THEORY OF COMPUTING, Volume 8 (2012), pp. 239–267
www.theoryofcomputing.org

Tight Bounds on the Approximability of Almost-Satisfiable Horn SAT and Exact Hitting Set *

Venkatesan Guruswami[†] Yuan Zhou

Received: February 21, 2011; published: June 10, 2012.

Abstract: We study the approximability of two natural Boolean constraint satisfaction problems: Horn satisfiability and exact hitting set. Under the Unique Games conjecture, we prove the following *optimal* inapproximability and approximability results for finding an assignment satisfying as many constraints as possible given a *near-satisfiable* instance.

1. Given an instance of Max Horn-3SAT that admits an assignment satisfying $(1-\varepsilon)$ of its constraints for some small constant $\varepsilon > 0$, it is hard to find an assignment satisfying more than $1 - 1/O(\log(1/\varepsilon))$ of the constraints. This matches a linear programming based algorithm due to Zwick (STOC 1998), resolving the natural open question raised in that work concerning the optimality of the approximation bound.

   Given a $(1-\varepsilon)$-satisfiable instance of Max Horn-2SAT for some constant $\varepsilon > 0$, it is possible to find a $(1-2\varepsilon)$-satisfying assignment efficiently. This improves the algorithm given in Khanna et al. (2000) which finds a $(1-3\varepsilon)$-satisfying assignment,

*A conference version of this paper was earlier published at the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2011).

†Supported in part by a Packard Fellowship, NSF CCF 0953155, and US-Israel BSF grant 2008293.

ACM Classification: G.1.6

AMS Classification: 90C59

Key words and phrases: approximation algorithms, hardness of approximation, Unique Games Conjecture, Horn-SAT, exact hitting set

© 2012 Venkatesan Guruswami and Yuan Zhou
Licensed under a Creative Commons Attribution License DOI: 10.4086/toc.2010.v008a011

and also matches the $(1-c\varepsilon)$ hardness for any $c<2$ derived from vertex cover (under UGC).

2. An instance of Max 1-in-k-HS consists of a universe U and a collection $\mathcal{C}$ of subsets of U of size at most k, and the goal is to find a subset of U that intersects the maximum number of sets in $\mathcal{C}$ at a unique element. We prove that Max 1-in-k-HS is hard to approximate within a factor of $O(1/\log k)$ for every fixed integer k. This matches (up to constant factors) an easy factor $\Omega(1/\log k)$ approximation algorithm for the problem, and resolves a question posed in Guruswami and Trevisan (APPROX 2005).

   It is crucial for the above hardness that sets of size *up to k* are allowed; indeed, when all sets have size k, there is a simple factor $1/\mathrm{e}$-approximation algorithm.

Our hardness results are proved by constructing integrality gap instances for a semidefinite programming relaxation for the problems, and using Raghavendra's result (STOC 2008) to conclude that no algorithm can do better than the SDP assuming the UGC. In contrast to previous such constructions where the instances had a good SDP solution by design and the main task was bounding the integral optimum, the challenge in our case is the construction of appropriate SDP vectors and the integral optimum is easy to bound. Our algorithmic results are based on rounding appropriate linear programming relaxations.

1 Introduction

Schaefer proved long ago that there are only three non-trivial classes of Boolean constraint satisfaction problems (CSPs) for which satisfiability is polynomial time decidable [18]. These are LIN-mod-2 (linear equations modulo 2), 2-SAT, and Horn-SAT. The maximization versions of these problems (where the goal is to find an assignment satisfying the maximum number of constraints) are NP-hard, and in fact APX-hard, i. e., NP-hard to approximate within some constant factor bounded away from 1. An interesting special case of the maximization version is the following problem of "finding almost-satisfying assignments": *Given an instance which is $(1-\varepsilon)$-satisfiable (i. e., only ε fraction of constraints need to be removed to make it satisfiable for some small constant ε), can one efficiently find an assignment satisfying most (say, $1-f(\varepsilon)-o(1)$ where $f(\varepsilon)\to 0$ as $\varepsilon\to 0$) of the constraints?* [1]

The problem of finding almost-satisfying assignments was first suggested and studied in a beautiful paper by Zwick [19]. This problem seems well-motivated, as even if a Max CSP is APX-hard in general, in certain practical situations instances might be close to being satisfiable (for example, a small fraction of constraints might have been corrupted by noise). An algorithm that is able to satisfy most of the constraints of such an instance could be very useful.

