Electronic Colloquium on Computational Complexity, Revision 2 of Report No. 132 (2015)

On Percolation and $\mathcal{NP}$-Hardness

Huck Bennett * Daniel Reichman † Igor Shinkar *

July 13, 2016

Abstract

The edge-percolation and vertex-percolation random graph models start with an arbitrary graph G, and randomly delete edges or vertices of G with some fixed probability. We study the computational hardness of problems whose inputs are obtained by applying percolation to worst-case instances. Specifically, we show that a number of classical $\mathcal{NP}$-hard graph problems remain essentially as hard on percolated instances as they are in the worst-case (assuming $\mathcal{NP} \nsubseteq \mathcal{BPP}$). We also prove hardness results for other $\mathcal{NP}$-hard problems such as Constraint Satisfaction Problems and Subset-Sum, with suitable definitions of random deletions.

We focus on proving the hardness of the Maximum Independent Set problem and the Graph Coloring problem on percolated instances. To show this we establish the robustness of the corresponding parameters $\alpha(\cdot)$ and $\chi(\cdot)$ to percolation, which may be of independent interest. Given a graph G, let G' be the graph obtained by randomly deleting edges of G. We show that if $\alpha(G)$ is small, then $\alpha(G')$ remains small with probability at least 0.99. Similarly, we show that if $\chi(G)$ is large, then $\chi(G')$ remains large with probability at least 0.99.

1 Introduction

The theory of $\mathcal{NP}$-hardness suggests that we are unlikely to find optimal solutions to $\mathcal{NP}$-hard problems in polynomial time. This theory applies to the worst-case setting where one considers the worst running-time over all inputs of a given size. It is less clear whether these hardness results apply to "real-life" instances. One way to address this question is to examine to what extent known $\mathcal{NP}$-hardness results are stable under random perturbations, as it seems reasonable to assume that a given instance of a problem may be subjected to noise.

Recent work has studied the effect of random perturbations of the input on the runtime of algorithms. In their seminal paper Spielman and Teng [33] introduced the idea of *smoothed analysis* to explain the superior performance of algorithms in practice compared with formal worst-case bounds. Roughly speaking, smoothed analysis studies the running time of an algorithm on a perturbed worst-case instance. In particular, they showed that subjecting the weights of an arbitrary linear program to Gaussian noise yields instances on which the simplex algorithm runs in expected polynomial time, despite the fact that there are pathological linear programs for which the simplex algorithm requires exponential time. Since then smoothed analysis has been applied to a number of other problems [13, 34].

*Department of Computer Science, Courant Institute of Mathematical Sciences, New York University. Email: hbennett@cs.nyu.edu, ishinkar@cims.nyu.edu

†Electrical Engineering and Computer Science, University of California Berkeley, Berkeley, CA, USA. Email: daniel.reichman@gmail.com.

ISSN 1433-8092

In contrast to smoothed analysis, we study when worst-case instances of problems remain hard under random perturbations. Specifically, we study to what extent $\mathcal{NP}$-hardness results are robust when instances are subjected to random deletions. Previous work is mainly concerned with *Gaussian* perturbations of *weighted* instances. Less work has examined the robustness of hardness results of unweighted instances with respect to discrete noise.

We focus on two forms of percolation on graphs. Given a graph $G = (V, E)$ and a parameter $p \in (0, 1)$, we define $G_{p,e} = (V, E')$ as the probability space of graphs on the same set of vertices, where each edge $e \in E$ is contained in E' independently with probability p. We say that $G_{p,e}$ is obtained from G by *edge percolation*. We define $G_{p,v} = (V', E')$ as the probability space of graphs, in which every vertex $v \in V$ is contained in V' independently with probability p, and $G_{p,v}$ is the subgraph of G induced by the vertices V'. We say that $G_{p,v}$ is obtained from G by *vertex percolation*. We also study appropriately defined random deletions applied to instances of other $\mathcal{NP}$-hard problems, such as 3-SAT and Subset-Sum.

Throughout we refer to instances that are subjected to random deletions as *percolated instances*. Our main question is whether such percolated instances remain hard to solve by polynomial-time algorithms assuming $\mathcal{NP} \nsubseteq BPP$.

The study of random discrete structures has resulted with a wide range of mathematical tools which have proven instrumental in proving rigorous results regarding such structures [5, 17, 19, 26]. One reason for studying percolated instances is that it may offer the opportunity to apply these methods to a broader range of distributions of instances of $\mathcal{NP}$-hard problems beyond random graphs and random formulas. Furthermore, percolated instances are studied in host of different disciplines and are frequently used to explain properties of the the WWW [1, 8], hence it is of interest to understand the computational complexity of instances arising from percolated networks.

1.1 A first example – 3-Coloring

Consider the 3-Coloring problem, where given a graph $G = (V, E)$ we need to decide whether G is 3-colorable. Suppose that given a graph G we sample a random subgraph G' of G, by deleting each edge of G independently with probability $p = \frac{1}{2}$, and ask whether the resulting graph is 3-colorable. Is there a polynomial time algorithm that can decide with high probability whether G' is 3-colorable?

We demonstrate that a polynomial-time algorithm for deciding whether G' is 3-colorable is impossible assuming $\mathcal{NP} \nsubseteq BPP$. We show this by considering the following polynomial time reduction from the 3-Coloring problem to itself.

We will need the following definition.

Definition 1.1. *Given a graph G, the R-blowup of G is a graph $G' = (V', E')$, where every vertex v is replaced by an independent set $\widetilde{v}$ of size R, which we call the cloud corresponding to v. We then connect the clouds $\widetilde{u}$ and $\widetilde{v}$ by a complete $R \times R$ bipartite graph $(u, v) \in E$.*

Given an n-vertex graph H the reduction outputs a graph G that is an R-blow-up of H for $R = C\sqrt{\log(n)}$, where $C > 0$ is large enough. It is easy to see that H is 3-colorable if and only if G is 3-colorable.

In fact, the foregoing reduction satisfies a stronger robustness property for random subgraphs G' of G obtained by deleting edges of G with probability $\frac{1}{2}$. Namely, if H is 3-colorable, then G is 3-colorable, and hence G' is also 3-colorable with probability 1. On the other hand, if H is not 3-colorable, then G is not 3-colorable, and with high probability G' is not 3-colorable either.

Indeed, for any edge (v_1, v_2) in H let U_1, U_2 be two clouds in G corresponding to v_1 and v_2. Fixing two arbitrary sets $U_1' \subseteq U_1$ and $U_2' \subseteq U_2$ each of size $R/3$, the probability that there is no edge in G' connecting a vertex from U_1 to a vertex in U_2 is $2^{-R^2/9}$. By union bounding over the $|E| \cdot \binom{R}{R/3}^2 \ll 2^{R^2/9}$ choices of U_1', U_2' we get that there is at least one edge between U_1' and U_2' with high probability. When this holds we can decode any 3-coloring of G' to a 3-coloring of H by coloring each vertex v of H with the color that appears the largest number of times in the coloring of the corresponding cloud in G', breaking ties arbitrarily. Therefore, a polynomial time algorithm for deciding the 3-colorability of G' implies a polynomial time algorithm for determining the 3-colorability of H with high probability, and hence unless $\mathcal{NP} \subseteq \text{co}\mathcal{RP}$ there is no polynomial time algorithm that given a 3-colorable graph G finds a 3-coloring of a random subgraph of G.[1]

Toward a stronger notion of robustness

The example above raises the question of whether the blow-up described above is really necessary. Naïvely, one could hope for stronger hardness of the 3-Coloring problem, namely, that for any graph H if H is not 3-colorable, then with high probability a random subgraph H' of H is not 3-colorable either. However, this is not true in general, as H can be a 3-critical graph, i.e., a 3-colorable graph such that deletion of *any* edge of H decreases its chromatic number (consider for example the case of an odd cycle).

Nonetheless, if random deletions do not decrease the chromatic number of a graph by much, then one could use hardness of approximation results on the chromatic number to deduce hardness results for coloring percolated graphs. This naturally leads to the following question.

Question. Let G be an arbitrary graph, and let G' be a random subgraph of G obtained from G by deleting each edge of G with probability $1/2$. Is it true that if $\chi(G)$ is large, then $\chi(G')$ is also large with probability at least 0.99?[2]

In this paper we give a positive answer to this question and show that in some sense the chromatic number of any graph is robust against random deletions. We also consider the question of robustness for other graph parameters. For independent sets we demonstrate that if the independence number of G is small, then with high probability the independence number of a random subgraph of G is small as well. Hardness results are derived for other graph-theoretic problems such as Minimum Vertex Cover and Hamiltonian Cycle. Similarly, we show that for CSP formulas that are sufficiently dense randomly deleting its clauses does not change the maximum possible fraction of clauses that can be satisfied simultaneously. In particular, this implies that these problems remain essentially as hard on percolated instances as they are on worst-case instances.

Remark. *It is worth noting that there are graph parameters for which hardness on percolated instances differs significantly from hardness on the original instance. For example, standard results in random graph theory imply that for every n-vertex graph G, with high probability the size of the largest clique in the graph G' obtained by edge percolation with $p = \frac{1}{2}$ is $O(\log n)$. In particular, a*

[1] Note that in the foregoing example, if we start with a bounded degree graph H, we can reduce it to a bounded degree graph G by using an $R \times R$ bipartite expander instead of the complete bipartite graph.

[2] We note that if instead of choosing the subgraph G' at random, we choose an *arbitrary* subgraph of G with $|E|/2$ edges, then it is possible that $\chi(G')$ is much smaller than $\chi(G)$. For example, consider the n-vertex graph $G = (V, E)$ that consists of a clique of size $n/3$ and a complete bipartite graph with $n/3$ edges on each side. Then $\chi(G) = n/3$, whereas if we remove all the edges of the $n/3$-clique, the graph becomes 2-colorable, while the number of removed edges is $\binom{n/3}{2} < n^2/18 < |E|/2$.

maximum clique in G' can be found in time $n^{O(\log n)}$, which is significantly faster than the fastest known algorithm for finding a maximum clique in the worst-case.

1.2 Robustness of $\mathcal{NP}$-hard problems under percolation

In proving hardness results for percolated instances we use the concept of *robust reductions* which we explain below. It will be convenient to consider promise problems[3]. We start by introducing the following definition.

Definition 1.2. *Let $A = (A_{YES}, A_{NO})$ and $B = (B_{YES}, B_{NO})$ be two promise problems. For each $y \in \{0,1\}^*$ (an instance of the problem B) let $\mathsf{noise}(y)$ be a distribution on $\{0,1\}^*$, that is samplable in time that is polynomial in $|y|$.*

- *A polynomial time reduction R from A to B is said to be* noise-robust *if*
  1. *For all $x \in A_{YES}$ it holds that $\mathsf{R}(x) \in B_{YES}$, and $\Pr[\mathsf{noise}(\mathsf{R}(x)) \in B_{YES}] > 0.99$.*
  2. *For all $x \in A_{NO}$ it holds that $\mathsf{R}(x) \in B_{NO}$, and $\Pr[\mathsf{noise}(\mathsf{R}(x)) \in B_{NO}] > 0.99$.*
- *If in the first item we have $\Pr[\mathsf{noise}(\mathsf{R}(x)) \in B_{YES}] = 1$, then we say that R is a noise-robust co$\mathcal{RP}$-reduction. Similarly, if in the second item we have $\Pr[\mathsf{noise}(\mathsf{R}(x)) \in B_{NO}] = 1$, then we say that R is a noise-robust $\mathcal{RP}$-reduction.*
- *The problem $B = (B_{YES}, B_{NO})$ is said to be $\mathcal{NP}$-*hard under a noise-robust reduction *if there exists a* noise-robust *reduction from an $\mathcal{NP}$-hard problem to B.*
- *We say that the problem A is* strongly-noise-robust *to B if*
  1. *For all $x \in A_{YES}$ it holds that $x \in B_{YES}$, and $\Pr[\mathsf{noise}(x) \in B_{YES}] > 0.99$.*
  2. *For all $x \in A_{NO}$ it holds that $x \in B_{NO}$, and $\Pr[\mathsf{noise}(x) \in B_{NO}] > 0.99$.*

We use the term noise-robust to avoid confusion with other notions of robust reductions that have appeared in the literature. In order to ease readability, we will often write robust reductions instead, always referring to noise-robust reductions as defined above.

Note that in the last item of Definition 1.2 there is no reduction involved. Instead, we think of the problem A as a relaxation of B with $A_{YES} \subseteq B_{YES}$ and $A_{NO} \subseteq B_{NO}$, and hence any algorithm that solves B in particular solves A. However, the relaxation is also robust: after applying noise to a YES-instance (resp. NO-instance) of A, it stays a YES-instance (resp. NO-instance) of B with high probability.

