. Then, $\mathcal{P}[\bigwedge_i \bar{\mathcal{E}_i}] > 0$.*

Independently color each vertex randomly with one of $\ell = \lfloor (\delta+1)/(3 \cdot \ln(3^{1/3} \cdot \Delta)) \rfloor$ colors. For each (v, c) pair, $\mathcal{P}[A_{v,c}] \le (1 - 1/\ell)^{d^+(v)} \le 1/(3 \cdot \Delta^3)$. We note that each event $A_{v,c}$ is independent of all but at most $\ell(1 + d(v) + d(v) \cdot (\Delta - 1))$ other such events, since vertices of distance at least 3 from v are completely irrelevant for v. The following lemma now directly follows from the LLL, using the fact that $\ell < \Delta$ and $d(v) \le \Delta$.

Lemma 3 *Any graph admits a domatic partition of size $(1/3 - o(1))\delta/\ln\Delta$, where the "$o(1)$" term tends to zero as $\Delta \to \infty$.*

In Section 2.3, we will refine the above approach by conducting a two-stage partitioning that attains the tight value of $1 - o(1)$ instead of the value $1/3 - o(1)$ of Lemma 3. However, the above direct approach will help us develop a simple algorithmic version of Lemma 3 in Section 2.2. It also motivates the reason for developing the approach of Section 2.3.

Remark: It is interesting to note that the Δ in the bound of Lemma 3 cannot be replaced by δ, nor by $\overline{d}$, the average degree. Consider, for example, the bipartite graph with $3 \cdot \delta$ vertices on the left side, and $\binom{3 \cdot \delta}{\delta}$ vertices on the right side, with each vertex on the right side connected to a particular subset of δ vertices in the left side. The domatic number of this graph is two. Indeed, say that there are 3 disjoint dominating sets. One of these sets, S, contains at least δ vertices on the left side. There exists a vertex v on the right side all of whose neighbors are in S. Hence, the two remaining sets must both contain the vertex v, a contradiction.

A parameter that is intermediate between average and maximum degree is the *inductiveness* $\Delta^*(G) = \max_{H \subseteq G} \delta(H)$. It is most notable for giving a tighter upper bound on the chromatic number than the maximum degree, as $\chi(G) \le \Delta^*(G) + 1 \le \Delta(G) + 1$. The above construction shows however that Δ^* cannot replace Δ in the bound of Lemma 3 on the domatic number.

2.2 $O(\log \Delta)$–approximation algorithm

We use an algorithmic version of the LLL due to Beck [4] that derandomizes the probabilistic argument with some loss in the constants. We may assume without loss of generality that G is connected, since we can treat the connected components separately. Our algorithm has three phases, assigning colors to successively larger fractions of the vertices. After the first phase, each vertex is either fully satisfied, seeing all colors within its neighborhood, or has at most one-third of its neighbors colored. We show that the subgraph induced by nodes that are still active, i.e. either unsatisfied or not yet colored, consists with high probability of only small connected components, of $O(\Delta^6 \log n)$ vertices each. After the second phase, more vertices are colored, with at most two-thirds of the neighbors of yet-unsatisfied vertices being colored. The connected components induced by active vertices are now of only $O(\Delta^7 \log\log n)$ size. Then, depending on the value of Δ, we can either solve each component by exhaustive search, or apply Theorem 1 to obtain a full coloring where each vertex is satisfied.

The first phase proceeds as follows. Each vertex is either colored or gets *frozen*. Let $\ell = \delta/(c \ln \Delta)$ for a suitably large constant c. Order the vertices arbitrarily as $v_1, v_2, \ldots, v_n$ and process them in this order. The *pre-neighbors* (*post-neighbors*) of vertex v_i are the neighbors of v_i among $v_1, v_2, \ldots, v_{i-1}$ ($v_{i+1}, v_{i+2}, \ldots, v_n$, respectively). When processing v_i, if it has by then been frozen, we ignore it in this phase, otherwise we assign it one of ℓ colors independently at random. We mark v_i (whether colored or frozen) as *dangerous* iff:

1. at least $\delta/3$ pre-neighbors of v_i have been colored, and
2. not all the ℓ colors appear in the pre-neighborhood v.

If v is dangerous, we freeze all the post-neighbors of v (i.e., all neighbors of v that are numbered higher than v). When the process ends, some vertices are colored and some are frozen, and some are dangerous and some are not. Let $X(u)$ be the indicator random variable for vertex u becoming dangerous, and let $q = \ell(1-1/\ell)^{\delta/3}$.

Lemma 4 *Let $U = \{u_1, u_2, \ldots, u_k\}$ be any set of vertices with pairwise distance at least 3. Then,* $\Pr[X(u_1) = X(u_2) = \cdots = X(u_k) = 1] \le q^k$.

Proof. Note that if a vertex v is a neighbor of the set U, then it has a unique neighbor in U since the elements of U have pairwise distance at least 3. Let $\vec{a} = (a_1, a_2, \ldots, a_k)$ be any sequence of k colors, and let S_i be the random variable denoting the set of the first i nonfrozen neighbors of U. Let $D_i(\vec{a})$ be the event that "for all $j = 1, 2, \ldots, k$, all neighbors of u_j in S_i avoid color a_j". Thus, $D_i(\vec{a})$ is the event that even after processing the first i nonfrozen neighbors of U, each u_j in U was missing a particular color a_j. In order to let i range all the way up to $\delta k/3$, add $\delta k/3$ dummy vertices just before the last (i.e., highest-numbered) vertex in U. These are considered to be virtual neighbors of U that are never frozen. Hence, regardless of the freezing process, $D_i(\vec{a})$ is defined for all $i \le \delta k/3$. We have $\Pr[D_0(\vec{a})] = 1$. Now for every $i < \delta k/3$, $\Pr[D_{i+1}(\vec{a})] = \Pr[D_i(\vec{a})] \cdot (1 - 1/\ell)$, because the color of the $(i+1)$'st nonfrozen neighbor of U is chosen at random independent of the previous colors, and independent of which vertex it happens to be. Hence, $\Pr[D_i(\vec{a})] = (1-1/\ell)^i$; so, $\Pr[\exists \vec{a} : D_{\delta k/3}(\vec{a})] \le \ell^k (1-1/\ell)^{\delta k/3} = q^k$.

The proof is completed by noting that if all the u_j are dangerous, then there is some $\vec{a}$ for which $D_{\delta k/3}(\vec{a})$ is true. □

The vertices that are not dangerous belong to one of two categories:

Good: A good vertex sees all colors in its neighborhood.

Neutral: A neutral vertex v does not see all colors in its neighborhood, but is not dangerous. This can happen only if more than 2/3 of v's neighbors were frozen.

Thus, we have two orthogonal partitions: colored/frozen, and good/neutral/dangerous.

Vertices that are both good and colored do not need to be considered further in the later phases. Call the other vertices *saved*, i.e. those that are dangerous, frozen, or neutral. As in [4], we are interested in the maximum size of a connected component of the subgraph induced by saved vertices, as this bounds the size of the independent subproblems in the next phase. We show that with high probability the maximum number of vertices in a spanning tree of such a component is "small". This is done as follows. By a standard argument, a large connected component contains many vertices with a particular minimum pairwise distance. We first prove that the number of vertices with large pairwise mutual distance which are all saved is "small". This indirectly bounds the maximum number of vertices in a connected component as a function of Δ, which is enough for our purposes.

A set of vertices is said to be *7-separated* if it is of mutual distance at least 7. A 7-separated set of k vertices is said to be a *bad k-set* if, additionally, it becomes connected if we connect all vertices of distance exactly 7. As shown next, the number of such sets in G is at most

$$n(4\Delta^7)^k. \tag{3}$$

Consider a spanning tree on the set where vertices of distance exactly 7 are connected. The number of distinct ordered rooted trees on k vertices is at most 4^{k-1}. Namely, such a tree can be uniquely represented by ordering the vertices in a lexicographic breadth-first order and attaching two flag bits to each nonroot vertex: whether it has the same parent as the previous vertex in the order, and whether it has a child or not. For each shape of a tree, there are n possibilities of choosing the root, and thereafter at most Δ^7 possibilities of choosing each new vertex since we already chose its parent in the tree.