As pointed out in [12], Schaefer's reductions together with the PCP theorem imply that the previous goal is NP-hard to achieve for any Boolean CSP for which the satisfiability problem is NP-complete. Indeed, all but the above three tractable cases of Boolean CSPs have a "gap at location 1," which means

[1] Throughout the paper, constraints could have weights, and by a "fraction α of constraints" we mean any subset of constraints whose total weight is a fraction α of the sum of the weights of all constraints. For CSPs with no unary constraints, the approximability of the weighted and unweighted versions are known to be the same [5].

that given a satisfiable instance it is NP-hard to find an assignment satisfying α fraction of the constraints for some constant $\alpha < 1$. This result has been extended to CSPs over arbitrary domains recently [11].

The natural question therefore is whether for the three tractable Boolean CSPs, LIN-mod-2, 2-SAT, and Horn-SAT, one can find almost-satisfying assignments in polynomial time. Effectively, the question is whether there are "robust" satisfiability checking algorithms that can handle a small number of inconsistent constraints and still produce a near-satisfying assignment.

With respect to the feasibility of finding almost-satisfying assignments, LIN-mod-2, 2-SAT, and Horn-SAT behave rather differently from each other. For LIN-mod-2, Håstad in his breakthrough paper [9] showed that for any $\varepsilon, \delta > 0$, finding a solution satisfying $1/2 + \delta$ of the equations of a $(1-\varepsilon)$-satisfiable instance is NP-hard. In fact, this result holds even when each equation depends on only 3 variables. Since just picking a random assignment satisfies $1/2$ the constraints in expectation, this shows, in a very strong sense, that there is no robust satisfiability algorithm for LIN-mod-2.

In sharp contrast to this extreme hardness for linear equations, Zwick [19] proved that for 2-SAT and Horn-SAT one can find almost-satisfying assignments in polynomial time. For Max 2SAT, Zwick gave a semidefinite programming (SDP) based algorithm that finds a $(1-O(\varepsilon^{1/3}))$-satisfying assignment (i. e., an assignment satisfying a fraction $(1-O(\varepsilon^{1/3}))$ of the constraints) given as input a $(1-\varepsilon)$-satisfiable instance. This algorithm was later improved to one that finds a $(1-O(\sqrt{\varepsilon}))$-satisfying assignment by Charikar, Makarychev, and Makarychev [4]. The $1-O(\sqrt{\varepsilon})$ bound is known to be best possible under the Unique Games conjecture (UGC) [13, 14]. In fact, this hardness result for Max 2SAT was the first application of the UGC and one of the main initial motivations for its formulation by Khot [13].

For Max Horn-SAT, Zwick gave a linear programming (LP) based algorithm to find an assignment satisfying $1-O(\log\log(1/\varepsilon)/\log(1/\varepsilon))$ of constraints of a $(1-\varepsilon)$-satisfiable instance. Recall that an instance of Horn-SAT is a CNF formula where each clause consists of at most one unnegated literal.[2] Equivalently, each clause is of the form x_i, $\overline{x_i}$, or $x_i \wedge x_2 \wedge \ldots \wedge x_k \to x_{k+1}$ for variables x_i. For Max Horn-3SAT, where each clause involves at most three variables, the algorithm finds a $(1-O(1/\log(1/\varepsilon)))$-satisfying assignment. Note that the fraction of unsatisfied constraints is *exponentially* worse for Max Horn-SAT compared to Max 2SAT.

Horn-SAT is a fundamental problem in logic and artificial intelligence. Zwick's robust Horn satisfiability algorithm shows the feasibility of solving instances where a small number of constraints are faulty and raises the following natural question, which was also explicitly raised in [19]. Is this $1/\log(1/\varepsilon)$ deficit inherent? Or could a more sophisticated algorithm, say based on an SDP relaxation instead of the LP relaxation used in [19], improve the deficit to something smaller (such as ε^b for some constant b as in the case of the SDP based algorithm for Max 2SAT)? It is known that for some absolute constant $c < 1$, it is NP-hard to find a $(1-\varepsilon^c)$-satisfying assignment given a $(1-\varepsilon)$-satisfiable instance of Max Horn-SAT [12].

In this work, we address the above question and resolve it (conditioned on the UGC), showing the $1/\log(1/\varepsilon)$ deficit to be inherent. We also investigate another problem, the "exact hitting set" problem for set sizes bounded by k, which has a very peculiar approximation behavior [8]. It admits a much better approximation algorithm on satisfiable instances, as well as when sets all have size exactly (or close to) k. We prove that these restrictions are inherent, and relaxing these rules out a constant factor approximation

[2] The dual variant dual-Horn-SAT is an instance of SAT where each clause has at most one negated literal and it is also polynomial time solvable.

algorithm (again, under the UGC). We describe our results in more detail below in Section 2.