Proposition 1.3. *Let $L = (L_{YES}, L_{NO})$ be a promise problem, and for each y instance of L, let $\mathsf{noise}(y)$ be a distribution on instances of L that is samplable in time that is polynomial in $|y|$.*

If L is $\mathcal{NP}$-hard under a noise-robust reduction, then there is no polynomial time algorithm that when given an input y decides with high probability whether $\mathsf{noise}(y) \in L_{YES}$ or $noise(y) \in L_{NO}$, unless $\mathcal{NP} \subseteq \mathcal{BPP}$.

[3]Recall, that a promise problem is a generalization of a decision problem, where for the problem L there are two disjoint subsets L_{YES} and L_{NO}, such that an algorithm that solves L must accept all the inputs in L_{YES} and reject all inputs in L_{NO}. If the input does not belong to $L_{YES} \cup L_{NO}$, there is no requirement on the output of the algorithm.

Indeed, the example given in Section 1.1 gives a noise-robust reduction from the 3-Coloring problem to itself, where noise refers to random deletions of the edges in a given graph (edge percolation). Therefore, the 3-Coloring problem is $\mathcal{NP}$-hard under a noise-robust reduction.

1.3 Our results

In this paper we show that a number of $\mathcal{NP}$-hard problems remain hard to solve even after random deletions, i.e., they are $\mathcal{NP}$-hard under noise-robust reductions. Furthermore, we show that some gap $\mathcal{NP}$-hard problems are, in fact, strongly-noise-robust to the same problems with a smaller gap. Specifically, we focus on showing these results for the gap versions of the maximum independent set and chromatic number problems. As technical tools, we prove a number of combinatorial results about the independence number and the chromatic number of percolated graphs that might be of independent interest.

Maximum Independent Set and Percolation

Theorem 1.4. *Let $G = (V, E)$ be an n-vertex graph. Then, with high probability $\alpha(G_{p,e}) \leq O\left(\frac{\alpha(G)}{p}\log(np)\right)$.*

We observe that in general, the upper bound above cannot be improved, as it is well known that the independence number of $G(n,p)$ is $\Omega\left(\frac{\log(np)}{p}\right)$ with high probability (see, e.g., [5]).

In the Coloring-vs-MIS(q, a) problem, given a graph G the goal is to distinguish between the YES-case where $\chi(G) \leq q$ and the NO-case where $\alpha(G) \leq a$. By using Theorem 1.4 together with inapproximability results of Feige and Kilian [14] we obtain the following hardness result.

Theorem 1.5. *For any q, a the Coloring-vs-MIS(q, a) problem is strongly-noise-robust to Coloring-vs-MIS$(q, O\left(\frac{a}{p}\log(np)\right))$, where n denotes the number of vertices in the given graph, and noise is edge-percolation with probability p.*

In particular, for any constant $\varepsilon > 0$, there is no polynomial time algorithm that given an n-vertex graph G approximates either $\alpha(G_{p,e})$ or $\chi(G_{p,e})$ within a $\frac{1}{pn^{1-2\varepsilon}}$ (resp. $pn^{1-2\varepsilon}$) factor for any $p > \frac{1}{n^{1-2\varepsilon}}$ unless $\mathcal{NP} \subseteq \mathcal{BPP}$.

We also prove analogous theorems for vertex percolation.

Graph Coloring and Percolation Theorem 1.4 says that it is hard to approximate the chromatic number of a percolated graph within a $n^{1-\varepsilon}$ factor, but says nothing about hardness of coloring percolated graphs with small (constant) chromatic number. We address this question below by proving lower bounds[4] on the chromatic number of percolated graphs. To do this we use results from additive combinatorics and discrete Fourier analysis.

Theorem 1.6. *Let $G = (V, E)$ be an n-vertex graph, and let $k = \chi(G)$. Then, for every $\lambda > 0$ it holds that $\Pr[\chi(G_{\frac{1}{2},v}) \geq k/2 - \lambda\sqrt{k}] > 1 - e^{-\lambda^2/2}$.*

Theorem 1.7. *Let $G = (V, E)$ be a graph with m edges, and let $k = \chi(G)$. Then, for every $\alpha \in (0, 1)$ it holds that $\Pr[\chi(G_{\frac{1}{2},e}) \geq \max\{\Omega_\alpha(k^{1/3}), \Omega_\alpha(k/m^{1/4})\}] > 1 - \alpha$.*

[4] The notation $O_\alpha(f(n))$ (resp. $\Omega_\alpha(f(n))$) means that $O(f(n))$ (resp. $\Omega(f(n))$) holds for fixed α.

In Theorem 1.7, the $\Omega_\alpha(\chi(G)/m^{1/4})$ lower bound is better when $\chi(G) = \omega(m^{3/8})$, and the $\Omega_\alpha(\chi(G)^{1/3})$ lower bound is better when $\chi(G) = o(m^{3/8})$.

We also obtain lower bounds on the chromatic numbers of $G_{p,v}$ and $G_{p,e}$ for $p < \frac{1}{2}$ by composing the bounds in Theorems 1.6 and 1.7 $\lceil \log_2(1/p) \rceil$ times.

In the Gap-Coloring(q, Q) problem we are given an n-vertex graph G and the goal is to distinguish between the YES-case where G is q-colorable, and the NO-case where the chromatic number of G is at least Q. There is a large body of work proving hardness results for this problem [18, 23, 21] including stronger results assuming variants of the Unique Games Conjecture [10, 12]. The strongest $\mathcal{NP}$-hardness result of this form, due to Huang [21], shows that Gap-Coloring$(q, \exp(\Omega(q^{1/3}))$ is $\mathcal{NP}$-hard. Combining this with Theorem 1.7 we obtain an analogous hardness result under noise-robust reductions for this problem.

Theorem 1.8. *For all $q < Q$ the Gap-Coloring(q, Q) problem is strongly-noise-robust to the Gap-Coloring$(q, \Omega(Q^{1/3}))$ problem, where noise is $\frac{1}{2}$-edge-percolation applied to the graph.*

In particular, for any sufficiently large constant q given a q-colorable graph G it is $\mathcal{NP}$-hard to find a $\exp(\Omega(q^{1/3}))$-coloring of $G_{\frac{1}{2},e}$ with high probability.

Satisfiability and other CSP's We prove hardness of approximating the value of percolated k-CSP instances. An instance Φ of k-CSP over some alphabet Σ (e.g. $\Sigma = \{0, 1\}$) is a formula consisting of a collection of clauses $C_1, ..., C_m$ over n variables $x_1, ..., x_n$ taking values in Σ, where each clause is associated with some k-ary predicate $f : \Sigma^k \to \{0, 1\}$ over variables $x_{i_1}, \ldots, x_{i_k}$. An instance Φ is said to be *simple* if all clauses in Φ are distinct. Given an assignment $\sigma : \{x_1, ..., x_n\} \to \Sigma$ we say that the constraint C on the variables $x_{i_1}, \ldots, x_{i_k}$ is a satisfied by σ if $f_C(\sigma(x_{i_1}), ..., \sigma(x_{i_k})) = 1$, where f_C is the predicate corresponding to C. Given a formula Φ, and an assignment σ to its variables the value of Φ with respect to the assignment σ, denoted by $\mathrm{val}_\sigma(\Phi)$, is the fraction of constraints of Φ satisfied by σ. The value of Φ is defined as $val(\Phi) = \max_\sigma \mathrm{val}_\sigma(\Phi)$. If $\mathrm{val}(\Phi) = 1$ we say that Φ is satisfiable.

We are typically interested in CSP where constraints belong to some fixed family of predicates $\mathcal{F}$. For example, in the k-SAT problem, the constraints are all of the form $f(z_1, \ldots, z_k) = \bigvee_{i=1}^k (z_i = b_i)$, for $b_1, \ldots, b_k \in \{0, 1\}$. We assume that k, the arity of the constraints, is some fixed constant that does not depend on the number of variables n.

These definitions give rise to the following optimization problem. Given a CSP instance Φ find an assignment that maximizes the value of Φ. We refer to this maximization problem as Max-CSP-$\mathcal{F}$, where $\mathcal{F}$ denotes the family of predicates constraints are taken from. For $0 < s < c \leq 1$, Gap-CSP-$\mathcal{F}(c, s)$ is the promise problem where YES-instances are formulas Φ such that $\mathrm{val}(\Phi) \geq c$, and NO-instances are formulas Φ such that $\mathrm{val}(\Phi) \leq s$. Here we assume the constraints of CSP instances are restricted to be in the family $\mathcal{F}$.

We study two models of percolation on instances of CSP, *clause percolation* and *variable percolation.* Given an instance Φ of CSP its clause percolation is a random formula Φ_p^c over the same set of variables, that is obtained from Φ by keeping each clause of Φ independently with probability p.

Theorem 1.9. *Let $\varepsilon, \delta \in (0, 1)$ be fixed constants. There is a polynomial time reduction such that given a simple unweighted instance Φ of Max-CSP-$\mathcal{F}$ outputs a simple unweighted instance Ψ of Max-CSP-$\mathcal{F}$ on n variables, such that $val(\Psi) = val(\Phi)$, and for any $p > \frac{1}{n^{k-1-\delta}}$ the following holds.*

1. *If $val(\Phi) = 1$, then $val(\Psi_p^c) = 1$ with probability 1.*

2. *If* $val(\Phi) < 1$*, then with high probability* $|val(\Psi_p^c) - val(\Phi)| < \varepsilon$*.*

This immediately implies the following corollary.

Corollary 1.10. *Let* $\mathcal{F}$ *be a collection of Boolean constraints of arity* k*, and suppose that for some* $0 < s < c \leq 1$ *the problem Gap-CSP-*$\mathcal{F}(c, s)$ *is* $\mathcal{NP}$*-hard. Then Gap-CSP-*$\mathcal{F}(c-\varepsilon, s+\varepsilon)$ *is* $\mathcal{NP}$*-hard under a* noise*-robust reduction, where* noise *is p-clause percolation with* $p > \frac{1}{n^{k-1-\delta}}$*, with* n *denoting the number of variables in a given formula, and* $\varepsilon, \delta > 0$ *arbitrary constants.*

One ingredient of the proof of Theorem 1.9 that may be of independent interest is establishing that k-CSP does not admit a non-trivial approximation on formulas with $n^{k-\eta}$ clauses, where $\eta > 0$ is an arbitrary small positive constant.

We also consider variable percolation. Given an instance Φ of CSP we consider a random formula Φ_p^v whose set of variables is a subset S of the variables of Φ, where each variable of Φ is in S independently with probability $p \in (0, 1)$ and the clauses of Φ_p^v are all clauses of Φ induced by S. In other words, a clause C of Φ survives if and only if all variables of C are in S. Using ideas similar to those used for clause percolation we show that p-vertex percolated instance are essentially as hard as in the worst case for $p > \frac{1}{n^{1-\delta}}$ for any $\delta \in (0, 1)$.

Other Problems

For the Minimum Vertex Cover problem, we prove that for any α and $\delta > 0$ an algorithm that gives α approximation for percolated instances implies also a $\alpha - \delta$ approximation algorithm for worst-case instances. Our results hold for both edge and vertex percolation, where the edges or the vertices of a given graph remain with probability $p > \frac{1}{n^{1-\varepsilon}}$ for some $\varepsilon \in (0, 1)$. In particular, assuming the Unique Games Conjecture, the result of [24] implies that $2 - \delta$ approximation for the Minimum Vertex Cover problem is $\mathcal{NP}$-hard under a noise-robust reduction for any constant $\delta > 0$.

For the Hamiltonicity problem, we prove hardness results for percolated instances of both directed and undirected graphs with respect to edge percolation. We show that the problem where one needs to determine whether a graph contains a Hamiltonian cycle is also hard for percolated graphs, where each edge of a given graph is kept in the graph with probability $p > \frac{1}{n^{1-\varepsilon}}$ for any $\varepsilon \in (0, 1)$.

We additionally study percolation on instances of the Subset-Sum problem, where each item of the set is deleted with probability $1 - p$. We show that the problem remains hard as long as $p = \Omega(\frac{1}{n^{1/2-\varepsilon}})$, where $\varepsilon \in (0, 1/2)$ is an arbitrary constant, and n is the number of items in the given instance.

1.4 Related Work

There is a wide body of work on random discrete structures that has produced a wide range of mathematical tools [5, 17, 19, 26]. Randomly subsampling subgraphs by including each edge independently in the sample with probability p has been studied extensively in works concerned with cuts and flows (e.g., [22]). More recently, sampling subgraphs has been used to find independent sets [15]. The effect of subsampling variables in mathematical relaxations of constraint satisfaction problems on the value of these relaxations was studied in [4].

Edge-percolated graphs have been also used to construct hard instances for specific algorithms. For example, Kučera [25] proved that the well-known greedy coloring algorithm performs poorly on the complete r-partite graph in which every edge is removed independently with probability 1/2 and $r = n^{\varepsilon}$ for $\varepsilon > 0$. Namely, for this graph G, even if vertices are considered in a random order by the greedy algorithm, with high probability $\Omega(\frac{n}{\log n})$ colors are used to color the percolated graph whereas $\chi(G) \leq n^{\varepsilon}$.