Let $Y(u)$ be the indicator random variable for vertex u becoming saved. To complete our argument that no connected component of the saved vertices is "large", we will show:

Lemma 5 *For any 7-separated set of vertices $v_1, v_2, \ldots, v_k$,*

$$\Pr[Y(v_1) = Y(v_2) = \cdots = Y(v_k) = 1] \le (2.5\Delta q)^k.$$

Proof. If a vertex u is saved, then we have one of three cases: (i) u is dangerous and thus $X(u) = 1$; or (ii) u has a dangerous neighbor and so $\sum_{v \in N(u)} X(v) \ge 1$; or (iii) u is neutral, so at least $2d(u)/3$ of its neighbors are frozen, and by the preceding argument it holds for each frozen neighbor v of u that $\sum_{w \in N(v)} X(w) \ge 1$. Therefore, we have the simple but useful observation that *if* $Y(u) = 1$, *then* $Z(u) \ge 1$, where we define

$$Z(u) \doteq X(u) + (\sum_{v \in N(u)} X(v)) + \frac{\sum_{v \in N(u)} \sum_{w \in N(v)} X(w)}{2d(u)/3}.$$

By Markov's inequality,

$$\Pr[Y(v_1) = Y(v_2) = \cdots = Y(v_k) = 1] \le \Pr[\prod_{i=1}^{k} Z(v_i) \ge 1] \le \mathbf{E}[\prod_{i=1}^{k} Z(v_i)]. \tag{4}$$

Now, using the definition of $Z(\cdot)$ and the linearity of expectation, we can expand $\mathbf{E}[\prod_i Z(v_i)]$ as a linear combination of the terms $\Pr[X(w_1) = X(w_2) = \cdots = X(w_k) = 1]$. The main observation is that since the v_i have pairwise distance at least 7, the w_i have pairwise distance at least 3. Thus, by Lemma 4, any such term has probability at most q^k. Thus we get

$$\mathbf{E}[\prod_i Z(v_i)] \le \left(q + \Delta q + \frac{d(u)(\Delta - 1)q}{2d(u)/3}\right)^k \le (2.5\Delta q)^k,$$

by first replacing the probability of intersection of events by the product of probabilities, and then unfolding and reversing the above expansion. □

Consider now a connected component in the subgraph of G of saved vertices. If its size is $k\Delta^6$ or more, then it must contain a bad k-set (which is obtained by repeating the procedure of putting a vertex in the bad k-set and removing all vertices of distance at most 6). Setting the constant c in the expression $\ell = \delta/(c \ln \Delta)$ large enough, we get that $q \le \Delta^{-9}/10$. Set $k = \log(2n)/\log \Delta$, and recall (3) and Lemma 5. We get that the probability of existence of a connected component of the saved vertices with cardinality at least $k\Delta^6$, is at most $(2.5q\Delta)^k n(4\Delta^7)^k \leq 1/2$. Thus, by repeating the above process if necessary, we get with high probability that the largest connected component in the saved graph has size $O(\Delta^6 \log n)$. This completes phase one of the algorithm.

Phase two of the algorithm is run separately on each connected component induced by the saved vertices. Note that each dangerous or neutral vertex v has at least $d(v) - \delta/3$ frozen (i.e., uncolored) neighbors in its connected component. Phase two differs from phase one in that some of the vertices are colored before we begin. We leave these colors untouched, because they may be useful for the vertices that were not saved. We only color the frozen vertices, and again define the notion of phase-two dangerous, frozen and neutral vertices. Analysis similar to that of phase one shows that the connected components of the newly saved vertices has size at most $N = O(\Delta^6(\log \Delta + \log \log n))$.

In phase three, each dangerous/neutral vertex has at least $d(v)-2\delta/3 \geq \delta/3$ frozen neighbors in its connected component. If $\Delta > \log \log n$, we have a domatic partition of the frozen vertices to roughly $\delta/(3 \ln N)$ parts, via Theorem 1. As $\ln N = O(\log \Delta)$, this is good enough. If $\Delta \leq \log \log n$, then we can find a domatic partition of size $\Omega(\delta/\ln \Delta)$ whose existence is guaranteed by Theorem 3, using exhaustive search. Since $\Delta \leq \log \log n$, this only takes time

$$N^{O(\delta/\ln \Delta)} \leq (\log \log n)^{O((\log \log n))} \leq poly(n).$$

Finally, the above Las Vegas algorithm can be derandomized by the approach of *pessimistic estimators* [35], which is a generalization of the method of conditional probabilities. Briefly, we proceed as follows. As before, we process the vertices one-by-one. Suppose it is currently the turn of vertex v. If v is frozen, we skip over it. Otherwise, we deterministically choose a color for it that minimizes the probability of emergence of a large connected component of the saved vertices, using our bounds derived above. Since $k = \log(2n)/\log \Delta$, we see from (3) that the number of possible bad k-sets to be considered in our analysis above, is bounded by a polynomial in n. Hence, we can write down a pessimistic estimator and choose, in deterministic polynomial time, a color for vertex v that minimizes the pessimistic estimator.

2.3 Getting the right constant

Our next result improves the value $1/3 - o(1)$ of Lemma 3 to the existentially best-possible value of $1 - o(1)$.

Theorem 6 *There is a constant $a > 0$ such that for any graph G with $\Delta \geq 3$, $D(G) \geq \lfloor \frac{\delta}{\ln \Delta + a \ln \ln \Delta} \rfloor$.*

The value of $D(G)$ for $\Delta \leq 2$ is well-known via a simple case analysis; there is also a linear-time algorithm for the domatic partition problem if $\Delta \leq 2$.

Proof. We will assume throughout that Δ is large enough, i.e., for some absolute constant Δ_0, we will assume that $\Delta \geq \Delta_0$. (Indeed, if $\Delta \leq \Delta_0$, the theorem holds by setting a large enough.) For the rest of Section 2.3, any "$o(1)$" term will denote a function of Δ alone that goes to zero as Δ increases. We prove the theorem for a being any constant greater than 7, for all large enough Δ; this choice of a can be further improved, but we do not attempt this optimization here.

Preprocessing. We preprocess the graph as follows. As long as there is an edge that has both end-points with degree more than δ, remove such an edge from the graph. At the end of this process, the minimum degree remains at δ, and the maximum degree is at most Δ. The useful property that now holds is that for each vertex u, either $d(u) = \delta$, or all neighbors v of u have $d(v) = \delta$.

There are two cases, the first one being simpler.

Case I: $\delta \leq \ln^4 \Delta$**.** We assume that $\delta > \ln \Delta + a \ln\ln \Delta$, since the theorem is trivially true otherwise. Define

$$\ell = \left\lfloor \frac{\delta}{\ln \Delta + a \ln\ln \Delta} \right\rfloor .$$

Color each vertex with a random color from $[\ell]$, independent of all other vertices. We will now use the LLL to show that $\mathcal{P}[\bigwedge_{u,c} \bar{A}_{u,c}] > 0$, in the same way as we did for Lemma 3. For each (u, c), we have

$$\Pr[A_{u,c}] = \left(1 - \frac{1}{\ell}\right)^{d^+(u)} \leq e^{-\ln(\Delta(\ln \Delta)^a)\cdot d^+(u)/\delta} \leq (\Delta(\ln \Delta)^a)^{-1}. \tag{5}$$

Let $N_2(u)$ denote the set of vertices at a distance of 0, 1, or 2 from u in G. As in our proof of Lemma 3, each event $A_{u,c}$ depends only on events $A_{v,c'}$ with $v \in N_2(u)$; so, it depends on at most $|N_2(u)| \cdot \ell$ other such events. Our preprocessing step above helps bound $|N_2(u)|$ for all u. If $d(u) = \delta$, then $|N_2(u)| \leq 1+\delta+\delta(\Delta-1)$. If $d(u) > \delta$, our preprocessing ensures that $d(v) = \delta$ for all neighbors v of u; so, $|N_2(u)| \leq 1+\Delta+\Delta(\delta-1)$. Thus, $|N_2(u)| \leq \delta\Delta+1 \leq \Delta \ln^4 \Delta+1$ for all u, since $\delta \leq \ln^4 \Delta$. So, each $A_{u,c}$ depends on at most $O(\Delta \ln^7 \Delta)$ other such events. Recalling the LLL and (5), we see that $\mathcal{P}[\bigwedge_{u,c} \bar{A}_{u,c}] > 0$ as required, since $a > 7$.

Case II: $\delta > \ln^4 \Delta$**.** Let $\epsilon = 1/(\ln \Delta)$; define $\ell_1 = \lfloor \epsilon^3 \delta \rfloor$ and $\ell_2 = \lfloor \ln^2 \Delta/(1 + b(\ln\ln \Delta)/\ln \Delta)\rfloor$, where b is any constant larger than 5. We will show the existence of a domatic partition of size $\ell_1\ell_2$, i.e., a coloring of V using $\ell_1\ell_2$ colors, in such a way that for every vertex u, there is at least one vertex of each color in $N^+(u)$. (Recall that $\delta > \ln^4 \Delta$. By choosing $b < 6$, for instance, we can ensure that $\ell_1\ell_2 \geq \delta/(\ln \Delta + 6 \ln\ln \Delta)$, for all large enough Δ.) It will be convenient to view the colors as elements of $[\ell_1] \times [\ell_2]$. We will apply a two-stage coloring: the first coloring determines the first components of the vertex-colors, and the second coloring is for the second components. We can view the first coloring as a coarse partition, which the second coloring turns into a fine partition. The primary purpose of the first coloring is to reduce the dependencies sufficiently for our analysis of the second coloring.