Remark 1.1. For $(1-\varepsilon)$-satisfiable instances of Max 2-SAT, even the hardness of finding a $(1-\omega_\varepsilon(1)\varepsilon)$-satisfying assignment is not known without assuming the UGC (and the UGC implies the optimal $1-\Omega(\sqrt{\varepsilon})$ hardness bound). For Max Horn-SAT, as mentioned above, we know the NP-hardness of finding a $(1-\varepsilon^c)$-satisfying assignment for some absolute constant $c<1$. Under the UGC, we are able to pin down the exact asymptotic dependence on ε.

2 Our results and previous work

2.1 Horn-SAT

We prove the following hardness result concerning finding almost-satisfying assignments for Max Horn-SAT (in fact for the arity 3 case where all clauses involve at most 3 variables). In the sequel, we use the terminology "UG-hard" to mean at least as hard as refuting the Unique Games conjecture.

Theorem 2.1. *For some absolute constant $C>0$, for every $\varepsilon>0$, given a $(1-\varepsilon)$-satisfiable instance of Max Horn-3SAT, it is UG-hard to find an assignment satisfying more than a fraction $1-C/\log(1/\varepsilon)$ of the constraints.*

Zwick gave a polynomial time algorithm that finds a $(1-O(\log(k)/\log(1/\varepsilon)))$-satisfying assignment on input a $(1-\varepsilon)$-satisfiable instance of Max Horn-kSAT. Our inapproximability bound is therefore optimal up to the constant C, and resolves Zwick's question on whether his algorithm can be improved in the negative. (For arbitrary arity Horn-SAT, Zwick's algorithm has the slightly worse $1-O(\log\log(1/\varepsilon)/\log(1/\varepsilon))$ performance ratio; we do not show this to be tight.)

Theorem 2.1 shows that Max Horn-SAT has a very different quantitative behavior compared to Max 2SAT with respect to approximating near-satisfiable instances: the fraction of unsatisfied clauses $\Omega(1/\log(1/\varepsilon))$ is exponentially worse than the $O(\sqrt{\varepsilon})$ fraction that can be achieved for Max 2SAT.

A strong hardness result for Min Horn Deletion, the minimization version for Horn-SAT, was shown in [12]. It follows from their reduction that for some absolute constant $c<1$, it is NP-hard to find a $(1-\varepsilon^c)$-satisfying assignment given a $(1-\varepsilon)$-satisfiable instance of Max Horn-SAT. The constant c would be extremely close to 1 in this result as it is related to the soundness in Raz's parallel repetition theorem. While our inapproximability bound is stronger and optimal, we are only able to show UG-hardness and not NP-hardness.

In light of our strong hardness result for Max Horn-3SAT, we also consider the approximability of the arity two case. For Max Horn-2SAT, given a $(1-\varepsilon)$-satisfiable instance, an approximation preserving reduction from vertex cover shows that it is UG-hard to find a $(1-c\varepsilon)$-satisfying assignment for $c<2$. It is also shown in [12] that one can find a $(1-3\varepsilon)$-satisfying assignment efficiently. We improve the algorithmic bound (to the matching UG-hardness) by proving the following theorem, based on half-integrality of an LP relaxation for the problem.

Theorem 2.2. *Given a $(1-\varepsilon)$-satisfiable instance for Max Horn-2SAT, it is possible to find a $(1-2\varepsilon)$-satisfying assignment in polynomial time.*

2.2 Exact hitting set

We consider the "exact hitting set" problem where the goal is to find a subset that intersects a maximum number of sets from an input family at exactly one element. Formally,

Definition 2.3. Let $k \geqslant 2$ be a fixed integer. An instance of Max 1-in-k-HS consists of a universe $U = \{x_1, x_2, \ldots, x_n\}$ and a collection $\mathcal{C}$ of subsets of U each of size at most k. The objective is to find a subset $S \subseteq U$ that maximizes the number of sets $T \in \mathcal{C}$ for which $|S \cap T| = 1$. When all sets in $\mathcal{C}$ have size *equal to k*, we refer to the problem as Max 1-in-Ek-HS.

In addition to being a natural CSP, the exact hitting set problem arises in many contexts where one has to make unique choices from certain specified subsets. The complexity of this problem was investigated in [8] and [7], where applications of the problem to pricing, computing ad-hoc selective families for radio broadcasting, etc. are also discussed.