Misra [27] studied edge percolated instances of the Max-Cut problem. He proves that assuming $\mathcal{NP} \neq \mathcal{BPP}$ there is no polynomial time algorithm that computes the size of the maximum cut in $G_{p,e}$ for any $p > \frac{1+\varepsilon}{d-1}$ in graphs of maximal degree d. This result is tight in the sense that it is easy to show that once $p < \frac{1-\varepsilon}{d-1}$, with high probability $G_{p,e}$ breaks into connected components of logarithmic size, and hence can be solved optimally in polynomial time. The techniques used in [27] differ from ours and rely on a recent hardness result for counting independent sets in sparse graphs [32].

Bukh [7] has considered coloring edge-percolated graphs, and states the question of whether $\mathbf{E}[\chi(G_{\frac{1}{2},e})] = \Omega(\chi(G)/\log(\chi(G)))$ as an "interesting problem." Bukh observed that the chromatic number of $G_{\frac{1}{2},e}$ has the same distribution as the chromatic number of the complement of $G_{\frac{1}{2},e}$, and therefore $\mathbf{E}[\chi(G_{\frac{1}{2},e})] \geq \sqrt{\chi(G)}$. However, it is not clear how to leverage the lower bound on the expectation to obtain a lower bound on $\chi(G_{\frac{1}{2},e})$ *with high probability*, which is required for our noise-robust reductions. For instance, standard martingale methods do not seem to imply that $\chi(G_{\frac{1}{2},e}) \geq \Omega(\sqrt{\chi(G)})$ holds with high probability.

1.5 Preliminaries

An *independent set* in a graph $G = (V, E)$ is a set of vertices that spans no edges. The *independence number* $\alpha(G)$ denotes the maximum size of an independent set in G. A *legal coloring* of a graph G is an assignment of colors to vertices such that no two adjacent vertices share the same color. The *chromatic number* $\chi(G)$ denotes the minimum number of colors needed for a legal coloring of G. Note that in a legal coloring of G each color class forms an independent set, and hence $\chi(G) \cdot \alpha(G) \geq n$.

A *vertex cover* in a graph $G = (V, E)$ is a set of vertices $S \subseteq V$ such that every edge $e \in E$ is incident to at least one vertex in S. Note that a subset of the vertices $S \subseteq V$ is an independent set in G if and only if $V \setminus S$ is a vertex cover. In particular, G contains a vertex cover of size k if and only if it contains an independent set of size $n - k$.

We will always measure the running time of algorithms in terms of the size of the percolated instance. Since G and $G_{p,e}$ have the same number of vertices, this generally does not affect the size of the instance by more than a polynomial factor. On the other hand, $G_{p,v}$ may be much smaller than G for very small values of p. However, in this work we will be only dealing with the case where $p = \frac{1}{n^{1-\Omega(1)}}$, hence with high probability the size of the vertex percolated and original graphs are polynomially related as well.

We will use the following version of the Chernoff bound.

Lemma 1.11 (Chernoff bound, Theorem 7.3.2 in [20])**.** *Let $x_1, \ldots, x_n$ be independent Bernoulli trials with $\Pr[x_i = 1] = p$, and let $\mu = \mathbf{E}[\sum_{i=1}^{n} x_i] = pn$. Let $r \geq e^2$. Then*
$\Pr[\sum_{i=1}^{n} x_i > (1+r)\mu] < \exp(-(\mu r/2)\ln r)$.

2 Maximum Independent Set and Percolation

In this section we demonstrate the hardness of approximating $\alpha(G)$ and $\chi(G)$ in both edge percolated and vertex percolated graphs. We base our results on a theorem of Feige and Kilian, saying that for every fixed $\varepsilon > 0$ the problem Coloring-vs-MIS$(n^\varepsilon, n^\varepsilon)$ is $\mathcal{NP}$-hard.

Theorem 2.1 ([14])**.** *For every $\varepsilon > 0$ it is $\mathcal{NP}$-hard to decide whether a given n-vertex graph G satisfies $\chi(G) \le n^\varepsilon$ or $\alpha(G) \le n^\varepsilon$.*

Edge percolation Below we prove Theorem 1.4. We will use the following lemma, due to Turan (see, e.g. [2]).

Lemma 2.2. *Every graph H with l vertices and e edges contains an independent set of size at least $\frac{l^2}{2e+l}$.*

As a corollary we observe that if a graph contains no large independent sets, then it can also cannot contain large subsets of the vertices that span a small number of edges.

Corollary 2.3. *Let $G = (V, E)$ be an n-vertex graph satisfying $\alpha(G) \le k$. Then every set of vertices of size $l \ge k$ spans at least $l(l-k)/2k$ edges.*

Proof. Let H be a subgraph of G induced by l vertices, and suppose that H spans e edges. Then, by Lemma 2.2 we have $\alpha(H) \ge \frac{l^2}{2e+l}$. On the other hand, $\alpha(H) \le \alpha(G) \le k$, and hence $\frac{l^2}{2e+l} \le k$, as required. □

We are now ready to prove Theorem 1.4 saying that for any n-vertex graph $G = (V, E)$ it holds that with high probability $\alpha(G_{p,e}) \le O\left(\frac{\alpha(G)}{p}\log(np)\right)$.

Proof of Theorem 1.4. For a given graph G, let $k = \alpha(G) + 1$ and let $C > 0$ be a large enough constant. By Corollary 2.3, every subset of size $l = C\frac{\alpha(G)}{p}\log(np)$ spans at least $\frac{l(l-k)}{2k}$ edges in G. Hence, by taking union bound over all subsets of size l, the probability there exists a set of size l in $G_{p,e}$ that spans no edge is at most

$$\binom{n}{l} \cdot (1-p)^{\frac{l(l-k)}{2k}} < \left(\frac{en}{l}\right)^l \cdot \exp\left(-p \cdot \frac{l(l-k)}{2k}\right) < (np)^{-\Omega(l)},$$

where the last inequality uses the choices of l and k, implying that $\left(\frac{en}{l}\right)^l < (np)^l$ and $\exp(-p\frac{l(l-k)}{2k}) < \exp(-\Omega(l \cdot \log(np))) = (np)^{-\Omega(l)}$. Therefore, with high probability $\alpha(G_{p,e}) \le C\frac{\alpha(G)}{p}\log(np)$. □

Theorem 1.5 follows immediately from Theorem 1.4.

Proof of Theorem 1.5. Let G be an instance G of the Coloring-vs-MIS(q, a) problem Note that for the YES-case if $\chi(G) \le q$, then clearly $\chi(G_{p,e}) \le q$. For the NO-case by Theorem 1.4 if $\alpha(G) \le a$, then with high probability $\alpha(G_{p,e}) \le O\left(\frac{a}{p}\log(np)\right)$ which implies the strongly-noise-robust hardness.

The "in particular" part follows immediately from Theorem 2.1. □

Remark. *Note that for constant $p > 0$ (e.g., $p = 1/2$) this theorem establishes inapproximability for the independence number of $G_{p,e}$ that matches the inapproximability for the worst case.*

Remark. *Note also that for* $p > \frac{1}{n^{1-\varepsilon}}$ *(in fact, for* $p > \frac{\log(n)}{n}$*) such random percolated graphs have maximal degree at most* $O(pn)$ *with high probability. Therefore, such graphs* $G_{p,e}$ *can be colored efficiently using* $O(pn)$ *colors. In particular, with high probability* $G_{p,e}$ *contains an independent set of size* $\Omega(1/p)$ *and hence, the independence number can be approximated within a factor of* pn *on* p*-percolated instances.*

Vertex percolation Next we handle vertex percolation. We show that approximating $\alpha(G)$ and $\chi(G)$ on vertex percolated instances with $p > \frac{1}{n^{1-\delta}}$ is essentially as hard as on worst-case instances. Here n is the number of vertices in the graph and $\delta \in (0,1)$ does not depend on n. We show this again by relying on the hardness of the gap problem Coloring-vs-MIS for percolated instances.

Note that in the case of vertex percolation, the (in)approximability guarantee should depend on the number of vertices in the percolated graph $G_{p,v}$, and not on the number in the original graph.

Theorem 2.4. *The Coloring-vs-MIS*(q, a) *problem is strongly-*noise*-robust to itself, where* noise *is vertex percolation with* $p > 0$.

In particular, for any $\delta, \varepsilon > 0$ *unless* $\mathcal{NP} \subseteq \mathcal{BPP}$ *there is no polynomial time algorithm that approximates either* $\alpha(G_{p,v})$ *or* $\chi(G_{p,v})$ *within a* $(n')^{1-\varepsilon}$ *factor for any constant* $\varepsilon > 0$*, where* n' *denotes the number of vertices in* $G_{p,v}$*, and any* $p > \frac{1}{n^{1-\delta}}$.

Proof. The strong robustness of Coloring-vs-MIS(q, a) is clear, since for any graph G if G' is a vertex induced subgraph of G, then $\chi(G') \leq \chi(G)$, and $\alpha(G') \leq \alpha(G)$, which is, in particular, true for $G' \sim G_{p,v}$.

For the "in particular" part, let $p > \frac{1}{n^{1-\delta}}$ and let $c = \frac{\log(pn)}{\log(n)} \in (\delta, 1)$ so that $p = \frac{1}{n^{1-c}}$. Let $\eta = \varepsilon \cdot c$.

Let G be an n-vertex graph, let $G_{p,v}$ be it vertex-percolated subgraph, and let n' be the number of vertices in $G_{p,v}$. By the Chernoff bound in Lemma 1.11 with high probability we have $|n' - pn| < 0.1pn$, and so, we assume from now on that $n^{\eta} < 2(n')^{\varepsilon}$.

By Theorem 2.1 it is $\mathcal{NP}$-hard to decide whether a given n-vertex graph G satisfies $\chi(G) \leq n^{\eta}$ or $\alpha(G) \leq n^{\eta}$. By the choice of parameters, if $\chi(G) \leq n^{\eta}$ then $\chi(G_{p,v}) \leq n^{\eta} < 2(n')^{\varepsilon}$, and similarly, if $\alpha(G) \leq n^{\eta}$ then $\alpha(G_{p,v}) < n^{\eta} < 2(n')^{\varepsilon}$. This completes the proof of the theorem. □

3 Graph Coloring and Percolation

We present our results in terms of the *maximum coverage problem* (see, for example, [35]), which is a variant of the set cover problem, and show later how graph coloring is related to maximum coverage.

3.1 Maximum Coverage

In the maximum coverage problem we are given a family of sets $\mathcal{F} = \{S_1, \ldots, S_m\}$ with $S_i \subseteq [n]$ and a number c. The goal is to find c sets in $\mathcal{F}$ such the cardinality of the union of these c sets is as large as possible. We will make use of the representation of a set S in terms of its incidence vector $x(S) \in \{0,1\}^n$. In this way, we can reformulate the maximum coverage problem as follows. Given $A \subseteq \mathbb{F}_2^n$, find elements $y_1, \ldots, y_c \in A$ that maximize $\|\vee_{i=1}^{c} y_i\|_1$, the Hamming weight of the bitwise-OR of the vectors.

We will prove two existential results saying that if A is of constant density $\alpha > 0$, then there exists a good cover using only 2 or 3 vectors.

Lemma 3.1. *Let $A \subseteq \mathbb{F}_2^n$ with $|A| = \alpha 2^n$. Then there exist $y_1, y_2, y_3 \in A$ such that $\|y_1 \vee y_2 \vee y_3\|_1 \geq n - 4/\alpha^3$.*

Lemma 3.2. *Let $A \subseteq \mathbb{F}_2^n$ with $|A| = \alpha 2^n$. Then there exist $y_1, y_2 \in A$ such that $\|y_1 \vee y_2\|_1 \geq n - (1+r)\sqrt{n}$, where $r = \max\{e^2, 2\ln 1/\alpha\}$.*

3.2 Proof of Lemma 3.1 using additive combinatorics

Lemma 3.1 follows almost immediately from a result about sumsets. Recall that the Minkowski sum of two sets A, B is defined as $A + B = \{x + y : x \in A, y \in B\}$.

Lemma 3.3 (Corollary 3.5 in [31])**.** *Let $A \subseteq \mathbb{F}_2^n$ with $|A| = \alpha 2^n$. Then $A + A + A$ contains an affine subspace of dimension at least $n - 4/\alpha^3$.*

Because an affine subspace of dimension at least $n - 4/\alpha^3$ must contain an element of Hamming weight at least $n - 4/\alpha^3$, Lemma 3.1 follows from Lemma 3.3 and the observation that $\|\sum_{i=1}^c y_i\|_1 \leq \|\bigvee_{i=1}^c y_i\|_1$.

We note that using the recent result of Sanders on the quasi-polynomial Freiman-Ruzsa Theorem [29] we can also get better dependence on α by adding an additional copy of A.