The first partitioning is as follows. Color each vertex with a random color from $[\ell_1]$, independent of all other vertices. For each vertex u and each color c, define $X^+_{u,c}$ to be the subset of $N^+(u)$ that receives color c. We have $\mathbf{E}[|X^+_{u,c}|] = d^+(u)/\ell_1$. Let $B_{u,c}$ be the "bad" event that $|\ |X^+_{u,c}| - d^+(u)/\ell_1\ | \geq 3\epsilon d^+(u)/\ell_1$. A Chernoff bound shows that

$$\Pr[B_{u,c}] \leq 2 \cdot \exp(-(9/2 - o(1)) \cdot d^+(u)\epsilon^2/\ell_1) \leq \exp(-(9/2 - o(1)) \ln \Delta). \tag{6}$$

Once again, $B_{u,c}$ is independent of any Boolean combination of events of the form $B_{v,c'}$ for vertices v at a distance of 3 or more from u. Thus, each $B_{u,c}$ "depends" on $o(\Delta^3)$ other such events. Recalling (6), the LLL shows that $\Pr[\bigwedge_{u,c} \overline{B_{u,c}}] > 0$.

Fix a coloring $\chi_1 : V \to [\ell_1]$ which avoids all the events $B_{u,c}$. Choose a random color $\chi_2(u) \in [\ell_2]$ for each u, independent of all other vertices; the final color of u is the pair $(\chi_1(u), \chi_2(u))$. Let $\mathcal{B}_{u,c_1,c_2}$ be the bad event that there is no vertex of color (c_1, c_2) in $N^+(u)$. We now use the LLL to show that all these bad events can be avoided with positive probability.

For each vertex u and each $c \in [\ell_1]$, let $N^+_{u,c} = \{v \in N^+(u) : \ \chi_1(u) = c\}$, and define $d^+_{u,c} = |N^+_{u,c}|$. Since χ_1 avoids all the events $B_{u,c}$, we have

$$\forall(u, c),\ (1 - 3\epsilon)d^+(u)/\ell_1 \leq d^+_{u,c} \leq (1 + 3\epsilon)d^+(u)/\ell_1. \tag{7}$$

Fix an event $\mathcal{B}_{u,c_1,c_2}$. First,

$$\Pr[\mathcal{B}_{u,c_1,c_2}] = \left(1 - \frac{1}{\ell_2}\right)^{d^+_{u,c_1}} \leq e^{-(1-3\epsilon)d^+(u)/(\ell_1\ell_2)} \leq (\Delta(\ln\Delta)^b)^{-(1-3\epsilon)} \leq O((\Delta(\ln\Delta)^b)^{-1}); \quad (8)$$

the first inequality here follows from (7). Next, which other events does $\mathcal{B}_{u,c_1,c_2}$ depend on? Given $S \subseteq V$, let $N^+(S) \doteq \bigcup_{v \in S} N^+(v)$. Note that $\mathcal{B}_{u,c_1,c_2}$ simply says that all elements of N^+_{u,c_1} got a $\chi_2(\cdot)$ value different from c_2. Thus, we can check that $\mathcal{B}_{u,c_1,c_2}$ only depends on the events in

$$S(u, c_1, c_2) = \{\mathcal{B}_{v,c'_1,c'_2} : \ v \in N^+(N^+_{u,c_1}) \text{ and } c'_1 = c_1\}. \quad (9)$$

More precisely, we claim that $\mathcal{B}_{u,c_1,c_2}$ is independent of any Boolean function of the events lying outside $S(u, c_1, c_2)$; this can be verified by seeing that

$$N^+_{u,c_1} \cap \Big(\bigcup_{(v,c'_1,c'_2):\ \mathcal{B}_{v,c'_1,c'_2} \notin S(u,c_1,c_2)} N^+_{v,c'_1} \Big) = \emptyset.$$

We now bound $|S(u, c_1, c_2)|$, in order to apply the LLL; once again, our preprocessing step will be of help. If $d(u) = \delta$, then $|N^+(N^+_{u,c_1})| \leq \Delta |N^+_{u,c_1}|$; this is at most $O(\delta\Delta/\ell_1)$, by (7). If $d(u) > \delta$, our preprocessing ensures that $|N^+(N^+_{u,c_1})| \leq \delta |N^+_{u,c_1}|$; so, from (7), we again get that $|N^+(N^+_{u,c_1})| \leq O(\delta\Delta/\ell_1)$. Thus, $|S(u, c_1, c_2)| \leq O(\delta\Delta\ell_2/\ell_1) = O(\Delta \ln^5 \Delta)$. We can now apply the LLL, using (8) and the facts that: (i) $b > 5$, and (ii) each bad event $\mathcal{B}_{u,c_1,c_2}$ depends on at most $|S(u, c_1, c_2)|$ others. This completes the proof. $\square$

To see why our two-stage coloring helps, note that the "dependence" $|S(u, c_1, c_2)|$ in the second coloring above is only $\Delta^{1+o(1)}$, as compared to the dependence of $\Delta^{3+o(1)}$ that we could get in the direct-coloring approach underlying Lemma 3. The constraint "$v \in N^+(N^+_{u,c_1})$" in (9) saves us a factor of $\Delta^{1-o(1)}$, and the constraint "$c'_1 = c_1$" saves another factor of $\Delta^{1-o(1)}$. That the first-stage coloring eliminates many dependencies in this fashion, is the main idea motivating this approach.

It is an open question if a domatic partition of the size guaranteed by Theorem 6 can be found in polynomial time.

2.4 Greedy algorithm

One natural approach to the domatic partition problem is to try to greedily choose small dominating sets. The greedy algorithm iteratively pulls out dominating sets from the graph, until the remainder is no longer dominating. The dominating sets are found by a standard $O(\log n)$-approximate greedy algorithm [20, 24].

Lemma 7 *Suppose $D(G) = n/k$. Then, the greedy algorithm finds a domatic partition of $\Omega(n/(k^2 \log n))$ sets.*

Proof. We count how many disjoint dominating sets our algorithm finds before the set of vertices in them intersects at least half of the n/k vertex-disjoint dominating sets in the graph. During this period, there are at least $n/(2k)$ disjoint dominating sets, thus in each step there exists a dominating set of size at most $2k$ and we find one of size at most $2k \ln n$. Hence, in each step, a vertex from at most $2k \ln n$ different dominating sets is removed. It then requires at least $(n/2k)/(2k \ln n)$ steps to halve the original number of dominating sets in the graph. $\square$

Since $D(G) \leq n$, the approximation ratio is maximized when $k \approx \sqrt{n/(4 \ln n)}$.

Corollary 8 *The performance ratio of the greedy domatic partition algorithm is* $O(\sqrt{n \ln n})$.

Fujita [14] has shown examples where the performance of this and some other greedy algorithms is $\Omega(\sqrt{n})$.

3 Hardness of approximating the domatic number

We say that a problem is *hard to approximate within ratio* ρ if having a polynomial time (randomized) ρ-approximation algorithm for it would violate some standard hardness assumption, such as P $\neq$ NP. The hardness assumption that we use in this paper is that NP does not have (randomized) algorithms that run in time $n^{O(\log \log n)}$; for brevity we shall just use the term *hard to approximate.*

We shall prove the following theorem.

Theorem 9 *For every fixed* $\epsilon > 0$*, it is hard to approximate the domatic number within a ratio of* $(1 - \epsilon) \ln |V|$.

For this purpose, it is helpful to work with a related but different problem.

Definition 1 *A* one-sided dominating set *in a bipartite graph* $G(V_1, V_2, E)$ *is a set of vertices* $U \subseteq V_1$ *such that for every* $v \in V_2$ *there is some* $u \in U$ *with* $(u, v) \in E$*. Here it is assumed that the intended bipartition* (V_1, V_2) *is given explicitly as part of the input and that every vertex in* V_2 *has some neighbor in* V_1.

The one-sided domatic number *of a bipartite graph is the maximum number of mutually disjoint one-sided dominating sets that the graph contains.*

A related problem is the *set cover* problem, where a collection of subsets $\mathcal{S}$ of a base set U is given, and we are to find a minimum cardinality subcollection that contains all elements of U. The *set cover packing number* is then the maximum number of mutually disjoint set covers. It is well-known that minimum dominating set, minimum one-sided dominating set and minimum set cover are strongly related, and that $\ln n$ is the best approximation ratio for all of them within lower order terms (details in Section 3.2). The one-sided domatic number problem can be shown to be equivalent to the set cover packing problem in terms of optimization. We are not aware of a similar relationship between one-sided domatic number and domatic number. However, we can give a reduction that yields the necessary result.

Proposition 10 *Let* $c > 1$ *and consider bipartite graphs* $G(V_1, V_2, E)$ *with* $|V_1|$ *large enough (e.g.,* $|V_1| \geq 4^c$*),* $|V_2| > |V_1|^c$*, and the following* promise*: for some* $0 \leq \epsilon \leq 1 - 1/c$ *and for* r *and* q *satisfying* $rq > (1 - \epsilon)|V_1| \ln |V_2|$ *(*r *and* q *may depend on the size of* G*) either*

- *The size of the smallest one-sided dominating set in* G *is at least* r*, or*
- *The one-sided domatic number of* G *is at least* q.

