Hardness of Approximate Coloring

Girish Varma[*]

Advisor : Prof. Prahladh Harsha

Tata Institute of Fundamental Research, Mumbai

Given a graph $G = (V, E)$, a C-coloring of G is an assignment of *colors* denoted by $\{1, \cdots, C\}$ to the vertices such that the end points of every edge have different colors. For a given graph, a C-coloring might not exist for all values of C. However a coloring, which gives different colors to all vertices in V, is a $|V|$-coloring. Furthermore, if the maximum degree of G is d it is not difficult to see, that G has a $(d + 1)$-coloring and it can also be found using a linear time algorithm.

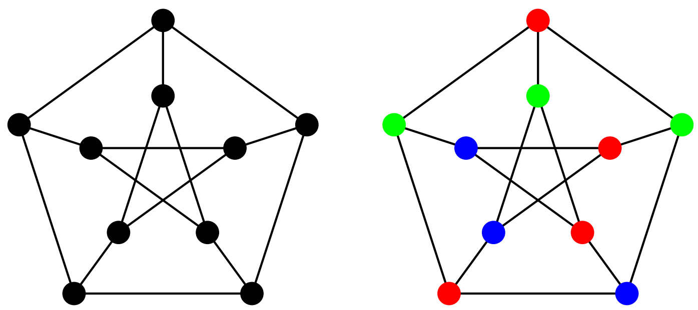

Figure 1: The figure on the right is a 3-coloring of the graph on the left. However it does not have a 2-coloring.

The goal of the graph coloring problem is, given a graph, find a C-coloring, for the minimum value of C for which there exists one. This minimum value is commonly known as the *chromatic number* of the graph.

The problem of computing the chromatic number is very well studied. However, for

[*] girishrv@gmail.com. Supported by Google India under the Google India PhD Fellowship Award.

a general graph it is a *hard* problem. That is, assuming the famous P$\neq$ NP conjecture, this problem cannot be solved in polynomial time. Furthermore, computing this number approximately within a multiplicative factor of $n^{1-\epsilon}$ (where n is the number of vertices in the graph) for any small constant ϵ is known to be a hard, assuming a different complexity conjecture [FK00]. Therefore, there is not much hope of having an efficient algorithm, which does much better than the trivial n-coloring that was mentioned earlier.

Since the general problem is well understood, the focus shifted on solving it for subclasses of graphs. A natural subclass of graphs to consider is the ones for which the chromatic number is a small constant c. For $c = 2$, such graphs are commonly called *bipartite*. There is a simple linear time algorithm for finding a 2-coloring in such graphs. It just assigns a vertex one color, the other color to all its neighbors, and continues until all vertices are colored. However for 3-colorable graphs, finding a 3-coloring is *NP-hard* (it cannot be solved by polynomial time algorithms, assuming P$\neq$ NP).

Hence a long series of works was aimed at solving this problem *approximately*. The problem of *approximate graph coloring* is, given a c-colorable graph, find a C-coloring for some $C > c$. For 3-colorable graphs, Wigderson gave the first not trivial improvement of finding a $O(\sqrt{n})$-coloring [Wig83] using combinatorial techniques. This was further improved to $O(n^{3/8})$-coloring [Blu94] by Blum. A major breakthrough for coloring algorithms was made by Karger, Motwani & Sudan, using semidefinite programming (SDP). For a 3-colorable graph with maximum degree d, they gave an $O(d^{1/3})$-coloring [KMS98] algorithm. Combining this with Wigderson's algorithm, they obtained a $O(n^{1/4})$-coloring algorithm. Blum & Karger combined the combinatorial methods of Blum [Blu94] with the SDP to get an $O(n^{3/14})$-coloring. The current best known (see [KT14]) efficient algorithms output a $n^{0.19996}$-coloring. Assuming P$\neq$ NP, it is known that efficient algorithms cannot find a 4-coloring (see [GK04]). Hence there is a large gap ($n^{0.19996}$ vs 4) between what current algorithms can achieve and the *hardness results* known.