Lemma 3.4 (Theorem A.1 in [29])**.** *Let $A \subseteq \mathbb{F}_2^n$ with $|A| = \alpha 2^n$. Then $A + A + A + A$ contains an affine subspace of dimension at least $n - O(\log^4 1/\alpha)$.*

3.3 Proof of Lemma 3.2 using Fourier analysis

We use an inequality from Fourier analysis to give a proof of Lemma 3.2 via the probabilistic method.

Definition 3.5. *Given $x \in \mathbb{F}_2^n$, define $y \sim N_\rho(x)$ by letting each y_i be equal to x_i with probability $\frac{1+\rho}{2}$, and be equal to $1 - x_i$ with probability $\frac{1-\rho}{2}$.*

Let $\mathrm{Uni}(S)$ denote the uniform distribution on a set S, and let U_n denote $\mathrm{Uni}(\mathbb{F}_2^n)$. The following lemma is a corollary of the reverse Bonami-Beckner inequality.

Lemma 3.6 (Corollary 3.5 in [28])**.** *Let $A, B \subseteq \mathbb{F}_2^n$ with $|A| = |B| = \alpha 2^n$. Then*

$$\Pr_{\substack{x \leftarrow Uni(A) \\ y \leftarrow N_\rho(x)}}[y \in B] \geq \alpha^{(1+\rho)/(1-\rho)}.$$

Proof of Lemma 3.2. Let $A \subseteq \mathbb{F}_2^n$ with $|A| = \alpha 2^n$, and let $B = A + \vec{1} = \{x + \vec{1} : x \in A\}$, where $\vec{1}$ is the n-dimensional all 1s vector. Note that to prove Lemma 3.2 it suffices to show that there exist $x \in A, y \in B$ such that $\|x+y\|_1 = (1+r)\cdot\sqrt{n}$, since then $y+\vec{1} \in A$ and $\|x+(y+\vec{1})\|_1 = n-(1+r)\cdot\sqrt{n}$.

Let $\varepsilon = 1/\sqrt{n}$ and let $\rho = 1 - 2\varepsilon$. By Lemma 3.6,

$$\Pr_{\substack{x \leftarrow U_n \\ y \leftarrow N_\rho(x)}}[x \in A, y \in B] = \Pr_{\substack{x \leftarrow \mathrm{Uni}(A) \\ y \leftarrow N_\rho(x)}}[y \in B] \cdot \Pr_{x \leftarrow U_n}[x \in A] \geq \alpha^{2/(1-\rho)} = \alpha^{\sqrt{n}}. \tag{1}$$

Set $r = \max\{e^2, 2\ln(1/\alpha)\}$. Note that by definition of $y \sim N_\rho(x)$ we have that $\Pr[x_i \neq y_i] = 1/\sqrt{n}$ for each i independently. Therefore, by the Chernoff bound in Lemma 1.11,

$$\Pr_{\substack{x \leftarrow U_n \\ y \leftarrow N_\rho(x)}}[\|x+y\|_1 \leq (1+r)\sqrt{n}] \geq 1 - e^{-(r/2\ln r)\sqrt{n}} \geq 1 - \alpha^{2\sqrt{n}}. \quad (2)$$

Since the sum of the probabilities in Equations (1) and (2) is strictly greater than 1, the corresponding events cannot be disjoint. Hence there exist $x \in A, y \in B$ such that $\|x+y\|_1 \leq (1+r)\sqrt{n}$. □

3.4 Coloring Using Subgraphs

We now show how to apply the results in the previous subsection to the graph coloring problem. Throughout this section we let $G = (V, E)$ with $n = |V|, m = |E|$. We will identify the elements of $[n]$ with vertices V in the vertex percolation case and the elements of $[m]$ with edges E in the edge percolation case. Let $G_{|U}$ denote the subgraph of G induced by $U \subseteq V$.

Lemma 3.7. *Let $G = (V, E)$ and let $V_1, V_2 \subseteq V$ with $V_1 \cup V_2 = V$. If $\chi(G_{|V_1}) \leq k_1$ and $\chi(G_{|V_2}) \leq k_2$ then $\chi(G) \leq k_1 + k_2$.*

Proof. Assume that $V_1 \cap V_2 = \emptyset$ (if not, replace V_1 with $V_1 \setminus V_2$ in the following argument). Color $G_{|V_1}$ with k_1 colors and color $G_{|V_2}$ with k_2 fresh colors. Because $G_{|V_1}$ and $G_{|V_2}$ are colored with separate colors any edges between V_1 and V_2 have endpoints with distinct colors. □

Lemma 3.8. *Let $G = (V, E)$, let $E_1, E_2 \subseteq E$ with $E_1 \cup E_2 = E$, and let $G_1 = (V, E_1), G_2 = (V, E_2)$. If $\chi(G_1) \leq k_1$ and $\chi(G_2) \leq k_2$ then $\chi(G) \leq k_1 k_2$.*

Proof. Let c_1 be a coloring of G_1 with k_1 colors, and let c_2 be a coloring of G_2 with k_2 colors. We claim that the coloring $c(v) = (c_1(v), c_2(v))$ is a legal coloring of G with $k_1 k_2$ colors. Consider an edge $e = (u, v) \in E$. If $e \in E_1$ then $c(u)$ differs from $c(v)$ in the first coordinate. Otherwise $e \in E_2$ in which case $c(u)$ differs from $c(v)$ in the second coordinate. □

3.5 Lower Bounding the Chromatic Number

We now prove lower bounds on the chromatic number of percolated graphs. We will consider both vertex and edge percolation with $p = \frac{1}{2}$. This choice of p is important because $G_{\frac{1}{2},v}$, $G_{\frac{1}{2},e}$ become the distributions of graphs induced by uniformly random subsets of V and E, respectively. However, it is easy to obtain bounds for $p < \frac{1}{2}$ by composing the bounds for $p = \frac{1}{2}$. When stating bounds based on Lemma 3.2 we set $r = \max\{e^2, 2\ln(1/\alpha)\}$.

The idea will be to argue that if many subgraphs of a graph G are k-colorable, then G is colorable with $f(k)$ colors for relatively small $f(k)$. To see how this idea works, consider the following easy case. Suppose that $\Pr[\chi(G_{\frac{1}{2},v}) \leq k] > \frac{1}{2}$. Then there exists $V' \subseteq V$ such that $G_{|V'}$ and $G_{|\overline{V'}}$ are both k-colorable. It follows that G is $2k$-colorable by Lemma 3.7. We now consider the case where the density of k colorable subgraphs α is less than $\frac{1}{2}$.

Vertex percolation We now prove Theorem 1.6, saying that if G is an n-vertex graph with $k = \chi(G)$, then for every $\lambda > 0$ it holds that $\Pr[\chi(G_{\frac{1}{2},v}) \geq k/2 - \lambda\sqrt{k}] > 1 - e^{-\lambda^2/2}$.

Proof of Theorem 1.6. Note first that if $k = \chi(G)$, then $\mathbf{E}[\chi(G_{\frac{1}{2},v})] \geq k/2$. Indeed, let $\overline{G_{\frac{1}{2},v}}$ be the subgraph of G induced by the vertices that are not in $G_{\frac{1}{2},v}$. Observe that $G_{\frac{1}{2},v}$ has the same distribution as $\overline{G_{\frac{1}{2},v}}$, and hence, by Lemma 3.7 we have

$$\mathbf{E}[\chi(G_{\frac{1}{2},v})] = \frac{1}{2}\mathbf{E}[\chi(G_{\frac{1}{2},v}) + \chi(\overline{G_{\frac{1}{2},v}})] \geq k/2. \tag{3}$$

Next we define a martingale $0 = X_0, X_1, \ldots, X_k$ as follows. Fix some k-coloring of G, and for each $i = 1, \ldots, k$ let X_i be the chromatic number of the percolated subgraph induced by the first i color classes. It is clear that $|X_{i-1} - X_i| \leq 1$ for all $i = 1, \ldots, k$, and $X_k = \chi(G_{\frac{1}{2},v})$. Therefore, by Azuma's inequality (see e.g. Theorem 7.2.1 in [2]) it follows that $\Pr[X_k \leq \mathbf{E}[X_k] - \lambda\sqrt{k}] < e^{-\lambda^2/2}$. This concludes the proof of the Theorem. $\square$

Edge percolation For a random $G_{\frac{1}{2},e}$ let $\overline{G_{\frac{1}{2},e}}$ be the graph obtained from G by removing all edges in $G_{\frac{1}{2},e}$. By observing that $G_{\frac{1}{2},e}$ and $\overline{G_{\frac{1}{2},e}}$ have the same distribution, and using Lemma 3.8 we get that

$$\mathbf{E}[\chi(G_{\frac{1}{2},e})] = \frac{1}{2} \cdot \mathbf{E}[\chi(G_{\frac{1}{2},e}) + \chi(\overline{G_{\frac{1}{2},e}})] \geq \mathbf{E}[\sqrt{\chi(G_{\frac{1}{2},e}) \cdot \chi(\overline{G_{\frac{1}{2},e}})}] \geq \sqrt{\chi(G)} = \sqrt{k},$$

analogous to the bound in (3) for vertex percolation. However, using the martingale as above Azuma's inequality implies $\Pr[\chi(G_{\frac{1}{2},e}) \leq (1-\lambda)\sqrt{k}] < e^{-\lambda^2/2}$, which is not enough to prove that $\chi(G_{\frac{1}{2},e})$ is large with high probability.

Below we use alternative techniques to prove Theorem 1.7 asserting that a weaker bound on $\chi(G_{\frac{1}{2},e})$ holds with probability $1 - \alpha$ for any $\alpha > 0$. To the best of our knowledge these techniques are new to this area, and may be of independent interest.

Lemma 3.9. $\Pr[\chi(G_{\frac{1}{2},e}) \leq k] \geq \alpha \Rightarrow \chi(G) \leq k^3 + 8/\alpha^3$.

Proof. Identify subsets of edges E with vectors in $\mathbb{F}_2^m$. Because $\Pr[\chi(G_{\frac{1}{2},e}) \leq k] \geq \alpha$ by Lemma 3.1 there exist $E_1, E_2, E_3 \subseteq E$ such that each $G_i = (V, E_i)$ is k-colorable and $|E_1 \cup E_2 \cup E_3| \geq m - 4/\alpha^3$. Using Lemma 3.8, we can then color $G(V, E_1 \cup E_2 \cup E_3)$ with k^3 colors. We then color the endpoints of the remaining $E \setminus (E_1 \cup E_2 \cup E_3)$ edges using $8/\alpha^3$ new colors to achieve a $(k^3 + 8/\alpha^3)$-coloring of G. $\square$

The next lemma gives an unconditional upper bound on the chromatic number of a graph.

Lemma 3.10. *Let $G = (V, E)$ be a graph with $|E| = m$. Then $\chi(G) \leq 3\sqrt{m} + 1$.*

Proof. Partition V into sets $V_0 = \{v \in V : \deg(v) < \sqrt{m}\}$ and $V_1 = \{v \in V : \deg(v) \geq \sqrt{m}\}$. By Brooks' Theorem [6], $\chi(G_{|V_0}) \leq \max_{v \in V_0} \deg(v) + 1 \leq \sqrt{m} + 1$. Furthermore, because $\sum_{v \in V_1} \deg(v) \leq 2m$, it follows that $|V_1| \leq 2\sqrt{m}$, and in particular $\chi(G_{|V_1}) \leq 2\sqrt{m}$. The result follows by Lemma 3.7. $\square$

We use a variant of the same partitioning trick in the following lemma.

Lemma 3.11. $\Pr[\chi(G_{\frac{1}{2},e}) \leq k] \geq \alpha \Rightarrow \chi(G) \leq (4 + 2r) \cdot k \cdot m^{1/4}$.

Proof. Note first that if $k \geq m^{1/4}$, then the claimed bound holds by Lemma 3.10. So we assume henceforth that $k < m^{1/4}$.

Identify subsets of edges E with vectors in $\mathbb{F}_2^m$. Because $\Pr[\chi(G_{\frac{1}{2},e}) \leq k] \geq \alpha$ by Lemma 3.2 there exist $E_1, E_2 \subseteq E$ such that $G_1 = (V, E_1), G_2 = (V, E_2)$ are k-colorable and $|E_1 \cup E_2| \geq m - (1+r)\sqrt{m}$.

Let $E_3 = E \setminus (E_1 \cup E_2)$, and define the graph $G_3 = (V, E_3)$. Define a partition $U, \overline{U}$ of V, where $U = \{v \in V : \deg_{G_3}(v) < m^{1/4}/k\}$ and $\overline{U} = \{v \in V : \deg_{G_3}(v) \geq m^{1/4}/k\}$. We claim (1) that $\chi(G_{|U}) \leq 2km^{1/4}$ and (2) that $\chi(G_{|\overline{U}}) \leq 2(1+r)km^{1/4}$. By Lemma 3.7 we then get the upper bound $\chi(G) \leq \chi(G_{|U}) + \chi(G_{|\overline{U}}) \leq (4+2r)km^{1/4}$.