Note that the two cases for the promise cannot both hold. If it is hard to distinguish which of the two cases holds, then it is hard to approximate the domatic number within a ratio of $(1 - 1/c - \epsilon) \ln |V|$.

The proof of Proposition 10 is given in Section 3.2. The main result of the section is the following theorem, proved in Section 3.5.

Theorem 11 *For every $\epsilon > 0$ and every integer $c > 0$ the two cases of Proposition 10 cannot be distinguished in (random) polynomial time unless NP has (randomized) algorithms that run in time $n^{O(\log \log n)}$.*

Theorem 9 now follows from Theorem 11 and Proposition 10.

Remark: Theorem 9 implies a $\ln \Delta$ hardness of approximation result, for $\Delta \simeq n^{\psi}$ for some $0 < \psi < 1$ which is close to 1. To obtain $\ln \Delta$ hardness of approximation results when Δ is much smaller compared to n, simply make many disjoint copies of the graph, increasing n without changing Δ or the domatic number.

3.1 Overview and intuition

Before presenting the proof of Theorem 11, let us provide some background on proving hardness of approximation results in general, and how hardness of approximation results were proved for problems related to domatic number.

A convenient starting point for proving hardness of approximation results is the problem of Max 3SAT. The input to this problem is a 3CNF formula and the desired output is an assignment to the variables that satisfies as many clauses as possible. The well known PCP theorem of [3] implies (or in fact, is equivalent to) the following *gap*: for some $\epsilon > 0$ it is NP-hard to distinguish between 3CNF formulas that are satisfiable (which we call *yes* instances) and 3CNF formulas in which every assignment satisfies at most a $(1 - \epsilon)$ fraction of the clauses (which we call *no* instances). This hardness result can be extended to a restricted version of Max 3SAT in which the input 3CNF formula has the property that each variable appears in exactly 5 clauses. (The choice of 5 is arbitrary here. Any other constant greater than 5 would do as well.) We call this restricted version Max 3SAT-5.

As noted above, the set cover problem is strongly related to the dominating set problem, which in turn is related to the domatic number problem. Moreover, the one-sided domatic number problem is equivalent to the set cover packing problem. Hence our plan for proving Theorem 11 is to take known results regarding the hardness of approximation of set cover, and modify their proof so that it shows hardness of approximation for the set cover packing problem as well (and hence also for one-sided domatic number). To see more explicitly what needs to be done, let us first review at a very high level the known result [11] that set cover (and one-sided dominating set) is hard to approximate within a factor of $(1 - \epsilon) \ln n$.

The proof in [11] reduces instances of Max 3SAT-5 to instances of one-sided dominating set. The reduction is slightly super polynomial (instances of size n are mapped to instances of size $n^{O(\log \log n)}$) and is a *gap reduction* in the following sense: *yes* instances give bipartite graphs that have small one-sided dominating sets and *no* instances give graphs all of whose one-sided dominating sets are much larger. To prove hardness of approximation for one-sided domatic number, the requirement for *no* instances does not change, as it already implies a small one-sided domatic number. However, we would like *yes* instances to give bipartite graphs that have not just one small one-sided dominating set, but many disjoint small one-sided dominating sets, and hence a large one-sided domatic number.

To achieve this extra property, we invoke an idea used in [13] when proving hardness of approximation for chromatic number. In our context, it suffices to change the starting point of the reduction, from the problem Max 3SAT-5 to the problem Max-3-colorability-5. This is the problem of coloring a 5-regular graph with 3 colors, so as to maximize the number of edges legally colored (see Section 3.3). It also has a gap similar to that of Max 3SAT-5. The important property of Max 3-colorability-5 is that *yes* instances of it necessarily have many "disjoint" solutions. When these instances are reduced to instances of one-sided dominating set, the resulting bipartite graph has many disjoint small one-sided dominating sets. This implies a large one-sided domatic number, as required by Theorem 11.

Hence, to complete the proof of Theorem 11, we need to accomplish three things.

1. Introduce the problem of Max 3-colorability-5 and its properties. This is done in Section 3.3.

2. Give the reduction from Max 3-colorability-5 to one-sided dominating set. This reduction closely follows the reduction of [11] from Max 3SAT-5 to set cover, except for one small extra step (vertices on one side of the bipartite graph are duplicated many times, an operation that was not performed in the reduction to set cover). This reduction is fairly complicated, but essentially all complications come from the reduction in [11].

3. Prove the properties of the reduction. This has two parts. One is to show that *yes* instances of Max 3-colorability-5 are reduced to bipartite graphs with high one-sided domatic number, which is done in Lemma 17. The other is to show that *no* instances are reduced to bipartite graphs with only large one-sided dominating sets. This part is not proved in this paper, because the proof in [11] that *no* instances of Max 3SAT-5 give instances with a large set cover, extends virtually without change to our adaptation of the reduction of [11].

Appendix C reviews some of the ingredients of the reduction used in [11] from Max 3SAT-5 to set cover, and explains some modifications used in our context. Some of these ingredients are only used in the analysis of what happens on *no* instances, and hence do not come into any of the proofs in this paper. They are included in the overview only so as to give some indication on what led to the construction of the reduction. For more details, see [11].

3.2 Domination and one-sided domination

The three problems, minimum dominating set, one-sided dominating set and set cover, are equivalent in the following sense (see e.g., [31]).

Proposition 12 *There is a polynomial-time reduction between any of the three problems dominating set, one-sided dominating set and set cover that preserves the value of the minimum solution.*

Proof. To reduce dominating set to set cover, let V (the set of vertices) become U (the ground set) and let the collection $\mathcal{S}$ include the sets $N^+(v)$ for every $v \in V$. To reduce set cover to one-sided dominating set, construct a bipartite graph $G(V_1, V_2, E)$ in which V_2 is the ground set U, and the vertices of V_1 each represent a set in $\mathcal{S}$. Put an edge $(u, v) \in E$ if $v \in V_2$ corresponds to an item that is contained in the set that corresponds to the vertex $u \in V_1$. Finally, to reduce one-sided dominating set to dominating set make a clique out of all vertices of V_1 and let $V = V_1 \bigcup V_2$; it is not hard to see that the size of the minimum dominating set is preserved. $\square$

For set cover, let $n = |U|$. It is known that minimum set cover can be approximated within a ratio of $\ln n$ [20, 24], and that for every fixed $\epsilon > 0$, it is hard to approximate it within a ratio of $(1-\epsilon)\ln n$ [11]. By proposition 12, this implies a similar result for one-sided dominating set, with $n = |V_2|$. For dominating set, it is natural to take $n = |V|$. Then the approximation ratio of $\ln n$ trivially applies, but in order to transfer the $(1-\epsilon)\ln n$ hardness result from one-sided dominating set to dominating set using the reduction of Proposition 12 we also need that $\ln |V| \simeq \ln |V_2|$, which holds whenever $|V_1| \leq |V_2|^{1+\epsilon}$. It turns out that this is indeed the case in the construction of [11]. Hence it is known that up to low order terms, $\ln n$ is the best possible approximation ratio for all three problems.

A proof similar to that of Proposition 12 shows that there is a polynomial-time reduction from domatic number (whether one-sided or not) to set cover packing number that preserves the value of the maximum solution. Likewise, there is a polynomial-time reduction from set cover packing number to one-sided domatic number that preserves the value of the maximum solution. However, we are not aware of such a reduction from one-sided domatic number to domatic number. Instead, we use the following proposition.

Proposition 13 *For every integer $k > 0$ (where k may be an arbitrary function bounded by a polynomial in the size of the input) there is a reduction from one-sided domatic number to domatic number with the following properties:*

- $|V| = |V_2| + k|V_1|$; *and*
- *For the original problem, let q and r respectively denote the one-sided domatic number and the minimum cardinality of a one-sided dominating set; hence $q \leq |V_1|/r$. Let p denote the domatic number of the new problem. Then $kq \leq p \leq \min[|V|/r, 1 + 2k|V_1|/r]$.*

Proof. Let $G(V_1, V_2, E)$ be a bipartite graph for which we are interested in computing the one-sided domatic number. Construct a graph $G'(V, E')$ as follows. $V_1' = \bigcup_{i=1}^{k} V_1^i$, where for every i, V_1^i is a copy of V_1. $V = V_1' \bigcup V_2$, which gives $|V| = |V_2| + k|V_1|$. For every $v \in V_2$, $1 \leq i \leq k$ and $u \in V_1^i$, place an edge $(u, v) \in E'$ iff there is an edge $(u, v) \in E$. In addition, all vertices of V_1' form a clique.

Observe that every one-sided dominating set in G corresponds in a natural way to k mutually disjoint dominating sets in G', one on each copy of V_1^i. Hence the optimal solution for G can be copied k times on G', giving $p \geq kq$. (If in addition, V_1 has no isolated vertices, then V_2 can serve as one more dominating set disjoint from all the others, giving $p \geq kq + 1$.)