The coloring problem has a natural generalization to hypergraphs. A k-*uniform hy-*

pergraph $G = (V, E)$ is similar to a graph, with the edges $E \subseteq V^k$ containing k vertices. A c-coloring of a hypergraph is coloring of vertices using colors $\{1, \cdots, c\}$ such that no edge is *monochromatic* (all vertices in the edge have the same color). For $k = 2$, a hypergraph is simply a graph. The *approximate hypergraph coloring* problem is defined similar to the graph coloring case, having parameters k, c, C, as given a c-colorable k-uniform hypergraph, find a C-coloring. It turns out that for $k > 2$, even the problem of finding a 2-coloring for a 2-colorable k-uniform hypergraph is hard. Known algorithms only guarantee an n^α-coloring for some $\alpha < 1$ (see [AKMH96]). The hardness results are more difficult to prove when k, c gets closer to 2.

In this thesis, we improve the hardness results for graph/hypergraph coloring and a more general problem of covering. We will describe the previous works and our results about these problems in the next three sections. The results are obtained using the theory of Probabilistically Checkable Proofs (PCPs), which have been widely used for proving such results. PCPs and their application to proving hardness results for approximation problems were discovered in the following papers [ALM+98, AS98, FGL+96]. Håstad gave reductions [Hås01] applicable for a variety of problems, and analysed them using Fourier analysis.

1 Graph Coloring

The results known for graph coloring requires a complexity assumption commonly known as the Unique Games Conjecture (UGC), which is stronger than the P≠ NP assumption. Starting with the work of Khot [Kho02b], it was shown that UGC, explains the lack of efficient approximation algorithms for a variety of problems (eg. Vertex Cover, MAX-CUT). Assuming UGC, Dinur, Mossel & Regev [DMR09] proved that for any constant C, efficient algorithms cannot find a C-coloring when given an almost[1] 3-colorable graph. Dinur & Shinkar [DS10] showed that hardness results for super-constant $C =$

[1]The exact statement of the result is, given a graph which is 3-colorable by dropping an ϵ fraction of the vertices, its hard to find an independent set of constant density.

$\Omega(\text{poly}(\log\log n))$, by assuming a stronger version of UGC.

Our Results

We improve the hardness result of Dinur & Shinkar exponentially [DHSV15]. That is, assuming the same UGC assumption with certain linearity requirement, we get hardness result for $C = 2^{\Omega(\text{poly}(\log\log n))}$. The previous reductions ([DMR09], [DS10]) followed the template of Håstad [Hås01], which employed a particular error correcting code known as the long code. As the name implies, this code has a large size which made the reductions inefficient. A shorter code called the low degree long code was proposed by Barak *et al.* [BGH+12] . Dinur & Guruswami [DG14] showed improved approximate covering hardness results (which we define in the third section) using this shorter code. We obtain our results by adapting this shorter code to the reduction of Dinur, Mossel & Regev [DMR09] for graph coloring.

2 Hypergraph Coloring

The study of hardness of approximate hypergraph coloring was initiated by Guruswami, Håstad & Sudan [GHS02]. They showed that, assuming NP doesn't have $2^{\text{poly}(\log n)}$-time algorithms, given a 2-colorable 4-uniform hypergraph, it is hard to find a $\text{poly}(\log\log n)$-coloring. Khot [Kho02a] showed the hardness of $C = (\log n)^{\Omega(q)}$ for q-colorable 4-uniform hypergraphs for $q \geq 7$, assuming NP does not have algorithms that runs in time $2^{\text{poly}(\log n)}$. Saket [Sak14] showed hardness for 2-colorable 4-uniform hypergraphs, using the similar assumption. Known algorithms only guarantee a factor of n^α for some $\alpha < 1$ [AKMH96].