To prove (1) note that by Brooks' Theorem [6] we have $\chi((G_3)_{|U}) \leq 2m^{1/4}/k$, and thus by Lemma 3.8, $\chi(G_{|U}) \leq \chi(G_1) \cdot \chi(G_2) \cdot \chi((G_3)_{|U}) \leq 2km^{1/4}$. For (2) note that $\sum_{v \in \overline{U}} \deg_{G_3}(v) \leq 2(1+r)\sqrt{m}$, and hence $\chi(G_{|\overline{U}}) \leq |\overline{U}| \leq 2(1+r)km^{1/4}$, as required ☐

Taking the contrapositive of Lemmas 3.9 and 3.11 implies Theorem 1.7.

Proof of Theorem 1.8

Finally, we use Theorem 1.7 to prove the strong robustness result for Gap-Coloring. Let G be an instance of the Gap-Coloring(q, Q) problem. We claim the following:

YES-case: If $\chi(G) \leq q$, then $\chi(G_{\frac{1}{2},e}) \leq q$.

NO-case: If $\chi(G) \geq Q$, then $\chi(G_{\frac{1}{2},e}) \geq \exp(\Omega(Q^{1/3}))$ with probability at least 0.99.

The YES-case is clear, since removing edges can only decrease the chromatic number. The NO-case follows from Theorem 1.7. Therefore, the Gap-Coloring(q, Q) problem is strongly-noise-robust to the Gap-Coloring$(q, \exp(\Omega(Q^{1/3})))$ problem. The "in particular" part of the theorem follows from the result of Huang [21] showing that Gap-Coloring$(q, \exp(\Omega(q^{1/3})))$ is $\mathcal{NP}$-hard.

4 Constraint Satisfaction Problems and Percolation

In this section we deal with percolation in Constraint Satisfaction Problems (CSP).

Clause percolation We show that for k-CSP the problem of approximating the optimal value on p-percolated instances is essentially as hard as approximating it on a worst-case instance as long as $p > \frac{1}{n^{k-1-\delta}}$ for any constant $\delta > 0$.

To prove Theorem 1.9 we start with the following lemma.

Lemma 4.1. *Let Φ be a simple unweighted k-CSP instance over an alphabet Σ with n variables and m clauses, and let $p > \frac{Cn}{\varepsilon^2 m}$ for some $\varepsilon \in (0,1)$ and some constant $C > 0$ that depends only on $|\Sigma|$. Then,*

1. *If $val(\Phi) = 1$, then $val(\Phi_p^c) = 1$.*
2. *If $val(\Phi) < 1$, then with high probability $|val(\Phi_p^c) - val(\Phi)| < \varepsilon$.*

Proof. The first item is clear, as any assignment that satisfies Φ will also satisfy Φ_p^c. For the second item, let m' be the number of clauses in Φ_p^c. By concentration bounds in Lemma 1.11 we have

$$\Pr[|m' - pm| > \varepsilon pm] < e^{-\Omega(\varepsilon^2 pm)} < e^{-\Omega(C\cdot n)},$$

where $\Omega(\cdot)$ hides some absolute constant. Fix an assignment σ to the variables of Φ, and let $s = \mathrm{val}_\sigma(\Phi)$. Then, the number of clauses in Φ satisfied by σ is sm. Let S_σ denote the number clauses in Φ_p^c satisfied by σ. Since we pick each clause with probability p independently, and recalling that $p > \frac{Cn}{\varepsilon^2 m}$ we have

$$\Pr[|S_\sigma - spm| > \varepsilon pm] < e^{-\Omega(\varepsilon^2 pm)} < e^{-\Omega(C\cdot n)}.$$

Denoting by E the event that $|S_\sigma - sm'| > \varepsilon m'$ we get

$$\begin{aligned}\Pr\left[|\mathrm{val}_\sigma(\Phi_p^c) - s| > \varepsilon\right] &= \Pr[|S_\sigma - sm'| > \varepsilon m'] \\ &\leq \Pr[E] + \Pr[|S_\sigma - sm'| > \varepsilon m' \;\big|\; \overline{E}] \\ &\leq 2e^{-\Omega(C\cdot n)}.\end{aligned}$$

Suppose now that $\mathrm{val}(\Phi) = s$. If σ is an optimal assignment to Φ, i.e., $\mathrm{val}_\sigma(\Phi) = s$, then we immediately have by the argument above that $\mathrm{val}_\sigma(\Phi_p^c) > s - \varepsilon$ with high probability. On the other hand, for any assignment σ' it holds that $\Pr[\mathrm{val}_{\sigma'}(\Phi_p^c) > s + \varepsilon] < e^{-\Omega(C\cdot n)}$ for some sufficiently large $C > 0$, and by taking union bound over all assignments σ we get

$$\Pr[\mathrm{val}(\Phi_p^c) > s + \varepsilon] < \Pr[\exists \sigma' \text{ such that } \mathrm{val}_{\sigma'}(\Phi_p^c) > s + \varepsilon] < 2e^{-\Omega(Cn)} \cdot |\Sigma|^n < c^n,$$

for some absolute constant $c < 1$. □

We note that we assume in the proof above that s is a constant independent of n. This assumption is justified as $s \geq \frac{1}{|\Sigma|^k}$, and we assume that $|\Sigma|$ and k are constants independent of n.

Next, we show a polynomial time reduction which, given a Max-CSP-$\mathcal{F}$ instance Φ, outputs a Max-CSP-$\mathcal{F}$ instance Ψ with n variables and $n^{k-\varepsilon}$ clauses such that $\mathrm{val}(\Psi) = \mathrm{val}(\Phi)$. We use similar ideas to those used in [9] who proved that unweighted instances of CSP problems are as hard to approximate as weighted ones.

Lemma 4.2. *For any $\delta \in (0,1)$ there is a polynomial time reduction which, given a simple unweighted Max-CSP-$\mathcal{F}$ instance Φ, outputs a simple Max-CSP-$\mathcal{F}$ instance Ψ with n variables and at least $n^{k-\delta}$ clauses such that $val(\Psi) = val(\Phi)$.*

Proof. The reduction works as follows. Let R be a parameter to be chosen later. Given an instance Φ of k-CSP with M clauses over the variables $x_1, \ldots, x_N$ the reduction creates the following instance Ψ. For each variable x_i of Φ, the instance Ψ will have a set of R corresponding variables $X_i = \{x_{i,j} : j \in [R]\}$, where we think of each variable in X_i as a copy of x_i. For each clause C of Φ we add to Ψ the R^k clauses obtained by taking the same constraint over each combination of the variables from the corresponding X_i's. We call this set of R^k clauses the *cloud* corresponding to C. So, Ψ has $n = NR$ variables and $m = M \cdot R^k$ clauses. Therefore, if $R > N^{k/\delta}$, then $m > n^{k-\delta}$.

Next we claim that $\mathrm{val}(\Phi) = \mathrm{val}(\Psi)$. Clearly, we have $\mathrm{val}(\Phi) \leq \mathrm{val}(\Psi)$, as any assignment $\sigma : \{x_1, \ldots, x_N\} \in \Sigma$ to Φ can be extended to the assignment τ to Ψ by letting $\tau(x_{i,j}) = \sigma(x_i)$ for all $i \in [N], j \in [R]$.

In the other direction, let τ be an assignment to the variables of Ψ.[5] For each $i \in [N]$ and a Σ let $p_i^a = \frac{|\{j \in [R] : \tau(x_{i,j}) = a\}|}{R}$ be the fraction of $x_{i,j}$'s that are assigned the value a. Construct an assignment σ to the variables of Φ randomly, by setting $\sigma(x_i) = a$ with probability p_i^a independently for each x_i. Equivalently we can choose one of the R copies of x_i in Ψ uniformly at random and assign to x_i the value assigned by τ to the variable chosen. Then for each clause C of Φ, the probability that σ satisfies C is equal to the fraction of the clauses in Ψ in the cloud corresponding to C that are satisfied by τ. Denote by $SAT_\sigma(C_i)$ the number of clauses that are satisfied by σ in the cloud corresponding to C_i. Since each clause of Φ corresponds to the same number of clauses in Ψ, it follows that the expected value of Φ under the assignment σ is

$$\begin{aligned}\mathbf{E}[\mathrm{val}_\sigma(\Phi)] &= \frac{1}{M}\sum_{i=1}^{M} \Pr[\sigma \text{ satisfies } C_i] \\ &= \frac{1}{M}\sum_{i=1}^{M} \frac{SAT_\sigma(C_i)}{R^k} \\ &= \mathrm{val}_\tau(\Psi).\end{aligned}$$

Hence, there exists an assignment σ to the variables of Φ such that $\mathrm{val}_\sigma(\Phi) \geq \mathrm{val}_\tau(\Psi)$, and thus $\mathrm{val}(\Phi) \geq \mathrm{val}(\Psi)$, as required. □

Theorem 1.9 follows immediately from Lemmas 4.1 and 4.2.

We observe that it is unlikely that Lemma 4.2 could also hold for Max-CSP-$\mathcal{F}$ instances with-arity k and $\Omega(n^k)$ constraints, as, for example, the value of a 3-SAT formula with $\Omega(n^3)$ clauses, can be $(1-\delta)$-approximated for every $\delta \in (0,1)$ in polynomial time [3].

Variable percolation

Next we show that Max-CSP-$\mathcal{F}$ is also hard under variable percolation. We prove below that for p that is no too small, with high probability Max-CSP-$\mathcal{F}$ is hard to approximate on percolated instances within the same factor as in the worst-case setting.

Theorem 4.3. *Let $\varepsilon, \delta > 0$ be fixed constants. There is a polynomial time reduction which, given a simple unweighted instance Φ, outputs a simple unweighted instance Ψ on n variables with the same constraints, such that $val(\Psi) = val(\Phi)$, and furthermore for any $p > \frac{1}{n^{1-\delta}}$ the following holds.*

1. *If $val(\Phi) = 1$, then $val(\Psi_p^v) = 1$ with probability 1.*
2. *If $val(\Phi) < 1$, then with high probability $|val(\Psi_p^v) - val(\Phi)| < \varepsilon$.*

The following corollary is the analogue of Corollary 1.10 for variable percolation.

Corollary 4.4. *Let $\mathcal{F}$ be a collection of Boolean constraints of arity k, and suppose that for some $0 < s < c \leq 1$ the problem Gap-CSP-$\mathcal{F}(c, s)$ is $\mathcal{NP}$-hard. Then Gap-CSP-$\mathcal{F}(c-\varepsilon, s+\varepsilon)$ is $\mathcal{NP}$-hard under a robust reduction with respect to vertex percolation with any parameter $p > \frac{1}{n^{1-\delta}}$, where n denotes the number of variables in a given formula, and $\varepsilon, \delta > 0$ are arbitrary constants.*

[5] Note that if for each $i \in [N]$ the assignment τ gave the same value to all variables in X_i, this would naturally induce a corresponding assignment to Φ. However, this need not be the case in general.

Proof of Theorem 4.3. The reduction is the same reduction as in the proof of Theorem 1.9. Namely, given a simple unweighted instance Φ with N variables and M clauses the reduction replaces each variable x_i of Φ, with a set of R corresponding variables $X_i = \{x_{i,j} : j \in [R]\}$, and replaces each clause of Φ with a cloud of R^k corresponding clauses, by taking all possible combinations of the variables from the corresponding X_i's. That is, the output of the reduction Ψ has $n = NR$ variables and $m = M \cdot R^k$ clauses. We choose $R = N^{1/c}$, where $c = \frac{\log pn}{\log n} \in (\delta, 1)$ so that $\sqrt{\frac{\log N}{pR}} < \frac{1}{N^{c/2}}$.

For each $i \in [N]$ let X'_i be variables from X_i that remain in Ψ_p^v after variable percolation. By the Chernoff bound in Lemma 1.11, it follows that for $p > \frac{1}{N^{1-\delta}}$ with high probability $||X'_i| - pR| < O(\sqrt{pR\log n})$ for all $i \in [N]$. We assume from now on that this is indeed the case. For a constraint C_i of Φ let $x_{i_1}, \ldots, x_{i_k}$ be the variables appearing in C_i. Then, the number of clauses in the cloud corresponding to C_i in Ψ_p^v is equal to $\prod_{j=1}^k |X'_{i_j}|$, and the total number of clauses in Ψ_p^v is $\sum_{i=1}^M \prod_{j=1}^k |X'_{i_j}|$.

By Lemma 4.2 we have $\mathrm{val}(\Psi) = \mathrm{val}(\Phi)$. In particular, if Φ is satisfiable, then so is Ψ, as any assignment that satisfies Ψ also satisfies any subformula of Ψ, which implies that Ψ_p^v is also satisfiable with probability 1.