To upper bound p, let $D_1, \ldots, D_p$ be a maximum cardinality collection of disjoint dominating sets in G'. Similar to the proof of Proposition 12, we can see that $|D_j| \geq r$ for all j. Hence $p \leq |V|/r$. When $|V_2| \gg k|V_1|$, this upper bound on p can be improved by observing that $(D_i \bigcup D_j) \bigcap V_1'$ is a dominating set contained in V_1'. So suppose we pair up any $2\lfloor p/2 \rfloor$ of the D_i's into $\lfloor p/2 \rfloor$ pairs, each of which forms a dominating set of size at least r. Then, $r\lfloor p/2 \rfloor \leq k|V_1|$; so, $p \leq 1 + 2k|V_1|/r$. □

We are now ready to prove Proposition 10.

Proof. Perform the reduction of Proposition 13 with $k = |V_2|$. If the first case holds then the domatic number of G' is at most $|V|/r$. If the second case holds then the domatic number is at least $q|V_2|$. The ratio between these two values is $q|V_2|/(|V|/r) = qr|V_2|/|V|$. We use $|V_2|/|V| = 1/(|V_1|+1)$ and $qr \geq (1-\epsilon)|V_1| \ln |V_2|$ to obtain that the ratio is at least $(1 - \epsilon - 1/|V_1|) \ln |V_2|$. Using $|V_2| > |V_1|^c$ and $|V_1| \geq 4^c$ we obtain that $\ln |V_2| \geq (1 - 1/(c + 1/2)) \ln |V|$. Now Proposition 10 follows assuming that $|V_1| \geq 2(c+1)^2$. □

Remark: The graphs G' formed in the above reduction are *split graphs*, which are graphs whose vertex-set can be partitioned into a clique and an independent set. These graphs are both chordal and complements of chordal graphs, where a graph is chordal if it contains no cycle with four or more vertices as an induced subgraph. Thus, a $(1 - o(1)) \ln n$ hardness for domatic number holds also for split (and thus chordal and co-chordal) graphs. Furthermore, one can get a similar result for bipartite graphs by the following modification. Instead of making a clique out of V_1, we add $(\delta + 1)$ vertices and connect them to every node in V_1. This affects the domatic number by at most 1, and the approximability hardness follows.

3.3 The problem Max 3-colorability-5

We now introduce the NP-language that serves as the basis to our reduction.

Definition 2 Max 3-colorability *is the problem of coloring the vertices of a graph with three colors, so as to maximize the number of legally colored edges (edges whose endpoints are colored differently).*

An NP-witness for 3-colorability is an assignment of colors to the vertices, such that each edge is legally colored. The witness can be checked in a probabilistic sense, by sampling an edge at random and checking whether the colors of its two endpoints disagree. The actual names of the colors of the vertices play no role because the names of the three colors can be arbitrarily permuted without changing the legality of the 3-coloring. Hence every NP-witness gives rise to six different witnesses (depending on the permutation used for the names of the colors), and cycling over the six witnesses, every edge gets its six legal colorings. In our reductions to one-sided dominating set, we shall use this structure of the set of witnesses for 3-colorability in order to show that if the resulting bipartite graph has a small one-sided dominating set, then in fact it has many disjoint one-sided dominating sets. We note that the same structure was used in [17] to construct a zero knowledge proof system for NP.

Call a graph H with some h vertices a *(γ, κ)-expander* if for any subset S of the vertices of H with $|S| \leq \gamma h$, the number of edges leaving S is at least $\kappa |S|$. For some constant $\kappa > 0$ and for any $\epsilon > 0$, there is a h_0 such that for all $h \geq h_0$, there is an explicitly constructible $(1/2, \kappa)$-expander with maximum degree at most 6 and with the number of vertices lying in the range $[h, h(1 + \epsilon)]$ [25]. In particular, this implies that for any given integer h, we can construct a $(2/3, \kappa')$-expander with maximum degree at most 6 and with the number of vertices being some $F(h)$ that satisfies $h \leq F(h) \leq 2h$, for some absolute constant $\kappa' > 0$. (We can proceed as follows. If $h \leq h_0$, where h_0 is a sufficiently large constant, just construct any connected h-vertex graph of maximum degree 6. If $h > h_0$, construct a $(1/2, \kappa)$-expander with maximum degree at most 6 and with the number of vertices lying in the range $[h, 2h]$, using [25]. It is easy to see that if h_0 is large enough, then this has the desired properties.)

We will also need the following theorem of [33]:

Theorem 14 ([33]) *For some explicit constant $\psi < 1$, it is NP-hard to distinguish between graphs that have a legal 3-coloring (that colors all edges legally), and graphs for which every 3-coloring legally colors at most a ψ-fraction of the edges.*

Proposition 15 *For some explicit constant $\psi < 1$, it is NP-hard to distinguish between 5-regular graphs that have a legal 3-coloring, and 5-regular graphs for which every 3-coloring legally colors at most a ψ-fraction of the edges.*

Proof. Given a graph $G(V, E)$, we show that it can be modified in polynomial time to a graph $G'(V', E')$ such that: (i) G' is 5-regular; (ii) $|E'| = O(|E|)$; (iii) G' is legally 3-colorable iff G is, and (iv) For some constant $c > 0$ and every $1 \leq k \leq |E'|$, every 3-coloring of G' that leaves k edges illegally colored can be transformed in polynomial time to a 3-coloring of G that leaves at most ck illegally colored edges. We can then invoke Theorem 14.

First, we may assume that every vertex in G has degree at least three. Indeed, suppose we repeatedly remove any vertex of degree at most 2 until the remaining graph G'' has minimum degree at least 3. It is easy to see that given any 3-coloring of the vertices of G'', we can add back the deleted vertices of G and color these added-back vertices in such a way that all the edges incident with them are legally colored. So, we assume that G has minimum degree at least 3.

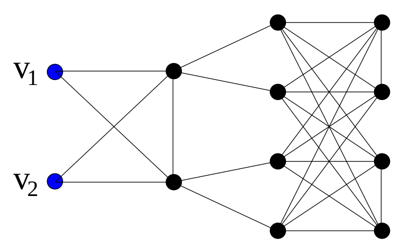

Figure 1: Equality gadget

Assume first that all vertices in G have degree at most 13. Then we replace each vertex v by a cluster of $d(v)$ vertices[1], where each vertex handles one outgoing edge. (That is, if (v, u) is an edge in G, then the vertex representing u in v's cluster, is made adjacent to the vertex representing v in u's cluster.) These $d(v)$ vertices are connected in a cycle by *equality gadgets*, shown in Fig. 1. These equality gadgets are subgraphs that contain twelve vertices. Two of the vertices are special and have degree two, and all the rest have degree five. The equality gadget has the property that it is legally 3-colorable iff the special vertices receive the same color. The two cluster vertices serve as the special vertices. As each cluster vertex participates in two gadgets and also has one outgoing edge its degree is 5. Hence G' is 5-regular. The number of edges added is at most $27 \sum_v d(v) \leq 54|E|$ and hence $|E'| \leq 55|E|$. Every legal 3-coloring of G' colors all cluster vertices with the same color, and hence naturally gives a legal 3-coloring of G. Similarly, a legal 3-coloring of G can be extended to a legal 3-coloring of G'. Any 3-coloring of G' in which cluster vertices receive different colors causes at least two edges of G' to be mis-colored. Coloring the vertex corresponding to the cluster with an arbitrary color in G causes at most $d(v) \leq 13$ edges to be mis-colored. Hence the number of illegally colored edges in G' is smaller than that of G by a factor of at most $13/2$.

If G has a vertex of degree more than 13, we create a new graph G_1 as follows. Let $F(\cdot)$ and κ' be as in our discussion on expanders above. Similarly as in [30], G_1 is the same as the G' constructed in the previous paragraph, except that for each vertex v, we create a $(2/3, \kappa')$-expander with $F(d(v))$ vertices and maximum degree at most 6 (instead of a cycle with $d(v)$ vertices), and do the above-seen operation of replacing the expander edges by equality gadgets. (Of the $F(d(v))$ vertices, $d(v)$ are "real" and represent the neighbors of v, and $F(d(v)) - d(v)$ are dummy vertices. As in the previous paragraph, if (v, u) is an edge in G, then the vertex representing u in v's cluster, is made adjacent to the vertex representing v in u's cluster.) Note

[1]Recall that $d(v)$ denotes the degree of vertex v.