Our Results

We showed the first super-polylogarithmic coloring hardness (i.e. $C >> \text{poly}(\log n)$) results [GHH+14], by giving a more efficient reduction using the low degree long code mentioned in the previous section. Subsequent to our work, Khot & Saket [KS14] got hardness results of $2^{(\log n)^{1/21}}$, by using the low degree long code with degree 2. Though their result was for 12-uniform hypergraphs. We further observed [Var14] that by combining their methods with ours, the same hardness results can be obtained for 4-uniform hypergraphs. Hence we improved the hardness results from $\mathrm{poly}(\log n)$ to $2^{(\log n)^{1/21}}$ for the case of 4-colorable 4-uniform hypergraphs. We also get hardness results for 3-uniform 3-colorable hypergraphs, with exponential improvement in the parameter C.

3 Covering Problems

The covering problem for constraint satisfaction (CSP) is a generalization of the hypergraph coloring problem, introduced by Guruswami, Håstad & Sudan [GHS02]. An instance of the problem consists of a hypergraph $G = (V, E)$ along with a *predicate* $P \subseteq \{0,1\}^k$ and a *literal* function $L : E \to \{0,1\}^k$. An *assignment* $f : V \to \{0,1\}$ *covers* an edge $e \in E$, if $f|_e \oplus L(e) \in P$[2]. A *cover* for a CSP instance is a set of assignments such that every edge is covered by one of the assignments. The goal of the covering problem is to find the minimum sized cover. Note that without the literal function, the problem becomes trivial for some predicates. For example, a 3-SAT instance without negations is always covered by the all 1 assignment. The covering problem can be thought of as a generalization of the coloring problem, since G has a cover of size t with the not-all-equal predicate $\{0,1\}^k \setminus \{\overline{0}, \overline{1}\}$ and the trivial literal function $L(e) = 0^k$ for every edge e iff it is 2^t-colorable (see [GHS02]). The *approximate covering* problem is defined similarly, as given a c-coverable instance, find a C-covering.

Some predicates like 3-SAT which has the *oddness* property that for every $x \in \{0,1\}^k$ either $x \in P$ or $\overline{x} \in P$, has a trivial algorithm with factor 2, since any assignment and its complement covers the CSP instance. Dinur & Kol [DK13] asked the question whether, the approximate covering problem is hard for any constant $C > c$, for all non-odd predicates. Assuming a slightly modified form of UGC, they proceeded to show that if a non-odd predicate has a pairwise independent distribution in its support then, this is

[2]For two k bit strings, the $\oplus$ operator does the coordinate-wise XOR operation.

indeed the case.

Our Results

We answer the question of Dinur & Kol in the affirmative [BHV15]. That is, the approximate covering problem for a non-odd predicate is hard for any constant $C > c$ (assuming the same conjecture as Dinur & Kol used). This leads to a complete characterization of predicates for which this result can be true, since there is a trivial 2-covering algorithm for odd predicates. We also show NP-hardness results, for the approximate covering problem with parameters $c = 2, C = \log\log n$, for a class of predicates. Previously such results were known due to Dinur & Kol for 4-LIN with $c = 2, C = \log\log\log n$.

Publications

Conference

[GHH+14] Venkat Guruswami, Prahladh Harsha, Johan Håstad, Srikanth Srinivasan, and Girish Varma. *Super-polylogarithmic hypergraph coloring hardness via low-degree long codes*. In *Proc. 46th ACM Symp. on Theory of Computing (STOC)*, pages 614–623. 2014. arXiv:1311.7407, doi:10.1145/2591796.2591882.

[DHSV15] Irit Dinur, Prahladh Harsha, Srikanth Srinivasan, and Girish Varma. *Derandomized Graph Product Results Using the Low Degree Long Code*. In Ernst W. Mayr and Nicolas Ollinger, eds., *32nd International Symposium on Theoretical Aspects of Computer Science (STACS 2015)*, volume 30 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 275–287. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2015. doi:http://dx.doi.org/10.4230/LIPIcs.STACS.2015.275.