Suppose now that $\mathrm{val}(\Phi) < 1$. We claim that with high probability $|\mathrm{val}(\Psi_p^v) - \mathrm{val}(\Phi)| < \varepsilon$.

To prove that $\mathrm{val}(\Psi_p^v) \geq \mathrm{val}(\Phi) - \varepsilon$, let σ be an optimal assignment to Φ. Extend σ to an assignment τ to Ψ_p^v by letting $\tau(x_{i,j}) = \sigma(x_i)$ for all $1 \leq i \leq R$. Note that for each constraint C_i of Φ if C_i is satisfied by σ, then in Ψ_p^v all clauses in the corresponding cloud are satisfied, and otherwise no clause in the corresponding cloud is satisfied. Denoting by $SAT_\tau(C_i)$ the number of clauses that are satisfied by τ in the cloud corresponding to C_i we have

$$\mathrm{val}_\tau(\Psi_p^v) = \frac{\sum_{i=1}^M SAT_\tau(C_i)}{\sum_{i=1}^M |X'_{i_1}| \cdots |X'_{i_k}|} \geq \frac{\mathrm{val}(\Phi) M \cdot (pR - \sqrt{pR\log N})^k}{M(pR + \sqrt{pR\log N})^k} \geq \mathrm{val}(\Phi) - O\Big(\sqrt{\frac{\log N}{pR}}\Big).$$

By the choice of R we get for large enough N

$$\mathrm{val}_\tau(\Psi_p^v) \geq \mathrm{val}(\Phi) - O(\frac{1}{N^{c/2}}) \geq \mathrm{val}(\Phi) - \varepsilon.$$

Next, we prove that $\mathrm{val}(\Phi) \geq \mathrm{val}(\Psi_p^v) - \varepsilon$. Given an assignment τ to the variables of Ψ_p^v we decode it into an assignment to Φ using the same decoding as in the proof of Lemma 4.2. Namely, we choose a random assignment σ to the variables of Φ by setting $\sigma(x_i) = a$ with probability p_i^a independently between i's, where $p_i^a = \frac{|\{x_{i,j} \in X'_i : \tau(x_{i,j}) = a\}|}{|X'_i|}$. Let C'_i be the set of clauses in C_i that belong to Ψ_p^v. Let $SAT_\tau(C'_i)$ be the number of clauses that are satisfied by τ in C'_i, it follows that the expected value of Φ under the assignment σ is

$$\mathbf{E}[\mathrm{val}_\sigma(\Phi)] = \frac{1}{M}\sum_{i=1}^M \Pr[\sigma \text{ satisfies } C'_i] = \frac{1}{M}\sum_{i=1}^M \frac{SAT_\tau(C'_i)}{|X'_{i_1}| \cdots |X'_{i_k}|}. \tag{4}$$

On the other hand we have

$$\mathrm{val}_\tau(\Psi_p^v) = \frac{\sum_{i=1}^M SAT_\tau(C'_i)}{\sum_{i=1}^M |X'_{i_1}| \cdots |X'_{i_k}|}. \tag{5}$$

Now, using the assumption that for all $i \in [n]$ it holds that $||X'_i| - pR| < \sqrt{pR\log n}$, we get that both (4) and (5) are between $\frac{\sum_{i=1}^M SAT_\tau(C'_i)}{M(pR+\sqrt{pR\log N})^k}$ and $\frac{\sum_{i=1}^M SAT_\tau(C'_i)}{M(pR-\sqrt{pR\log N})^k}$. A simple computation reveals that the difference between the two quantities is at most $O(\sqrt{\frac{\log N}{pR}})$, and hence

$$\mathbf{E}[\mathrm{val}_\sigma(\Phi)] \geq \mathrm{val}_\tau(\Psi_p^v) - O(\sqrt{\frac{\log N}{pR}}) \geq \mathrm{val}_\tau(\Psi_p^v) - O(\frac{1}{N^{c/2}}) \geq \mathrm{val}_\tau(\Psi_p^v) - \varepsilon.$$

This completes the proof of Theorem 4.3. □

5 More NP-hard Problems and Percolation

In this section we study more $\mathcal{NP}$-hard problems.

5.1 Vertex Cover and Percolation

In the Minimum Vertex Cover problem we are given a graph G and our goal is to find a vertex cover of G of minimum size. There is a simple 2-approximation algorithm for the Minimum Vertex Cover problem [35]. On the hardness side, the problem is $\mathcal{NP}$-hard to approximate within a factor of 1.3606 [11], and assuming the Unique Games Conjecture is known to be $\mathcal{NP}$-hard to approximate within a $(2-\varepsilon)$ factor for any constant $\varepsilon > 0$ [24].

We prove that the same hardness results are percolation robust.

Edge percolation

We have the following simple lemma regarding independent sets in edge percolated subgraph of $K_{R,R}$.

Lemma 5.1. *Consider the complete bipartite graph $G = K_{R,R}$ with bipartition A, B. Then, the probability that there is an independent set I in $G_{p,e}$ such that $|I \cap A| = |I \cap B| = C\log(R)/p$ is at most R^{-3}, where C is a large enough constant independent of n or p.*

Proof. For fixed sets $S_A \subseteq A$ and $S_B \subseteq B$ each of size $C\log(R)/p$ the probability that S_A and S_B span no edge is $(1-p)^{(C\log(R)/p)^2}$. Therefore, by union bound over all S_A and S_B the probability that there is is an independent set I in $G_{p,e}$ with $|I \cap A| = |I \cap B| = C\log(R)/p$ is at most

$$\binom{R}{C\log(R)/p}^2 (1-p)^{(C\log(R)/p)^2} \leq m^{2C\log(R)/p} e^{-p(C\log(R)/p)^2}$$

which is at most R^{-3} for large enough C. □

Consider the following Gap-Vertex-Cover(c, s) problem where the YES-instances are graphs that have a vertex cover of size cn, and NO-instances are all graphs whose minimum vertex cover is larger than sn, where n is the number of vertices in G. Note that, equivalently, the YES-instances are graphs that contain an independent set of size $\alpha(G) \geq (1-c)n$, the NO-instances are graphs whose maximal independent set is of size $\alpha(G) \leq (1-s)n$.

We remark that the result of Khot and Regev [24] proves that assuming the Unique Games Conjecture the problem Gap-Vertex-Cover$(\frac{1}{2}+\varepsilon, 1-\varepsilon)$ if $\mathcal{NP}$-hard for all constant $\varepsilon > 0$. We show use this to show hardness of approximation for this problem on edge-percolated instances.

Theorem 5.2. *Let $\varepsilon, \delta \in (0,1)$ be fixed constants. Assuming the Unique Games Conjecture, Gap-Vertex-Cover$(\frac{1}{2}+\varepsilon, 1-\varepsilon)$ is $\mathcal{NP}$-hard under a* noise*-robust reduction, where* noise *is edge percolation with parameter p for any $p > \frac{1}{n^{1-\delta}}$ and n denotes the number of vertices in the given graph.*

In particular, assuming the Unique Games Conjecture $(2-\varepsilon)$-approximation of the Vertex Cover problem is hard on edge percolated instances.

Proof. By [24] assuming the Unique Games Conjecture, for any $\varepsilon > 0$ the problem Gap-Vertex-Cover$(\frac{1}{2}+\varepsilon, 1-\varepsilon)$ is $\mathcal{NP}$-hard. Equivalently, given an N-vertex graph G is is $\mathcal{NP}$-hard to distinguish between the case that $\alpha(G) > (1/2-\varepsilon)N$ and the case that $\alpha(G) < \varepsilon N$. We show a reduction from this problem to itself (with slightly larger parameter ε) that is robust for edge percolation.

Consider the reduction that given a graph G outputs the R-blowup of G, which we denote by H, with R to be chosen later. That is the graph H is a graph on $n = NR$ vertices, and it is clear that $\alpha(H) = \alpha(G) \cdot R$. Therefore, this is indeed a reduction from the Gap-Vertex-Cover$(\frac{1}{2}+\varepsilon, 1-\varepsilon)$ to itself. We show below that in fact the reduction is robust for edge percolation. In order to do it we prove that with high probability

$$\alpha(G) \cdot R \leq \alpha(\widetilde{H}) \leq \alpha(G) \cdot R + (C \log(R)/p) \cdot N, \tag{6}$$

where $\widetilde{H} \sim H_{p,e}$ denotes the edge percolation of H with parameter p. Indeed, the left inequality is clear because $\alpha(\widetilde{H}) \geq \alpha(H) = \alpha(G) \cdot R$, since $\widetilde{H}$ is a subgraph of H.

For the right inequality, by Lemma 5.1 with probability at least $(1 - N^2/R^3)$ the following holds: for every edge (u,v) of G the corresponding clouds $\widetilde{u}$ and $\widetilde{v}$ in H are such that there is no independent set I in $\widetilde{H}$, such that $|I \cap \widetilde{u}| \geq C\log(R)/p$ and $|I \cap \widetilde{v}| \geq C\log(R)/p$. Therefore, if I is an independent set that intersects some clouds on more than $C\log(R)/p$, then the vertices corresponding to these clouds must form an independent set in G. Thus, with probability at least $(1 - N^2/R^3)$ we have $\alpha(\widetilde{H}) \leq \alpha(G) \cdot R + (C\log(R)/p) \cdot N$.

Next we choose the parameter R such that the reduction above is indeed a robust reduction for edge percolation with parameter p. For the parameter p let $c = \frac{\log(pn)}{\log(n)}$ so that $p = \frac{1}{n^{1-c}}$, and let $R = N^{2/c}$ (where N is the number of vertices in the original graph).

Now, if $\alpha(G) > (1/2-\varepsilon)N$, then by (6) we have $\alpha(\widetilde{H}) \geq \alpha(G) \cdot R > (1/2-\varepsilon)NR = (1/2-\varepsilon)n$, and hence $\widetilde{H}$ contains a vertex cover of size $(1/2+\varepsilon)n$ On the other hand, we claim that if $\alpha(G) < \varepsilon N$, then with high probability $\alpha(\widetilde{H}) < 2\varepsilon n$. Indeed, by the choice of R we have $p = \frac{1}{n^{1-c}} = \frac{1}{(NR)^{1-c}} > \frac{N}{\alpha(G)} \cdot \frac{C\log(R)}{R}$. Therefore, by the right inequality of (6) we have $\alpha(\widetilde{H}) \leq \alpha(G) \cdot R + (C\log(R)/p) \cdot N \leq 2\alpha(G) \cdot R < 2\varepsilon n$, and hence $\widetilde{H}$ does not have a vertex cover of size $(1-\varepsilon)n$. This completes the proof of Theorem 5.2. ☐

Vertex percolation We now proceed with vertex percolation. Note that when considering vertex percolation, the percolation parameter p depends on the number of vertices in the given (worst-case instance) graph, while the performance of the algorithm is measured with respect to the number of vertices in the percolated graph, which is close to pn with high probability.

We will need the following concentration bound, which is an immediate corollary of the Chernoff bound in Lemma 1.11.

Corollary 5.3. *Let $X_1^{(1)}, \dots, X_n^{(1)}, \dots, X_1^{(m)}, \dots, X_n^{(m)}$ be independent 0-1 random variables with $\Pr[X_i^{(j)} = 1] = p$. Then, for some absolute constant $C > 0$ it holds that*

$$\Pr[\exists j \in [m] : |\sum_{i=1}^{n} X_i^{(j)} - pn| \geq \sqrt{Cpn\log(m)}] \leq m^{-3}.$$

Proof. By the multiplicative Chernoff bound above for each $j \in [m]$ it holds that $\Pr[|\sum_{i=1}^{n} X_i^{(j)} - pn| \geq \sqrt{Cpn\log(m)}] \leq e^{-C\log(m)} < m^{-4}$, where $C > 0$ is some absolute constant. By taking union bound over we get $\Pr[\exists j \in [m] : |\sum_{i=1}^{n} X_i^{(j)} - pn| \geq \sqrt{Cpn\log(m)}] \leq m \cdot m^{-4} = m^{-3}$, as required. □

We can now deal with vertex percolation and vertex-cover.

Theorem 5.4. *Let $\varepsilon, \delta \in (0,1)$ be fixed constants. Assuming the Unique Games Conjecture, Gap-Vertex-Cover$(1-\varepsilon, 1/2+\varepsilon)$ is $\mathcal{NP}$-hard under a robust reduction with respect to vertex percolation with parameter p, for any $p > \frac{1}{n^{1-\delta}}$, where n is the number of vertices in the starting graph.*

In particular, assuming the Unique Games Conjecture $(2-\varepsilon)$-approximation of the Vertex Cover problem is hard on vertex percolated instances.

Proof. The reduction is the same as in the proof of Theorem 5.2. For the parameters p and ε let $c = \frac{\log(pn)}{\log(n)}$ so that $p = \frac{1}{n^{1-c}}$, and let $R = (\frac{N}{\varepsilon^2})^{1/c}$. Given a graph G the reduction produces the R-blowup of G, which we denote by H. Then H is a graph on $n = NR$ vertices.