that G_1 has maximum degree at most $2 \times 6 + 1 = 13$, and has $O(\sum_v d(v)) = O(|E|)$ edges. Once again, a 3-coloring of G can be efficiently transformed into one for G_1 in which the number of miscolored edges remains the same. Conversely, suppose we have a 3-coloring χ_1 of G_1. Let X_1 be the number of original edges of G that are miscolored by χ_1 and let Y_1 be the number of edges in the equality gadgets miscolored by χ_1. We will produce the following coloring χ of G and then analyze χ. For each vertex v, choose a largest subcluster $C(v)$ of vertices (from its cluster of $F(d(v))$ vertices) that receive the same color in χ_1, and define $\chi(v)$ to be the color assigned to the vertices in $C(v)$ by χ_1. Call an edge (u, v) of G *bad* if it was properly colored by χ_1 in G_1 and was miscolored by χ in G. Let X be the number of bad edges. Then the number of edges miscolored by χ is at most $X_1 + X$. An edge (u, v) is bad only if "$u \notin C(v)$ or $v \notin C(u)$" holds. Thus, letting $x(v) = F(d(v)) - |C(v)| \leq 2F(d(v))/3$, we see that $X \leq \sum_v x(v)$. Since $x(v) \leq 2F(d(v))/3$ and recalling the property of $(2/3, \kappa')$-expansion, we can check that the number of edges in all the equality gadgets of v that are mis-colored in χ_1 is $\Omega(x(v))$. Hence $Y_1 = \Omega(X)$. Armed with this property and the facts that G_1 has $O(|E|)$ edges and maximum degree 13, we transform G_1 into G' as described in the previous paragraph. □

3.4 Preliminaries

Our reduction closely follows that of [11]. The main differences are as follows. (i) The outcome of the reduction is one-sided dominating set, rather than set cover. This change is quite insignificant. (ii) The starting point of the reduction is Max 3-colorability rather than Max 3SAT. As mentioned earlier, the purpose of this change is to have the reduction apply to one-sided domatic number, rather than just one-sided dominating set. (iii) Every vertex in the V_1 side of the bipartite graph will be duplicated $2^{l/2}$ times, where l is a parameter of the reduction. This is a technical condition that seems to be required when we reason about the one-sided domatic number.

To describe the reduction, we recall two notions used in [11].

Definition 3 *A* (k, l)-Hadamard code *is a set of k binary words of length l, where every codeword has Hamming weight $l/2$ and the Hamming distance between every two codewords is $l/2$.*

There is a simple construction of Hadamard codes when l is a power of 2 and $k \leq l$ (see, e.g., [27]).

Definition 4 *A* partition system $B(m, L, k, d)$ *has the following properties:*
- *There is a ground set B of m points.*
- *There is a collection of L distinct* partitions $p_1, \ldots, p_L$.
- *For $1 \leq i \leq L$, partition p_i is a collection of k disjoint subsets whose union is B.*
- *Any cover of the m points by subsets that appear in pairwise different partitions requires at least d subsets.*

The following lemma is proved in [11].

Lemma 16 *For every $c \geq 0$ and m sufficiently large there is a* partition system $B(m, L, k, d)$ *whose parameters satisfy the following:*
- $L \simeq (\log m)^c$.
- *k can be chosen arbitrarily as long as* $k < \frac{\ln m}{3 \ln \ln m}$.
- $d = (1 - f(k))k \ln m$, *where $f(k) \to 0$ as $k \to \infty$.*

Moreover, a random collection of L partitions into k subsets with parameters chosen as above gives with high probability a partition system with $f(k) = 2/k$.

The randomized construction can be replaced by a deterministic construction (with a somewhat larger value for $f(k)$) using techniques developed in [29].

3.5 The construction

The input to the reduction is a 5-regular graph $G(V, E)$ for which we want to determine whether it is legally 3-colorable, or whether every 3-coloring of its vertices legally colors at most $\psi|E|$ edges. As noted in Proposition 15, for some explicit $\psi < 1$ this problem is NP-hard. The reduction uses the following parameters, which are chosen so that (k, l)-Hadamard codes exist and Lemma 16 holds:

- $k = l = c \log\log |V|$ for some sufficiently large constant c. We assume that l is a power of 2.
- $L = 3^l$, and $m = |V|^{\Theta(l)}$.

The output of the reduction is a bipartite graph $G'(V_1, V_2, E')$; when we say "color" below, we refer to a color-set of 3 colors. The left hand side vertex set V_2 is composed of $(2|E|)^l$ clusters of vertices, where each cluster contains m vertices. Each left hand side cluster is labeled by a sequence of l edges in G and a sequence of l bits; for each i, the ith bit in the bit-sequence denotes one endpoint (vertex) of the ith edge in the edge-sequence. Hence, this sequence of bits can also be viewed as a sequence of vertices. The right hand side vertex set V_1 is composed of k disjoint *rays*, and each ray is labeled by a codeword of the (k, l)-Hadamard code. Each ray is composed of $|V|^{l/2}|E|^{l/2}$ clusters, where each cluster contains 6^l vertices. Each right hand side cluster is labeled by a sequence of $l/2$ vertices in G and a sequence of $l/2$ edges in G. Equivalently, we may merge these two sequences to one sequence of length l, where the codeword of the ray containing the cluster is used as a selector function specifying the order in which vertices and edges are merged (a vertex when the corresponding bit in the codeword is 0, and an edge when the corresponding bit in the codeword is 1). This is called the merged label of a right hand side cluster. Individual vertices of right hand side clusters are further labeled by a sequence of $l/2$ colors (i.e., a ternary sequence), a sequence of $l/2$ pairs of distinct colors (i.e., a sequence in base 6), and a number between 1 and $2^{l/2}$. Simple counting shows that for each of the labeling schemes that we defined, the number of available labels is exactly equal to the number of objects that need to be labeled.

We say that a left hand side cluster and a right hand side cluster are *compatible* if their labels agree coordinate-wise in the following sense: for coordinate i, if the merged label of the right hand side cluster has an edge, then this is the ith edge in the sequence of edges labeling the left hand side cluster, and if the merged label has a vertex, then this is the ith vertex in the sequence of vertices labeling the left hand side cluster. Edges in G' only connect compatible clusters (compatibility is necessary but not sufficient, as will be seen shortly). Note that each left hand side cluster is compatible with exactly one cluster in each ray. Each right hand side cluster is compatible with $5^{l/2}2^{l/2}$ left hand side clusters; the term "5" here arises from the fact that G is 5-regular, and the term "2" follows from the fact that every edge has two end-points.

In order to describe the edge set E', we use the notion of a partition system. For each left hand side cluster C_ℓ, we have $L = 3^l$ partitions with properties as in Definition 4. (Recall that

C_ℓ has m elements as required by Definition 4.) Each partition is labeled by a sequence of l colors. This sequence of colors is interpreted as a sequence of colors for the sequence of vertices that label C_ℓ. Note that the same vertex of G may appear several times in the sequence of vertices that labels C_ℓ; we do not require the colors given to this vertex to be the same.

Consider an arbitrary vertex v in a cluster C_r that belongs to ray i. We now describe the set of neighbors that it has in a *compatible* left hand side cluster C_ℓ. Recall that v is labeled by a sequence of $l/2$ colors and a sequence of $l/2$ pairs of distinct colors. These colors give in a natural way a coloring for the merged sequence of vertices and edges labeling C_r. The vertex v was also labeled by a number between 1 and $2^{l/2}$; this label is ignored when determining the set of neighbors of v (that is, C_r has $2^{l/2}$ identical copies of v).

The coloring of the merged sequence of C_r induces in a natural way a coloring for the sequence of vertices labeling C_ℓ. This coloring labels one particular partition p. Vertex v is connected to all vertices (points) of the ith part of partition p (recall that i is the ray to which C_r belongs). This completes the description of E'.

Lemma 17 *If G is legally 3-colorable, then the one-sided domatic number of G' is 6^l.*

Proof. We first show that G' has a one-sided dominating set that includes exactly one vertex from every right hand side cluster. Consider a sequence of l arbitrary legal 3-colorings of G (the same legal 3-coloring may appear multiple times in the sequence). Consider now an arbitrary right hand side cluster. It is labeled by a length l merged sequence of vertices and edges. The sequence of legal 3-colorings induces a coloring on this merged sequence. The cluster contains exactly $2^{l/2}$ vertices whose label induces the same coloring. Select one of them arbitrarily to be included in the one-sided dominating set.

To show that indeed we have a one-sided dominating set, we need to show that every left hand side vertex u is covered. Consider the cluster C_ℓ to which u belongs. It is labeled by a sequence of l edges and a sequence of l vertices. The sequence of legal 3-colorings induces a coloring for the sequence of vertices. This coloring agrees with the name of one partition p. Let i be the part of partition p to which u belongs. Consider the right hand side cluster C_r that is compatible with C_ℓ and belongs to ray i. The vertex selected from C_r necessarily covers u.

We now show that the one-sided domatic number is at least 6^l (in fact, it is exactly 6^l). Consider an arbitrary legal 3-coloring of G. From it, we can derive 6^l distinct length l sequences of legal 3-colorings, where in each of the l coordinates we put one of the six permutations of the legal 3-coloring. Each of these sequences gives a one-sided dominating set as described above. We show that these one-sided dominating sets can be chosen to be distinct. This follows from the fact that for each sequence of legal 3-colorings and every right hand side cluster, we can have $6^{l/2}3^{l/2}$ equivalence classes of $2^{l/2}$ vertices (who differ only in the number they are given in the third label), and $6^{l/2}3^{l/2}$ equivalence classes of $2^{l/2}$ sequences of colorings (who differ only in the way they color vertices not in the merged sequence of the right hand side cluster). Preserving the structure of the equivalent classes, we can match the sequences of legal 3-colorings with the vertices of a right hand side cluster. $\square$

Lemma 18 *If every 3-coloring of G legally colors at most $\psi|E|$ edges, than the smallest one-sided dominating set in G' is of cardinality at least $(1-o(1))|V_1|\ln m/6^l$.*

Proof. The proof is essentially identical to that of [11] and is omitted. $\square$

By making m sufficiently large, we can have $\ln m \simeq \ln|V_2|$, and the proof of Theorem 11 follows.