Our interest in this problem stems in part from the following peculiar approximability behavior of Max 1-in-k-HS, as pointed out in [8]. The Max 1-in-k-HS problem appears to be much easier to approximate on "satisfiable instances" (where a hitting set intersecting *all* subsets exactly once exists) or when all sets have size exactly equal to k (instead of at most k). In both these cases, there is a factor $1/\mathrm{e}$-approximation algorithm, and obtaining a $(1/\mathrm{e} + \varepsilon)$-approximation is NP-hard even when both restrictions hold simultaneously [8].

For Max 1-in-k-HS itself, the best approximation factor known to be possible in polynomial time is $\Omega(1/\log k)$. This is based on partitioning the collection $\mathcal{C}$ into $O(\log k)$ parts based on geometric intervals $[2^i, 2^{i+1})$ of set sizes, and running a simple randomized algorithm (that handles the case where all sets have sizes within a factor of two) on the sub-collection which has the most sets. Despite the simplicity and seeming naiveness of this algorithm, no factor $\omega(1/\log k)$ algorithm is known for the problem. No hardness factor better than the $1/\mathrm{e}$ bound (which holds even for Max 1-in-Ek-HS) is known either. Improving the gap in our understanding of the approximability of Max 1-in-k-HS was posed as an open question in [8].

For the case when k is not fixed but can also grow with the universe size n, a factor $(\log n)^{-\Omega(1)}$ hardness was shown in [7], under the assumption $\mathsf{NP} \not\subseteq \mathrm{TIME}(2^{n^\gamma})$ for some $\gamma > 0$. However, their method does not seem to be applicable to the case of bounded set size.

In this work, we prove the following tight result, establishing the difficulty of improving the simple $\Omega(1/\log k)$-approximation algorithm. This shows that it is hard to simultaneously do well on many different "scales" of set sizes. (If all set sizes are almost the same, then there is a constant factor approximation algorithm.)

Theorem 2.4. *For some absolute constant $C' > 0$, for every $\alpha > 0$, given a $(1 - 1/k^{1-\alpha})$-satisfiable instance of Max 1-in-k-HS, it is UG-hard to find a subset intersecting more than a fraction $C'/(\alpha \log k)$ of the sets exactly once.*

The gap in the above hardness result is also located at the "correct" satisfiability threshold, as we show the following complementary algorithmic result. Our algorithm in fact works for the more general Max 1-in-k-SAT problem where negated literals are allowed and the goal is to find an assignment for which a maximum number of clauses have exactly one literal set to true. For satisfiable instances of Max 1-in-k-SAT, a factor $1/\mathrm{e}$ approximation algorithm was given in [8].

Theorem 2.5. *For every constant $B > 1$, the following holds. There is a polynomial time algorithm that, given a $(1-1/(Bk))$-satisfiable instance of Max 1-in-k-SAT, finds a truth-assignment on variables satisfying exactly one literal in a fraction λ of the clauses, where $\lambda = (1-1/\sqrt{B})^2/\mathrm{e}$.*

3 Proof method

We construct integrality gap instances for a certain semidefinite programming relaxation (described in Section 3.1), and then use Raghavendra's theorem [17] to conclude that assuming the Unique Games conjecture, no algorithm can achieve an approximation ratio better than the SDP integrality gap.

In contrast to previous such integrality gap constructions (e. g., for Max Cut) where the instances had a good SDP solution "by design" and the technical core was bounding the integral optimum, in our case bounding the integral optimum is the easy part and the challenge is in the construction of appropriate SDP vectors. See Section 3.2 for an overview of our gap instances. It is also interesting that our SDP gaps match corresponding LP gaps. In general it seems like an intriguing question for which CSPs this is the case and therefore LPs suffice to get the optimal approximation ratio.

For our algorithmic results (see Section 3.3), we use a natural linear programming relaxation. For Max 1-in-k-SAT we show that randomized rounding gives a good approximation. The algorithm for Max Horn-2SAT proceeds by showing half-integrality of the LP.