Let $\widetilde{H} = H_{p,e}$ denote the vertex percolation of H with parameter p. By Corollary 5.3, with high probability the number of vertices in $\widetilde{H} \sim H_{p,e}$, which we denote by m is between $pNR - C\sqrt{pNR\log(NR)}$ and $pNR + C\sqrt{pNR\log(NR)}$, and the number of vertices in every cloud of $\widetilde{H}$ is between $pR - C\sqrt{pR\log N}$ and $pR + C\sqrt{pR\log N}$, for some absolute constant $C > 0$ independent of N or p.

Clearly any independent set I in $\widetilde{H}$ gives rise to an independent set in G by taking all vertices v of G such that I intersects the corresponding cloud $\widetilde{v}$. This implies that with high probability it holds (for N large enough) that

$$\alpha(G) \cdot (pR - C\sqrt{pR\log(N)}) \leq \alpha(\widetilde{H}) \leq \alpha(G) \cdot (pR + C\sqrt{pR\log(N)}).$$

By the choice of R we have $R > \frac{C^2 \lg(N)}{\varepsilon^2 p}$, and hence $|\alpha(\widetilde{H}) - \alpha(G)pR| \leq \varepsilon \cdot \alpha(G)pR$. Therefore, denoting by m the number of vertices in $\widetilde{H}$ if $\alpha(G) > (1/2-\varepsilon)N$, then $\alpha(\widetilde{H}) \geq (1/2-3\varepsilon)m$, and hence $\widetilde{H}$ contains a vertex cover of size $(1/2+3\varepsilon)m$. On the other hand, if $\alpha(G) < \varepsilon N$, then with high probability $\alpha(\widetilde{H}) < 3\varepsilon m$, and and hence $\widetilde{H}$ does not have a vertex cover of size $(1-3\varepsilon)m$. □

5.2 Hamiltonicity and Percolation

Recall that an Hamiltonian cycle in a graph is a cycle that visits every vertex exactly once. Deciding if a graph (whether directed or undirected) contains a Hamiltonian cycle is a classical $\mathcal{NP}$-hard problem, which we denote by HamCycle. A Hamiltonian path, is a simple path that traverses all vertices in the graph.

In this section we prove that unless $\mathcal{NP} = \text{co}\mathcal{RP}$, there is no polynomial time algorithm that given an n-vertex graph G decides with high probability whether $G_{p,e}$ contains a Hamiltonian cycle for any $p > \frac{1}{n^{1-\epsilon}}$ where $\epsilon \in (0,1)$.

A natural approach in proving that deciding the Hamiltonicity of percolated instances is hard, is to "blow up" edges. Namely to replace each edge (u,v) by a clique of size k and connect both endpoints of the edges to all vertices of the clique. The idea is that when k is large enough, there is a Hamiltonian path with high probability between all pairs of distinct vertices of the clique even after percolation. Hence with high probability, we can connect u and v after percolating the edges, by a path that traverses all the vertices of the percolated clique.

The problem with this idea, is that the resulting graph after this blowup operation may not be Hamiltonian (even if the starting graph is) as there is a new set of vertices for every edge in the original graph that needs to be traversed by an Hamiltonian cycle. For *directed* graphs, we overcome this problem by adding to each vertex v a large clique C, adding a directed edge (v,c) for every $c \in C$ and adding a directed edge (c,u) for every $c \in C$ and $u \in N(v)$ (where $N(v)$ is the set of all vertices having a directed edge from v). Combining this reduction with the standard $\mathcal{NP}$-hardness reduction from directed to undirected Hamiltonian cycle (see, e.g., [30]) yields a similar result for *undirected* graphs. We omit the details.

Theorem 5.5. *Let $\varepsilon \in (0,1)$ be a fixed constant. The HamCycle is $\mathcal{NP}$-hard under a noise-robust reduction, where noise is the edge percolation with probability $p > \frac{1}{n^{1-\varepsilon}}$.*

We will need the following claim.

Claim 5.6. *Let $H = (V,E)$ be the directed graph with $V = \{s,t\} \cup U$, where $U = \{u_1, \dots, u_R\}$, the vertex s is a source and t is a sink t. The edges of H are*

$$E = \{(s \to u_i) : i \in [R]\} \cup \{(u_i \to t) : i \in [R]\} \cup \{(u_i \to u_j) : i,j \in [R]\}.$$

Let $H' = (V,E')$ be an edge percolation of H, where we keep each directed edge with probability $p = \frac{3\log^5(R)}{R}$. Then, with probability $1 - \frac{1}{R^3}$ there is a Hamiltonian path in H from s to t.

Proof. Let $p_0 \in (0,1)$, and consider the random graph $H_{p_0,e}$ Note that with probability at least $1 - 2(1-p_0)^R - R(p_0(1-p_0)^{R-1})^2$ there are two distinct vertices $v_1, v_R \in U$ such that $(s \to v_1), (v_R \to t)$ are both edges of $H_{p_0,e}$. Conditioning on these specific $v_1, v_R \in U$, we show that with high probability there is a Hamiltonian path from v_1 to v_R in the subgraph of $H_{p_0,e}$ induced by U.

By a result of [16, Theorem 1.3] if D is a p_0-edge percolation of the complete directed graph with R vertices with $p_0 = \frac{p}{3\log(R)} = \frac{\log^4(R)}{R}$, then with high probability every edge of D is contained in some Hamiltonian cycle in D. Note that the probability that $H_{p_0,e}$ contains a Hamiltonian path from v_1 to v_R is equal to the probability that $H_{p_0,e}$ contains a Hamiltonian cycle that goes through the edge $(v_1 \to v_R)$, conditioned on the event that $(v_1 \to v_R)$ is an edge in $H_{p_0,e}$. Therefore, since the distribution of the subgraph of $H_{p_0,e}$ induced by U is distributed like D, it follows that with high probability the subgraph $H_{p_0,e}$ induced by U contains a Hamiltonian path from v_1 to v_R, and hence $H_{p_0,e}$ contains a Hamiltonian path from s to t with probability at least $\frac{1}{2}$.

Next, let $\ell = 3\log(R)$ so that $p = \ell \cdot p_0$. We claim that the graph $H_{p,e}$ contains an Hamiltonian path from s to t with probability at least $1 - \frac{1}{R^3}$. Observe that if $H'_1, \dots, H'_\ell$ are independent copies of $H_{p_0,e}$, then the probability that none of the H'_i contains a Hamiltonian path from s to t is at most $(1/2)^\ell < \frac{1}{R^3}$. Therefore, since $1 - (1-p_0)^\ell \le p$ it follows that H dominates $\cup_{i=1}^{\ell} H_i$, and hence $H_{p,e}$ contains an Hamiltonian path from s to t with probability at least $1 - \frac{1}{R^3}$, as required. ☐

Proof of Theorem 5.5. In order to prove the theorem, we show a reduction that given a directed graph $G = (V, E)$ produces a directed graph $G' = (V', E')$ such that

- If G contains a Hamiltonian cycle, then G' contains a Hamiltonian cycle, and with high probability $G'_{p,e}$ contains a Hamiltonian cycle.
- If G does not contain a Hamiltonian cycle, then neither G' nor $G'_{p,e}$ contains a Hamiltonian cycle.

The reduction works as follows. Let $V = [N]$ be the vertices of G, and let R be a parameter to be chosen later. The vertices of G' will be $V' = V \bigcup (\cup_{i=1}^N U_i)$, where $U_i = \{u_1^i, \dots, u_R^i\}$. For each $i \in [N]$ the graph G' contains all edges in both directions inside U_i. For each directed edge $(i \to j) \in E$ we add in G' the directed edges

$$\{(i \to u_\ell^i) : \ell \in [R]\} \cup \{(u_\ell^i \to j) : \ell \in [R]\}.$$

That is, we turn the graph G into G' by adding a clique U_i for each vertex $v_i \in V$, and letting all edges outgoing from v_i go through this clique. This completes the description of the reduction.

Let us first show that that G contains a Hamiltonian cycle if and only if G' contains a Hamiltonian cycle. Indeed, suppose that $C = (\sigma_1, \dots, \sigma_N)$ is a Hamiltonian cycle in G. Then $C' = (\sigma_1, u_1^{\sigma_1} \dots, u_R^{\sigma_1}, \dots, \sigma_N, u_1^{\sigma_N} \dots, u_R^{\sigma_N})$ is a Hamiltonian cycle in G'. In the other direction, suppose that G' contains a Hamiltonian cycle C'. It is easy to see that any $i \in V$ appearing in C' must be followed immediately by a permutation of all R vertices in U_i. Therefore, by restricting C' to the vertices in V we get a Hamiltonian cycle in G.

Next we show that the reduction above is robust to edge percolation. Let $\tilde{G}' = G'_{p,e}$ be the edge percolation of G'. Clearly if G' does not contain a Hamiltonian cycle, then neither does $\tilde{G}'$. Therefore, it is only left to show that if G' contain a Hamiltonian cycle C, then with high probability $\tilde{G}'$ also contains a Hamiltonian cycle. As explained above a Hamiltonian cycle in G' is given by a permutation $\sigma = (\sigma_1, \dots \sigma_N) \in S_N$ and some ordering of the vertices in each U_i, i.e., $C' = (\sigma_1, u_1^{\sigma_1} \dots, u_R^{\sigma_1}, \dots, \sigma_N, u_1^{\sigma_N} \dots, u_R^{\sigma_N})$. Note that for each $i \in [N]$ the vertices $\{\sigma_i, u_1^{\sigma_i} \dots, u_R^{\sigma_i}, \sigma_{i+1}\}$ induce a subgraph isomorphic to the graph H from Claim 5.6. Therefore, by Claim 5.6 if $p > \frac{\log^4(R)}{R}$, then for each $i \in [N]$ with probability $1 - \frac{1}{R^3}$ there is path from σ_i to σ_{i+1} that visits all vertices in U_{σ_i}. By taking union bound over all $i \in [N]$ we get that with probability $1 - \frac{N}{R^3}$ such paths exist for all $i \in [N]$, and by concatenating them we conclude that $\tilde{G}'$ contains a Hamiltonian cycle with high probability.

Finally, we specify the choice of the parameter R. The obtained graph H has $n = NR$ vertices, and the constraints we have are $p > \frac{\log^4 R}{R}$ and $R^3 \gg N$. Therefore, in order to prove the theorem for $p > \frac{1}{n^{1-\varepsilon}}$ with $\varepsilon \in (0, 1)$ it is enough to take $R = N^{1/c}$, where $c = \frac{\log(pn)}{\log(n)} > \varepsilon$ such that $p = \frac{1}{n^{1-c}}$. □

5.3 Subset-Sum and Percolation

In this section we consider the Subset-Sum problem. In the Subset-Sum problem we are given a set items $\{a_i\}_{i=1}^n$ which are positive integers, and a target integer S. The goal is to decide whether there is a subset of a_i's whose sum is S.

Given an instance $I = (\{a_i\}_{i=1}^n; S)$ of the Subset-Sum problem, we define p-percolation on I with probability p to be a random instance I_p, where each item a_i is included in I_p with probability p independently, with the target of I_p being the same as the target of I.

It is known the Subset-Sum problem is $\mathcal{NP}$-hard. Below we prove hardness of the Subset-Sum problem with respect to the above percolation.

Theorem 5.7. *The Subset-Sum problem is $\mathcal{NP}$-hard under a* noise*-robust reduction, where* noise *is p-percolation with $p > \frac{1}{n^{1/2-\varepsilon}}$, where n is the number of items in a given instance, and $\varepsilon > 0$ is any fixed constant.*

Proof. In order to prove the theorem, we show a reduction that given an instance $I = (\{a_i\}_{i=1}^N; S)$ of the Subset-Sum problem with all $a_i > 0$, produces an instance I' on n variables such that the following two properties are satisfied.

- If $I \in$ Subset-Sum, then $I' \in$ Subset-Sum, and with high probability $I'_p \in$ Subset-Sum.
- If $I \notin$ Subset-Sum, then $I' \notin$ Subset-Sum, and hence $I'_p \notin$ Subset-Sum with probability 1.

Let us assume that the number of items in I is even. (If N is odd, then, add an item to I that is equal to zero). Let R be a parameter to be chosen later, let $M = 2RN$, $L_0 = 6RM \cdot \left(\sum_{i=1}^N a_i\right) + M$ and for $i = 1, \dots, N$ let $L_i = (6R)^i \cdot L_0$. For each $i \in [N]$ define the following set

$$J_i = \{L_i + a_i \cdot M + k : k \in \{-R, \dots, R\}\} \quad \text{and} \quad J'_i = \{L_i + k : k \in \{-R, \dots, R\}\}$$

Consider now the instance

$$I' = (\cup_{i \in [N]}(J_i \cup J'_i); S'),$$

where $S' = S \cdot M + \sum_{i=1}^N L_i$. Clearly, if R is not too large, this is a polynomial time reduction that outputs a Subset-Sum instance with $n = 2N(2R+1)$ items.