Acknowledgments

This work appears in preliminary form in [12] and [38].

We would like to thank Satoshi Fujita and Mario Szegedy for helpful discussions. The first author is the Incumbent of the Joseph and Celia Reskin Career Development Chair. The research of the first author was supported in part by a Minerva grant. The second author would like to thank Kazuo Iwama at Kyoto University for his hospitality.

References

[1] N. Alon and J. Spencer. *The Probabilistic Method*. Wiley, 1992.

[2] S. Arnborg, J. Lagergren, and D. Seese. Easy problems for tree-decomposable graphs. *Journal of Algorithms*, 12(2):308–340, June 1991.

[3] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and intractability of approximation problems. *Journal of the ACM* 45(3):501–555, 1998.

[4] J. Beck. An algorithmic approach to the Lovász Local Lemma. *Random Structures & Algorithms*, 2:343–365, 1991.

[5] C. Berge. Balanced matrices. *Math. Programming*, 2:19–31, 1972.

[6] M. A. Bonucelli. Dominating sets and domatic number of circular arc graphs. *Discrete Appl. Math.*, 12:203–213, 1985.

[7] E. J. Cockayne and S. T. Hedetniemi. Towards a theory of domination in graphs. *Networks*, 7:247–261, 1997.

[8] P. Crescenzi and V. Kann. A compendium of NP optimization problems. `http://www.nada.kth.se/theory/problemlist.html`, 2000.

[9] P. Erdős and L. Lovász. Problems and results on 3-chromatic hypergraphs and some related questions. In A. Hajnal et al., editor, *Infinite and Finite Sets*, volume 11 of *Colloq. Math. Soc. J. Bolyai*, pages 609–627. North Holland, Amsterdam, 1975.

[10] M. Farber. Domination, independent domination, and duality in strongly chordal graphs. *Discrete Appl. Math.*, 7:115–130, 1984.

[11] U. Feige. A threshold of $\ln n$ for approximating set cover. *J. ACM*, 45(2):634–652, 1998.

[12] U. Feige, M. M. Halldórsson, and G. Kortsarz. Approximating the domatic number. In *Proc. 32nd Ann. ACM Symp. on Theory of Computing*, pages 134–143, 2000.

[13] U. Feige and J. Kilian. Zero knowledge and the chromatic number. *J. Comput. Syst. Sci.*, 57:187–199, 1998.

[14] S. Fujita. On the performance of greedy algorithms for finding maximum r-configurations. In *Korea-Japan Joint Workshop on Algorithms and Computation (WAAC99)*, 1999.

[15] S. Fujita, M. Yamashita, and T. Kameda. A study on r-configurations – a resource assignment problem on graphs. *SIAM J. Disc. Math.*, 13(2):227–254, 2000.

[16] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness.* Freeman, 1979.

[17] O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proofs. *J. ACM*, 38(3):691–729, 1991.

[18] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater. *Domination in Graphs: Advanced Topics.* Marcel Dekker, 1998.

[19] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater. *Fundamentals of Domination in Graphs.* Marcel Dekker, 1998.

[20] D. S. Johnson. Approximation algorithms for combinatorial problems. *J. Comput. Syst. Sci.*, 9:256–278, 1974.

[21] H. Kaplan and R. Shamir. The domatic number problem on some perfect graph families. *Inf. Process. Lett.*, 49(1):51–56, 1994.

[22] S. Khanna, M. Sudan, and D. Williamson. A complete classification of the approximability of maximization problems derived from Boolean constraint satisfaction. In *Proc. 29th Ann. ACM Symposium on Theory of Computing*, pages 11–20, 1997.

[23] J. Kratochvíl. Regular codes in regular graphs are difficult. *Discrete Math.*, 133:191–205, 1994.

[24] L. Lovász. On the ratio of optimal integral and fractional covers. *Discrete Math.*, 13:383–390, 1975.

[25] A. Lubotzky, R. Phillips and P. Sarnak. Ramanujan graphs. Combinatorica, 8:261–277, 1988.

[26] C. Lund and M. Yannakakis. On the hardness of approximating minimization problems. *J. ACM*, 41(5):960–981, 1994.

[27] F. MacWilliams and N. Sloane. *The Theory of Error Correcting Codes.* North Holland, 1983.

[28] M. V. Marathe, H. B. Hunt III, and S. S. Ravi. Efficient approximation algorithms for domatic partition and on-line coloring of circular arc graphs. *Discrete Appl. Math.*, 64:135–149, 1996.

[29] M. Naor, L. Schulman, and A. Srinivasan. Splitters and near-optimal derandomization. In *Proc. 36th Ann. IEEE Symp. on Found. of Comp. Sci.*, pages 182–191, 1995.

[30] C. H. Papadimitriou and M. Yannakakis. Optimization, approximation and complexity classes. In *Proc. 20th Ann. ACM Symp. on Theory of Computing*, pages 229–234, 1988.

[31] A. Paz and S. Moran. Nondeterministic polynomial optimization problems and their approximations. *Theoretical Comput. Sci.*, 15:251–277, 1981.

[32] S. L. Peng and M. S. Chang. A simple linear time algorithm for the domatic partition problem on strongly chordal graphs. *Inf. Process. Lett.*, 43:297–300, 1992.

[33] E. Petrank. The hardness of approximation: Gap location. *Comput. Complexity*, 4:133–157, 1994.

[34] A. Proskurowski and J. A. Telle. Complexity of graph covering problems. *Nordic J. Computing*, 5:173–195, 1998.

[35] P. Raghavan. Probabilistic construction of deterministic algorithms: approximating packing integer programs. *J. Comput. Syst. Sci.*, 37:130–143, 1988.

[36] R. Raz. A parallel repetition theorem. *SIAM J. Comput.*, 27(3):763–803, 1998.

[37] A. S. Rao and C. P. Rangan. Linear algorithm for domatic number problem on interval graphs. *Inf. Process. Lett.*, 33:29–33, 1989.

[38] A. Srinivasan. Domatic partitions and the Lovász Local Lemma. In *Proc. Twelfth ACM-SIAM Symp. on Discrete Algorithms*, 2001. To appear.

[39] B. Zelinka. Domatic number and degree of vertices of a graph. *Math. Slovaca*, 33:145–147, 1983.

A The domatic number of random graphs

We now show that the bound (1) is tight for a large range of values of $\delta = \delta(n)$, by studying $D(G)$ for suitable random graphs G. Suppose G is drawn from the random graph model $G(n, p)$: i.e., we take n labeled vertices, and put an edge with probability p independently between each pair of vertices. We will show that with probability $1 - o(1)$, $D(G) \leq (1 + o(1))\delta(G)/\ln \Delta(G)$. (Throughout this section, the "$o()$" and "$\omega()$" notation refers to n getting large.)

Choose any $p = p(n)$ such that $np = (\ln n)^{\omega(1)}$ and $p = o(1)$. It is easy to check via a Chernoff bound that with probability $1 - o(1)$, both δ and Δ will lie in the range $(1 \pm o(1))np$ for our random graph G. Fix any constant $\epsilon > 0$, and let $s = \lfloor (1 - \epsilon) \ln(np)/p \rfloor$. We will show that with probability $1 - o(1)$, any dominating set in G will have size more than s. (Thus, with probability $1 - o(1)$, we will have $D(G) \leq n/(s+1)$, completing the proof.) For any given subset of the vertices S with $|S| = s$,

$$
\begin{aligned}
\Pr[S \text{ is a dominating set}] &= (1 - (1 - p)^s)^{n-s} \\
&\leq (1 - e^{-(1+\Theta(p))sp})^{n-s} \\
&\leq (1 - (np)^{\epsilon - 1 - \Theta(p)})^{n-s} \\
&\leq e^{-(n-s)\cdot(np)^{\epsilon-1-\Theta(p)}} \\
&\leq e^{s-(1/p)\cdot(np)^{\epsilon-\Theta(p)}} \\
&= e^{s-(1/p)\cdot(np)^{\epsilon-o(1)}}.
\end{aligned}
$$

Thus, the probability of existence of a dominating set of size s is at most

$$\binom{n}{s} \cdot e^{s-(1/p)\cdot(np)^{\epsilon-o(1)}} \leq (n^s/s!) \cdot e^{s-(1/p)\cdot(np)^{\epsilon-o(1)}} = (e^s/s!) \cdot e^{s\ln n-(1/p)\cdot(np)^{\epsilon-o(1)}} = o(1);$$

the bound $s \ln n = o((1/p) \cdot (np)^{\epsilon - o(1)})$ follows from the definition of s and from the fact that $np = (\ln n)^{\omega(1)}$.