3.1 The canonical SDP for Boolean CSPs and UG-Hardness

For Boolean CSP instances, we write $\mathcal{C}$ as the set of constraints over variables $x_1, x_2, \ldots, x_n \in \{0,1\}$. The SDP relaxation from [17], which we call the ***canonical SDP***, sets up for each constraint $C \in \mathcal{C}$ a local distribution π_C on all the truth-assignments $\{\sigma : X_C \to \{0,1\}\}$, where X_C is the set of variables involved in the constraint C. This is implemented via scalar variables $\pi_C(\sigma)$ which are required to be non-negative and satisfy $\sum_{\sigma : X_C \to \{0,1\}} \pi_C(\sigma) = 1$. For each variable x, two orthogonal vectors $\boldsymbol{v}_{(x,0)}$ and $\boldsymbol{v}_{(x,1)}$, corresponding to the events $x = 0$ and $x = 1$, are set up. The SDP requires for each variable x, $\boldsymbol{v}_{(x,0)} \cdot \boldsymbol{v}_{(x,1)} = 0$ and $\boldsymbol{v}_{(x,0)} + \boldsymbol{v}_{(x,1)} = \boldsymbol{I}$ where $\boldsymbol{I}$ is a global unit vector. (In the integral solution, one of the vectors $\boldsymbol{v}_{(x,1)}, \boldsymbol{v}_{(x,0)}$—based on the x's Boolean value—is intended to be $\boldsymbol{I}$ and the other one to be $\mathbf{0}$.)

Then, as constraint (3.5), the SDP does a consistency check: for two variables x, y (that need not be distinct) involved in the same constraint C, and for every $b_1, b_2 \in \{0,1\}$, the SDP insists that the inner product $\boldsymbol{v}_{(x,b_1)} \cdot \boldsymbol{v}_{(y,b_2)}$ equals $\mathbf{Pr}_{\sigma \in \pi_C}[(\sigma(x) = b_1) \wedge (\sigma(y) = b_2)]$.

$$\text{Maximize} \qquad \mathbf{E}_{C \in \mathcal{C}}[\mathbf{Pr}_{\sigma \in \pi_C}[C(\sigma) = 1]] \tag{3.1}$$

$$\text{subject to} \qquad \boldsymbol{v}_{(x_i,0)} \cdot \boldsymbol{v}_{(x_i,1)} = 0 \qquad \forall i \in [n]\,, \tag{3.2}$$

$$\boldsymbol{v}_{(x_i,0)} + \boldsymbol{v}_{(x_i,1)} = \boldsymbol{I} \qquad \forall i \in [n]\,, \tag{3.3}$$

$$\|\boldsymbol{I}\|^2 = 1\,, \tag{3.4}$$

$$\mathbf{Pr}_{\sigma \in \pi_C}[\sigma(x_i) = b_1 \wedge \sigma(x_j) = b_2] = \boldsymbol{v}_{(x_i,b_1)} \cdot \boldsymbol{v}_{(x_j,b_2)} \qquad \forall C \in \mathcal{C}, x_i, x_j \in C, b_1, b_2 \in \{0,1\}\,. \tag{3.5}$$

Note that if we discard all the vectors by removing constraints (3.2)$\sim$(3.4), and changing constraints (3.5) to $\mathbf{Pr}_{\sigma \in \pi_S}[\sigma(x_i) = b_1 \wedge \sigma(x_j) = b_2] = X_{(x_i,b_1),(x_j,b_2)}$, the SDP becomes a lifted LP in a Sherali-Adams system. We call this LP scheme the *lifted LP* in this paper.

The following striking theorem (Theorem 1.1 in [17]) states that once we have an integrality gap for the canonical SDP, we also get a matching UG-hardness. Below and elsewhere in the paper, a c vs. s gap instance is an instance with SDP optimum at least c and integral optimum at most s.

Theorem 3.1. *Let $1 > c > s > 0$. If a constraint satisfaction problem Λ admits a c vs. s integrality gap instance for the above canonical SDP, then for every constant $\eta > 0$, given an instance of Λ that admits an assignment satisfying $(c-\eta)$ of constraints, it is UG-Hard to find an assignment satisfying more than $(s+\eta)$ of constraints.*

To make our construction of integrality gaps easier, we notice the following simplification of the above SDP. Suppose we are given the global unit vector $\boldsymbol{I}$ and a vector $\boldsymbol{v}_x$ for each variable x in the CSP instance, subject to the following constraints:

$$\begin{aligned}(\boldsymbol{I}-\boldsymbol{v}_x)\cdot\boldsymbol{v}_x &= 0 \quad \forall \text{ variables } x\,, &(3.6)\\ \mathbf{Pr}_{\sigma\in\pi_C}[\sigma(x_i)=1\wedge\sigma(x_j)=1] &= \boldsymbol{v}_{x_i}\cdot\boldsymbol{v}_{x_j} \quad \forall C\in\mathcal{C}, x_i,x_j\in C\,. &(3.7)\end{aligned}$$