We show first that $I \in$ Subset-Sum if and only if $I' \in$ Subset-Sum. Indeed, suppose that for some subset $T \subseteq [N]$ it holds that $\sum_{i \in T} a_i = S$. Consider the following subset of items of I'. For each $i \in T$ take the item from J_i that corresponds to $k = 0$, and for $i \in [N] \setminus T$ take the item from J'_i that corresponds to $k = 0$. Summing these items we get

$$\sum_{i \in T}(L_i + a_i \cdot M + 0) + \sum_{i \in [N] \setminus T}(L_i + 0) = S'.$$

In the other direction, suppose that $I' \in$ Subset-Sum. Then, there is some subset $T' \subseteq [N] \times \{0,1\} \times \{-R, \dots, R\}$ such that

$$\sum_{(i,t,k) \in T'}(L_i + a_i \cdot M \cdot t + k) = S' = \sum_{i \in [N]} L_i + S \cdot M. \tag{7}$$

Claim 5.8. *For each $i \in [N]$ there is a unique $t_i \in \{0,1\}$ and a unique $k_i \in \{-R, \dots, R\}$ such that $(i, t_i, k_i) \in T'$. Furthermore, $\sum_{i=1}^N k_i = 0$ and $\sum_{i=1}^N a_i \cdot t_i = S$.*

Proof. Note first that for each $i \in [N]$ there are at most $2 \cdot (2R+1) < 6R$ terms (i,t,k) in T'. Therefore, $\sum_{(i,t,k) \in T'}(a_i \cdot M \cdot t + k) < 6R \cdot \sum_{i=1}^N (a_i M + R) < L_0$. Considering the sum in the LHS of (7) modulo L_0 (and recalling that $L_i = L_0 \cdot (6R)^i$) we conclude that

$$\sum_{(i,t,k) \in T'} L_i = \sum_{i=1}^N L_i \tag{8}$$

and

$$\sum_{(i,t,k)\in T'} (a_i \cdot M \cdot t + k) = S \cdot M. \tag{9}$$

By the choice of L_i, and using again the fact that for each $i \in [N]$ there are at most $2\cdot(2R+1) < 6R$ terms (i,t,k) in T', it is now easy to see that each L_i term in the LHS of (8) must appear exactly once, i.e., for each $i \in [N]$ there is a unique $t_i \in \{0,1\}$ and a unique $k_i \in \{-R,\ldots,R\}$ such that $(i,t_i,k_i) \in T'$.

To see that $\sum_{i=1}^N k_i = 0$, note that $\sum_{(i,t_i,k_i)\in T'} k_i < RN < M$. Now, since in (7) we have $L_i \equiv 0 \pmod M$ and $S' \equiv 0 \pmod M$ it follows that $\sum_{i=1}^N k_i = 0$. This concludes the proof. □

Therefore, by defining $T = \{i \in [N] : t_i = 1\}$ we get that $\sum_{i\in T} a_i = S$, and so $I \in$ Subset-Sum.

Next, we claim that the reduction above is in fact robust to noise. Indeed, consider the percolated instance I'_p for some $p \in (0,1)$. Note that if $I \notin$ Subset-Sum, then $I' \notin$ Subset-Sum, and hence $I'_p \notin$ Subset-Sum with probability 1. Therefore, it remains to show that if $I \in$ Subset-Sum, then with high probability $I'_p \in$ Subset-Sum. The proof relies on the following claim.

Claim 5.9. *Let $N \in \mathbb{N}$ be even, and let $R \in \mathbb{N}$. Let $A_1,\ldots,A_n \subseteq \{-R,\ldots,R\}$ be random sets chosen by letting each $k \in \{-R,\ldots,R\}$ be in A_i with probability p independently. Then, with probability $\geq 1 - N/2 \cdot (1-p^2)^{2R}$ for each $i \in [n]$ there is $k_i \in A_i$ such that $\sum_{i=1}^N k_i = 0$.*

Proof. Note that for each odd $i \in [N]$, the probability for a fixed element $k \in \{-R,\ldots,R\}$ that both $k \in A_i$ and $-k \in A_{i+1}$ hold is p^2. Therefore,

$$\Pr[\nexists k \in \{-R,\ldots,R\} : k \in A_i \text{ and } -k \in A_{i+1}] = (1-p^2)^{2R+1}.$$

Hence, by taking the union bound over all pairs $(i,i+1)$ with odd values of i we get that with probability at least $1 - N/2\cdot(1-p^2)^{2R+1}$, for all odd i's there is $k_i \in A_i$ such that $-k_i \in A_{i+1}$. □

Suppose now that $I \in$ Subset-Sum, i.e., for some subset $T \subseteq [N]$ it holds that $\sum_{i\in T} a_i = S$. Note that the percolated instance I'_p is obtained from I' by taking random subsets of J_i and J'_i independently of each other. For $i \in [N]$ define A_i to be the p-percolated subsets of J_i if $i \in T$, and define A_i to be the p-percolated subsets of J'_i if $i \notin T$. Note that if $R > \frac{C\log(N)}{p^2}$, then the conclusion of Claim 5.9 holds with probability at least $1 - 1/N$. hence, in the percolated instance I'_p by taking the items from A_i's that correspond to $k_i \in A_i$'s from Claim 5.9 we get

$$\begin{aligned}\sum_{i\in T}(L_i + a_i\cdot M + k_i) + \sum_{i\in[N]\setminus T}(L_i + k_i) &= (\sum_{i\in[N]} L_i) + (\sum_{i\in T} a_i \cdot M) + (\sum_{i\in[N]} k_i) \\ &= (\sum_{i\in[N]} L_i) + S\cdot L + 0 \\ &= S'.\end{aligned}$$

Therefore, with high probability $I'_p \in$ Subset-Sum as required.

Finally, note that the reduction works as long as $R > C\frac{\log(N)}{p^2}$, or equivalently $p > \Omega\left(\sqrt{\frac{\log N}{R}}\right)$. It is easy to verify that if we set $R = N^{1/c}$, with $c = \frac{\log(pn)}{\log(n)} - \frac{1}{2}$ such that $p = \frac{1}{n^{1/2-c}}$, then the foregoing reduction is indeed a robust reduction with respect to percolation with parameter $p > \frac{1}{n^{1/2-\varepsilon}}$ for any constant $\varepsilon > 0$, where n is the number of items in I'. □

6 Conclusion

We have examined the complexity of percolated instances of several well studied $\mathcal{NP}$-hard problems and established the hardness of solving these problems on such instances. It might be of interest to study the hardness of percolated instances of other $\mathcal{NP}$-hard problems, and of other classes of problems such as counting, $W[1]$-hard problems, and parallel computation.

It might also prove worthwhile to determine whether percolated instances of 3-SAT remain hard to solve for $p = O(1/n^2)$ over n-variable formulas.

It would be interesting to determine whether the HamCycle problem is $\mathcal{NP}$-hard under a robust reduction with respect to *vertex* percolation. We only show that this is true with respect to *edge* percolation, and it could be the case that the problem is not $\mathcal{NP}$-hard under a vertex percolation robust reduction. Proving such a result (if true) could be very interesting.

Acknowledgements

We thank Itai Benjamini, Uri Feige, and Sam Hopkins for useful discussions. We are grateful to the anonymous referees for helpful comments and turning our attention to [7].

References

[1] Réka Albert, Hawoong Jeong, and Albert-László Barabási. Error and attack tolerance of complex networks. *nature*, 406(6794):378–382, 2000.

[2] N. Alon and J. H. Spencer. *The Probabilistic Method*. Wiley-Interscience series in discrete mathematics and optimization. J. Wiley & Sons, New York, 2000.

[3] S. Arora, D. R. Karger, and M. Karpinski. Polynomial time approximation schemes for dense instances of np-hard problems. *J. Comput. Syst. Sci.*, 58(1):193–210, 1999.

[4] B. Barak, M. Hardt, T. Holenstein, and Steurer D. Subsampling mathematical relaxations and average-case complexity. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, San Francisco, California, USA*, pages 512–531, 2011.

[5] B. Bollobás. *Random graphs*. Springer, 1998.

[6] R. L. Brooks. On colouring the nodes of a network. *Mathematical Proceedings of the Cambridge Philosophical Society*, 37:194–197, 4 1941.

[7] B. Bukh. Interesting problems that I cannot solve. Problem 2. http://www.borisbukh.org/problems.html.

[8] Reuven Cohen, Keren Erez, Daniel Ben-Avraham, and Shlomo Havlin. Resilience of the internet to random breakdowns. *Physical review letters*, 85(21):4626, 2000.

[9] P. Crescenzi, R. Silvestri, and L. Trevisan. On weighted vs unweighted versions of combinatorial optimization problems. *Inf. Comput.*, 167(1):10–26, 2001.

[10] I. Dinur, E. Mossel, and O. Regev. Conditional hardness for approximate coloring. *SIAM J. Comput.*, 39(3):843–873, 2009.

[11] I. Dinur and S. Safra. On the hardness of approximating minimum vertex cover. *Annals of Mathematics*, 162(1):439–485, 2005.

[12] I. Dinur and I. Shinkar. On the conditional hardness of coloring a 4-colorable graph with super-constant number of colors. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 13th International Workshop, APPROX 2010, and 14th International Workshop, RANDOM 2010, Barcelona, Spain, September 1-3, 2010. Proceedings*, pages 138–151, 2010.

[13] M. Etscheid and H. Röglin. Smoothed analysis of local search for the maximum-cut problem. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, Portland, Oregon, USA*, pages 882–889, 2014.

[14] U. Feige and J. Kilian. Zero knowledge and the chromatic number. *Journal of Computer and System Sciences*, 57(2):187 – 199, 1998.

[15] U. Feige and Daniel Reichman. Recoverable values for independent sets. *Random Struct. Algorithms*, 46(1):142–159, 2015.

[16] A. Ferber, G. Kronenberg, and E. Long. Packing, counting and covering hamilton cycles in random directed graphs. 2015. arXiv:1506.00618.

[17] A. M. Frieze and C. McDiarmid. Algorithmic theory of random graphs. *Random Struct. Algorithms*, 10(1-2):5–42, 1997.

[18] M. R. Garey and D. S. Johnson. The complexity of near-optimal graph coloring. *J. ACM*, 23(1):43–49, January 1976.

[19] G. Grimmett. *Percolation*. Springer, 1999.

[20] S. Har-Peled. Concentration of Random Variables – Chernoff's Inequality. Available at http://sarielhp.org/teach/13/b_574_rand_alg/lec/07_chernoff.pdf.

[21] S. Huang. Improved hardness of approximating chromatic number. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 16th International Workshop, APPROX 2013, and 17th International Workshop, RANDOM 2013, Berkeley, CA, USA, August 21-23, 2013. Proceedings*, pages 233–243, 2013.

[22] D. R. Karger. Random sampling in cut, flow, and network design problems. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing, Montréal, Québec, Canada*, pages 648–657, 1994.

[23] S. Khot. Improved inapproximability results for maxclique, chromatic number and approximate graph coloring. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*, pages 600–609, 2001.

[24] S. Khot and O. Regev. Vertex cover might be hard to approximate to within 2-epsilon. *J. Comput. Syst. Sci.*, 74(3):335–349, 2008.

[25] L. Kucera. The greedy coloring is a bad probabilistic algorithm. *J. Algorithms*, 12(4):674–684, 1991.

[26] M. Mezard and A. Montanari. *Information, physics, and computation.* Oxford University Press, 2009.

[27] S. Misra. *Decay of Correlation and Inference in Graphical Models.* PhD thesis, MIT, 2014.

[28] E. Mossel, R. O'Donnell, O. Regev, J. E. Steif, and B. Sudakov. Non-interactive correlation distillation, inhomogeneous Markov chains, and the reverse Bonami-Beckner inequality. *Israel Journal of Mathematics*, 154:299–336, 2006.

[29] T. Sanders. On the Bogolyubov–Ruzsa lemma. *Analysis & PDE*, 5(3):627–655, 2012.

[30] M. Sipser. *Introduction to the Theory of Computation.* International Thomson Publishing, 1st edition, 1996.

[31] O. Sisask. Discrete fourier analysis. Available at https://people.kth.se/~sisask/Shillong/DFA_2.pdf.

[32] A. Sly. Computational transition at the uniqueness threshold. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, Las Vegas, Nevada, USA*, pages 287–296, 2010.

[33] D. A. Spielman and S-H. Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *J. ACM*, 51(3):385–463, 2004.

[34] D. A. Spielman and S-H. Teng. Smoothed analysis: an attempt to explain the behavior of algorithms in practice. *Commun. ACM*, 52(10):76–84, 2009.

[35] V. V. Vazirani. *Approximation Algorithms.* Springer-Verlag New York, Inc., New York, NY, USA, 2001.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de