[BHV15] Amey Bhangale, Prahladh Harsha, and Girish Varma. *A Charac-*

terization of Hard-to-cover CSPs. In David Zuckerman, ed., *30th Conference on Computational Complexity (CCC 2015)*, volume 33 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 280–303. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2015. doi:http://dx.doi.org/10.4230/LIPIcs.CCC.2015.280.

Manuscripts

[Var14] Girish Varma. *Reducing uniformity in khot-saket hypergraph coloring hardness reductions*. CoRR, abs/1408.0262, 2014.

References

[AKMH96] Noga Alon, Pierre Kelsen, Sanjeev Mahajan, and Ramesh Hariharan. *Approximate hypergraph coloring*. Nord. J. Comput., 3(4):425–439, 1996.

[ALM+98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. *Proof verification and the hardness of approximation problems*. J. ACM, 45(3):501–555, May 1998. (Preliminary version in *33rd FOCS*, 1992). eccc:TR98-008, doi:10.1145/278298.278306.

[AS98] Sanjeev Arora and Shmuel Safra. *Probabilistic checking of proofs: A new characterization of NP*. J. ACM, 45(1):70–122, January 1998. (Preliminary version in *33rd FOCS*, 1992). doi:10.1145/273865.273901.

[BGH+12] Boaz Barak, Parikshit Gopalan, Johan Håstad, Raghu Meka, Prasad Raghavendra, and David Steurer. *Making the long code shorter*. In *Proc. 53th IEEE Symp. on Foundations of Comp. Science (FOCS)*, pages 370–379. 2012. arXiv:1111.0405, doi:10.1109/FOCS.2012.83.

[BHV15] Amey Bhangale, Prahladh Harsha, and Girish Varma. *A Characterization of Hard-to-cover CSPs*. In David Zuckerman, ed., *30th Conference on Computational Complexity (CCC 2015)*, volume 33 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 280–303. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2015. doi:http://dx.doi.org/10.4230/LIPIcs.CCC.2015.280.

[Blu94] Avrim Blum. *New approximation algorithms for graph coloring*. Journal of the ACM, 41:470–516, 1994.

[DG14] Irit Dinur and Venkatesan Guruswami. *PCPs via low-degree long code and hardness for constrained hypergraph coloring*. In *Proc. 54th IEEE Symp. on Foundations of Comp. Science (FOCS)*, pages 340–349. 2014. eccc:TR13-122, doi:10.1109/FOCS.2013.44.

[DHSV15] Irit Dinur, Prahladh Harsha, Srikanth Srinivasan, and Girish Varma. *Derandomized Graph Product Results Using the Low Degree Long Code*. In Ernst W. Mayr and Nicolas Ollinger, eds., *32nd International Symposium on Theoretical Aspects of Computer Science (STACS 2015)*, volume 30 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 275–287. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2015. doi:http://dx.doi.org/10.4230/LIPIcs.STACS.2015.275.

[DK13] Irit Dinur and Gillat Kol. *Covering CSPs*. In *Proc. 28th IEEE Conference on Computational Complexity*, pages 207–218. 2013. eccc:TR12-088, doi:10.1109/CCC.2013.29.

[DMR09] Irit Dinur, Elchanan Mossel, and Oded Regev. *Conditional hardness for approximate coloring*. SIAM J. Computing, 39(3):843–873, 2009. (Preliminary version in *38th STOC*, 2006). arXiv:cs/0504062, doi:10.1137/07068062X.

[DS10] Irit Dinur and Igor Shinkar. *On the conditional hardness of coloring a 4-colorable graph with super-constant number of colors*. In Maria J. Serna, Ronen Shaltiel, Klaus Jansen, and José D. P. Rolim, eds., *Proc. 13th International Workshop on Randomization and Approximation Techniques in Computer Science (APPROX)*, volume 6302 of *LNCS*, pages 138–151. Springer, 2010.