Defining $\boldsymbol{v}_{(x,1)} = \boldsymbol{v}_x$ and $\boldsymbol{v}_{(x,0)} = \boldsymbol{I}-\boldsymbol{v}_x$, it is easy to check that all constraints of the above SDP are satisfied. For instance, for variables x,y belonging to a constraint C,

$$\begin{aligned}\boldsymbol{v}_{(x,0)}\cdot\boldsymbol{v}_{(y,1)} &= (\boldsymbol{I}-\boldsymbol{v}_{(x,1)})\cdot\boldsymbol{v}_{(y,1)} = \|\boldsymbol{v}_{(y,1)}\|^2-\boldsymbol{v}_{(x,1)}\cdot\boldsymbol{v}_{(y,1)}\\ &= \mathbf{Pr}_{\sigma\in\pi_C}[\sigma(y)=1]-\mathbf{Pr}_{\sigma\in\pi_C}[(\sigma(x)=1)\wedge(\sigma(y)=1)]\\ &= \mathbf{Pr}_{\sigma\in\pi_C}[(\sigma(x)=0)\wedge(\sigma(y)=1)]\,,\end{aligned}$$

and other constraints of (3.5) follow similarly.

Henceforth in this paper, we will work with this streamlined canonical SDP with vector variables $\boldsymbol{I}$, $\{\boldsymbol{v}_x\}$, scalar variables corresponding to the local distributions π_C, constraints (3.6) and (3.7), and objective function (3.1).

3.2 Overview of construction of SDP gaps

Horn-3SAT In the concluding section of [19], Zwick remarks that there is an integrality gap for the LP he uses that matches his approximation ratio. Indeed such a LP gap is not hard to construct and we start by describing one such instance. The instance begins with clause x_1, and in the intermediate $(k-1)$ clauses, the i-th clause $x_1\wedge\ldots\wedge x_i\to x_{i+1}$ makes x_{i+1} true if all the previous clauses are satisfied. Then the last clause $\overline{x_k}$ generates a contradiction. Thus the optimal integral solution is at most $(1-1/k)$. On the other hand, one possible fractional solution starts with $x_1=(1-\varepsilon)$ for some $\varepsilon>0$. Then for $1\leqslant i<k$, by letting $(1-x_{i+1})=\sum_{j=1}^{i}(1-x_j)$, all the intermediate $(k-1)$ clauses are perfectly "satisfied" by the LP, while the gap $(1-x_{i+1})=2^{i-1}\varepsilon$ increases exponentially. Thus by letting $\varepsilon=1/2^{k-2}$, we get $x_k=0$ and the LP solution is at least $(1-1/2^{\Omega(k)})$. The instance gives a $(1-2^{-\Omega(k)})$ vs. $(1-1/k)$ LP integrality gap.

Now we convert this LP gap instance into an SDP gap instance in two steps. First, since we are going to give a gap instance for Max Horn-3SAT, we reduce the arity of the instance from k to 3. Then, we find a set of vectors for the LP solution to make it an SDP solution.

For the first step, to get an instance of Max Horn-3SAT, we introduce y_i which is intended to be $x_1 \wedge \ldots \wedge x_{i-1}$. For $1 \leqslant i < k$, we replace the intermediate clauses by $x_i \wedge y_i \to x_{i+1}$, and add $x_i \wedge y_i \to y_{i+1}$ to meet the intended definition of y_i. We call each of these two clauses as comprising one step . It is easy to show that for this instance there is a solution of value $(1-1/2^{\Omega(k)})$ even for the lifted LP. (The difference between the lifted LP and Zwick's LP is that the lift LP introduces local distributions over clauses which are consistent in the first and second moments, while Zwick's LP only has variables for singletons.)