B Multicoloring version

In the domatic multi-partition problem, we are additionally given an integral weight $x : V \mapsto N$ indicating in how many dominating sets each vertex can appear. The domatic number problem has $x(v) = 1$, for each v; the r-*Conf* problem of [15] has $x(v) = r$, for each v.

We can reduce the multi-partition problem to the ordinary partition problem. Given a graph G and weight vector x, form a graph G' as follows. G' has $x(v)$ copies of each vertex v connected as a clique. For each edge uv in G, the copies of u and v form a complete bipartite graph in G'. Any minimal dominating set in G' is also a dominating set in G, and taking one copy of each vertex of a dominating set in G also gives a dominating set in G'. Further, a domatic partition of G' is in one-to-one correspondence with a multi-partition of G (within the weight constraints). Thus, the results obtained in this paper for the domatic number problem carry over to the domatic multi-partition problem, replacing δ by $\min_v d(v)x(v)$ and n by $\sum_v x(v)$.

C Overview of some ingredients from [11]

Parallel repetition of two-prover proof systems: There is a straightforward one-round two-prover proof system for Max 3-colorability-5 that has the following properties: on *yes* instances, the verifier always accepts, and on *no* instances (when at most a $(1 - \epsilon)$ fraction of the edges can be legally colored simultaneously) the verifier accepts with probability at most $1 - \epsilon/2$. Give a 5-regular graph G, the proof system proceeds as follows. The verifier sends to the first prover a random edge in G, and to the second prover a random vertex from that edge. We call this vertex the *common* vertex. It is important that the first prover does not know which endpoint of the edge is the common vertex, and that the second prover does not know which edge was received by the first prover. The first prover replies with two different colors (out of the three allowable colors) for the two vertices that are the endpoints of the edge. The second prover replies with a color for the common vertex. The verifier accepts only if the two provers give the same color to the common vertex. For a *yes* instance G the two provers can answer according to the global legal 3-coloring and ensure that the verifier accepts with probability 1. For a *no* instance G, regardless of the strategy of each prover (where a strategy is a function from questions to answers), the acceptance probability is at most $1 - \epsilon/2$, where probability is computed over the random choices of the verifier.

The l-fold parallel repetition of this one-round two-prover proof system is as follows. The verifier sends to the first prover a tuple of l random edges and to the second prover a tuple of l random vertices, one from each of these edges. Each prover replies with colors to all the vertices that it receives. The verifier accepts if on every one of the l common vertices, the two provers agree on the color that they give. It is not hard to see that on *yes* instances, the provers still have a strategy that makes the verifier accept with probability 1. The parallel repetition theorem [36] shows that on *no* instances, the verifier accepts with probability at most $(1 - \epsilon)^{cl}$, where c is a constant (that depends on ϵ).

Separating codes and k-provers: To eventually get $\simeq \ln n$ hardness of approximation results, two-prover proof systems are extended to k-prover systems in a special way. As in the two-prover proof system, the verifier selects l edges at random and a random vertex within each edge. This gives a total of $(2|E|)^l$ possible queries. Each of the k provers is sent some subset of the l edges and l vertices using the following rule. A (k, l)-Hadamard code is a collection of k binary words of length l each, for which the Hamming distance between any pair of words is "large" (specifically, $l/2$). With each prover we associate one codeword from the Hadamard code. The verifier sends to each prover $l/2$ edges and $l/2$ vertices, selected according to the 1's and 0's in the codeword of the respective prover. Namely, if the ith bit in the code is 1, the next entry in the query is the respective ith edge (among the l edges in the tuple). If the ith entry is 0, the ith vertex in the vertex tuple is sent. Considering separately two of the provers, the fact that their codewords have a large Hamming distance implies that in many coordinates one prover will receive an edge while the other prover will receive a vertex on this edge. This is similar to the scenario in the two-prover proof system. Intuitively, we may view the k-prover proof system as $\binom{k}{2}$ correlated two-prover proof systems going on in parallel. It is natural to have the verifier accept if all $\binom{k}{2}$ proof systems are accepting. The definition of acceptance used in [11] is more subtle and will not be discussed in this overview.

Next a bipartite graph $G'(V_1, V_2, E')$ is built based on this scenario. Specifically, the vertices in V_1 describe the provers/queries schema. The set V_1 corresponds to the provers and is partitioned into k "rays" V_1^j, $1 \leq j \leq k$, one ray for each prover. The sets V_1^k are further partitioned into disjoint subsets $V_1^j(Q)$, where Q ranges over all the $|V|^{l/2}|E|^{l/2}$ possible questions that a prover can receive. On each possible question, the prover must answer with a coloring of the $l/2$ edges and $l/2$ vertices received. Thus, the $V_1^j(Q)$ sets are further divided into all possible answers. Therefore, there are $3^{l/2} \cdot 6^{l/2}$ points inside $V_1^j(Q)$ in this final division; a point for each possible coloring of the $l/2$ vertices and $l/2$ edges. In addition, for technical reasons, each point (possible answer) of each prover is duplicated $2^{l/2}$ times.

Ground sets and random partitions: To determine the structure of V_2, some ideas essentially due to [26] (and extended in [11]) are used.

The set V_2 is partitioned into $(2|E|)^l$ *ground sets* C_ℓ, one ground set per each possible query. Each ground set contains m points for $m = |V|^{\Theta(l)}$. It is instructive to note the following asymmetry. Given a query namely a sequence of l edges and l corresponding endpoints for the edges, consider its corresponding ground set in V_2. There are exactly k sets $V_1^j(Q)$ that are compatible with this sequence, one per prover (since the question to the provers is completely determined by the Hadamard code). On the other hand, given a question cluster $V_1^j(Q)$, many possible query-base ground sets in V_2 could have caused this question. Indeed, since the question contains only $l/2$ edges and $l/2$ vertices, this partial information can be completed in $5^{l/2}2^{l/2}$ ways to give compatible l edges and l vertices. Each vertex sent to this prover has 5 neighbors and thus 5 ways of "completing" this vertex into a (compatible) edge. For each edge sent to this prover, there are 2 possibilities for which of the two vertices to put in the resulting query.

We use $L = 3^l$ random partitions of C_ℓ, each partitioning C_ℓ into k parts. Consider the l-tuple of vertices in a possible query (namely, the part of the query corresponding to the chosen vertices, one per each edge). Note that the number $L = 3^l$ of partitions of each ground set exactly equals the number of possible colorings 3^l of the vertices in that query. We can thus form a 1-1 correspondence between the 3^l partitionings and the 3-coloring (not necessarily a consistent one) of the l vertices in the query base. Each partition is now regarded as a coloring, and vice-versa. Recall that a point in the final sub-partition cluster of V_1 corresponds to an answer to the coloring of the query of $l/2$ edge/vertex pairs. This corresponds to a coloring of the l vertices of the cluster C_ℓ, hence gives a vertex coloring, and thus a partition of C_ℓ. The edges between the vertices of V_1 and V_2 are defined as follows. Each vertex (namely, coloring or answer) $v \in V_1^j(Q)$ is connected to the jth part of the partition corresponding to v (as v is a coloring, a partition is also associated with v).

Now, consider the graph G' resulting from a 3-colorable graph G. We can choose one point (answer) per each question-cluster in V_1, so as to cover each $C\ell$. Thus all the C_ℓ are covered using at most $k \cdot (2 \cdot E)^l$ vertices. This will happen if the provers use strategies which are all part of a global 3-coloring, and this gives a small dominating set. On the other hand, consider the resulting graph G' when only $1-\epsilon$ of the edges of G can be simultaneously legally colored. Because of the parallel repetition which greatly increases the gap, the provers essentially cannot use a joint strategy. This means that each pair of answers $v_j \in V_1^j(Q), v_q \in V_1^q(Q)$ chosen, corresponds to two different colorings. Hence a ground set C_ℓ is essentially covered via random sets each containing m/k random elements of C_ℓ. Thus, the number of elements needed in order to cover C_ℓ is roughly $k \ln m$. This gives a total of roughly $k \cdot (2 \cdot E)^l \ln |V'|$ minimum one-sided dominating set for a logarithmic gap.

Zero knowledge and the domatic number: Given the fact that a single "small" one-sided dominating set exists, we can permute the colors (as done in zero-knowledge protocols in order to "hide" information) to show the existence of "many" disjoint dominating sets, hence a "large" domatic number. Namely, once a 3-coloring of G exists, one gets 6 legal colorings for G. This implies a total of 6^l colorings (which can be completed into global coordinatewise-compatible colorings) of any l tuple of vertices. The possibility for color permutation was the main reason for reducing from a coloring and not a satisfiability problem. Hence, we either get many small dominating sets which implies a large one-sided domatic number, or any dominating set is large, which gives a a small one-sided domatic number.