[FGL+96] Uriel Feige, Shafi Goldwasser, László Lovász, Shmuel Safra, and Mario Szegedy. *Interactive proofs and the hardness of approximating cliques*. J. ACM, 43(2):268–292, March 1996. (Preliminary version in *32nd FOCS*, 1991). doi:10.1145/226643.226652.

[FK00] Uriel Feige and Joe Kilian. *Zero-knowledge and the chromatic number*. J. Computer and System Sciences, 60(2):337–353, April 2000. (Preliminary version in *12th Conference on Computational Complexity*, 1997).

[GHH+14] Venkat Guruswami, Prahladh Harsha, Johan Håstad, Srikanth Srinivasan, and Girish Varma. *Super-polylogarithmic hypergraph coloring hardness via low-degree long codes*. In *Proc. 46th ACM Symp. on Theory of Computing (STOC)*, pages 614–623. 2014. arXiv:1311.7407, doi:10.1145/2591796.2591882.

[GHS02] Venkatesan Guruswami, Johan Håstad, and Madhu Sudan. *Hardness of approximate hypergraph coloring*. SIAM Journal on Computing, 31(6):1663–1686, 2002. (Preliminary Version in *41st FOCS*, 2000). doi:10.1137/S0097539700377165.

[GK04] Venkatesan Guruswami and Sanjeev Khanna. *On the hardness of 4-coloring a 3-colorable graph*. SIAM J. Discrete Math., 18(1):30–40, 2004.

[Hås01] Johan Håstad. *Some optimal inapproximability results*. J. ACM, 48(4):798–859, July 2001. (Preliminary version in *29th STOC*, 1997). doi:10.1145/502090.502098.

[Kho02a] Subhash Khot. *Hardness results for approximate hypergraph coloring*. In *Proc. 34th ACM Symp. on Theory of Computing (STOC)*, pages 351–359. 2002. doi:10.1145/509907.509962.

[Khoo2b] ———. *On the power of unique 2-prover 1-round games*. In *Proc. 34th ACM Symp. on Theory of Computing (STOC)*, pages 767–775. 2002. doi:10.1145/509907.510017.

[KMS98] David R. Karger, Rajeev Motwani, and Madhu Sudan. *Approximate graph coloring by semidefinite programming*. J. ACM, 45(2):246–265, 1998. doi:10.1145/274787.274791.

[KS14] Subhash Khot and Rishi Saket. *Hardness of coloring 2-colorable 12-uniform hypergraphs with* $\exp(\log^{\Omega(1)} n)$ *colors*. In *Proc. 55th IEEE Symp. on Foundations of Comp. Science (FOCS)*, pages 206–215. 2014. doi:10.1109/FOCS.2014.30.

[KT14] Ken ichi Kawarabayashi and Mikkel Thorup. *Coloring 3-colorable graphs with* $o(n^{1/5})$ *colors*. In Ernst W. Mayr and Natacha Portier, eds., *STACS 2014*, volume 25 of *LIPIcs*, pages 458–469. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2014. doi:10.4230/LIPIcs.STACS.2014.458.

[Sak14] Rishi Saket. *Hardness of finding independent sets in 2-colorable hypergraphs and of satisfiable csps*. In *Proceedings of the 2014 IEEE 29th Conference on Computational Complexity*, CCC '14, pages 78–89. IEEE Computer Society, Washington, DC, USA, 2014. doi:10.1109/CCC.2014.16.

[Var14] Girish Varma. *Reducing uniformity in khot-saket hypergraph coloring hardness reductions*. CoRR, abs/1408.0262, 2014.

[Wig83] Avi Wigderson. *Improving the performance guarantee for approximate graph coloring*. J. ACM, 30(4):729–735, October 1983. doi:10.1145/2157.2158.