Finding vectors for the SDP turns out to be more challenging. Note that if we want to perfectly satisfy all the intermediate clauses in SDP, we need to obey $\boldsymbol{v}_{x_i} \cdot \boldsymbol{v}_{y_i} \leqslant \|\boldsymbol{v}_{x_{i+1}}\|^2$ and $\boldsymbol{v}_{x_i} \cdot \boldsymbol{v}_{y_i} \leqslant \|\boldsymbol{v}_{y_{i+1}}\|^2$ for $1 \leqslant i < k$. Thus to make the norms $\|\boldsymbol{v}_{x_{i+1}}\|^2$ and $\|\boldsymbol{v}_{y_{i+1}}\|^2$ decrease fast (since we want $\|\boldsymbol{v}_{x_k}\|^2 = \|\boldsymbol{v}_{y_k}\|^2 = 0$), we need to make the inner product $\boldsymbol{v}_{x_i} \cdot \boldsymbol{v}_{y_i}$ decrease fast as well. But technically it is hard to make both kinds of quantities decrease at a high rate for all intermediate clauses. Our solution is to decrease the norms and inner products alternately. More specifically, we divide the intermediate clauses into blocks, each of which contains two consecutive steps. In the first step of each block, we need that the inner product is much smaller than the norms so that we can decrease the norms quickly, but we preserve the value of inner product. Thus we cannot do this step repeatedly, and we need the second step, where we decrease the inner product (while preserving the norms) in preparation to start the first step of the next block.

1-in-k Hitting Set We use a simple symmetric instance as our gap instance. Ideally, the instance includes all subsets of the universe with size at most k and we put uniform weights on sets of geometrically varying sizes (see Section 5.1 for our real gap instance which is slightly different). We first show that every subset intersects at most a (weighted) fraction $O(1/\log k)$ of the sets exactly once. Then, to prove a much better SDP solution, in contrast to Max Horn-3SAT, the main effort is in finding a good solution for lifted LP. Once we get a good solution for lifted LP, because of symmetry, the norms for all variables are defined to be the same value, and the pairwise inner products are also defined to be the same value. Then we only need to find vectors for a highly symmetric inner-product matrix, a step which is much easier than the counterpart of Max Horn-3SAT.

For the lifted LP, for each set in the instance, we place overwhelming weight on singleton subsets (only one variable is selected to be true) in all local distributions. This guarantees a good fractional solution. If we put all the weight on singletons though, the consistency check fails even for single-variable marginal distributions, whereas we need to ensure consistency of all pairwise-variable marginal distributions. Thus, for a feasible LP solution, we need to place some small weight on other subsets in order to obtain consistent marginal distributions. Indeed, we manage to generate a valid solution by giving an appropriate probability mass to the full set and all subsets of half the size in each local distribution.

3.3 Overview of algorithmic results

Our algorithmic results for Max Horn-2SAT and Max 1-in-k-SAT (Theorems 2.2 and 2.5 respectively) are obtained by rounding fractional solutions of appropriate linear programming (LP) relaxations.

The algorithm for Max Horn-2SAT is indeed a 2-approximation algorithm for Min Horn-2SAT Deletion problem (refer to Section 4.2 for the definition of Min Horn-2SAT Deletion). We prove a half-integrality property of the optimal solution to the natural LP relaxation of the problem, which can be viewed as a generalization of half-integrality property of (the natural LP for) Vertex Cover. We take the optimal solution of the natural LP relaxation, iteratively make every variable move towards half-integral values (0, 1, and $1/2$), while never increasing the value of the solution. This yields an optimal half-integral solution which can then be trivially rounded to obtain an integral solution that gives a factor 2 approximation.

For almost-satisfiable instances of Max 1-in-k-SAT, we prove that randomized rounding (according to the fractional value of any optimal LP solution) gives a constant factor approximation. This gives a robust version of the algorithm in [8] which achieved a factor $1/e$-approximation for (perfectly) satisfiable instances.

4 Approximability of Max Horn-3SAT

4.1 SDP gap and UG hardness for Max Horn-3SAT

4.1.1 Instance

We consider the following Max Horn-3SAT instance $\mathcal{I}_k^{\text{Horn}}$ parameterized by $k \geqslant 1$. (This construction is essentially the same as the one described in Section 3.2.)

$$\begin{array}{rrrl}
\text{Start point:} & & & x_0, y_0 \\
\text{Block } i\ (0 \leqslant i \leqslant k-1) & \text{Step } i.1: & x_{2i} \wedge y_{2i} \rightarrow x_{2i+1}, & x_{2i} \wedge y_{2i} \rightarrow y_{2i+1} \\
 & \text{Step } i.2: & x_{2i+1} \wedge y_{2i+1} \rightarrow x_{2i+2}, & x_{2i+1} \wedge y_{2i+1} \rightarrow y_{2i+2} \\
\text{End point:} & & x_{2k} \wedge y_{2k} \rightarrow x_{2k+1}, & x_{2k} \wedge y_{2k} \rightarrow y_{2k+1} \\
 & & \overline{x_{2k+1}}, & \overline{y_{2k+1}}
\end{array}$$

It is easy to see this instance contains $(4k+6)$ clauses, and cannot be completely satisfied. Thus we have:

Lemma 4.1. *Every Boolean assignment satisfies at most a fraction $1-1/(4k+6)$ of the clauses of $\mathcal{I}_k^{\text{Horn}}$.*

4.1.2 Construction of a good SDP solution

We will work with the SDP in simplified form described at the end of Section 3.1. Recall that the SDP requires a local distribution for each clause, and uses vectors to check the consistency on every pair of variables that belong to the clause. To construct a good solution for the SDP, we want to first find a good solution in the scalar part (i. e., local distributions), and then construct vectors which meet the consistency requirement. But it is difficult to construct a lot of vectors which meet all the requirements simultaneously. Thus, we break down the whole construction task into small pieces, each of which is easy to deal with. As long as there are solutions to these small pieces, and the solutions agree with each other on some interfaces, we can coalesce the small solutions together and come up with a global solution. The following definition and claim formally help us bring down the difficulty, and focus on one local block of variables at a time.

Definition 4.2 (partial solution)**.** Let $\mathcal{C}' \subseteq \mathcal{C}$ be a subset of clauses.

$$f = \{\pi_C = \pi_C(f), \boldsymbol{v}_x = \boldsymbol{v}_x(f), \boldsymbol{I} = \boldsymbol{I}(f) \mid \forall C \in \mathcal{C}', x \in C\}$$

is said to be a *partial solution* on $\mathcal{C}'$, if all constraints of the SDP restricted to the subset of variables defined in f are satisfied.

Claim 4.3. *Let $\mathcal{C}_1, \mathcal{C}_2 \subseteq \mathcal{C}$ be two disjoint set of clauses. Let f and g be partial solutions on $\mathcal{C}_1, \mathcal{C}_2$ respectively. If for all $\boldsymbol{v}_1, \boldsymbol{v}_2$ (not necessarily distinct) defined in both f and g, $\boldsymbol{v}_1(f)\cdot\boldsymbol{v}_2(f) = \boldsymbol{v}_1(g)\cdot\boldsymbol{v}_2(g)$, then there exists a partial solution, namely h, for $\mathcal{C}_1 \cup \mathcal{C}_2$, such that $\forall C_1 \in \mathcal{C}_1, C_2 \in \mathcal{C}_2$, $\pi_{C_1}(h) = \pi_{C_1}(f), \pi_{C_2}(h) = \pi_{C_2}(g)$.*

Proof. Let X be the set of variables x for which $\boldsymbol{v}_x(f)$ and $\boldsymbol{v}_x(g)$ are both defined. Denote

$$V_f = \{\boldsymbol{v}_x(f) \mid x \in X\} \cup \{\boldsymbol{I}(f)\} \quad \text{and} \quad V_g = \{\boldsymbol{v}_x(g) \mid x \in X\} \cup \{\boldsymbol{I}(g)\}.$$

Since the dot products of every pair of vectors in V_f exactly equals the dot product between the corresponding pair in V_g, there is a rotation (orthogonal transformation) T such that $\boldsymbol{I}(f) = T\boldsymbol{I}(g)$ and for all $x \in X$, $\boldsymbol{v}_x(f) = T\boldsymbol{v}_x(g)$.

Now define the partial solution g' as $\pi_C(g') = \pi_C(g)$ for all $C \in \mathcal{C}_2$ and $\boldsymbol{v}_x(g') = T\boldsymbol{v}_x(g)$, $\boldsymbol{I}(g') = T\boldsymbol{I}(g)$ for all $x \in C \in \mathcal{C}_2$. Obviously f and g' agree on all the scalar and vector variables that are defined in both f and g'. Letting

$$\boldsymbol{v}_x(h) = \begin{cases} \boldsymbol{v}_x(f) & x \in C \in \mathcal{C}_1 \\ \boldsymbol{v}_x(g') & x \in C \in \mathcal{C}_2 \end{cases}, \quad \pi_C(h) = \begin{cases} \pi_C(f) & C \in \mathcal{C}_1 \\ \pi_C(g') & C \in \mathcal{C}_2 \end{cases},$$

it is easy to see h is a partial solution on $\mathcal{C}_1 \cup \mathcal{C}_2$. □

By the above lemma, if we establish the following lemma which constructs a good partial solution on each block (the proof of which is deferred to Section 4.1.3), it is then easy to get a good global solution.

Lemma 4.4. *For each block i $(0 \leqslant i \leqslant k-1)$, each $0 < c \leqslant 0.2$, let $r_c = 1.5(1+c)/(1.5+c) > 1$, and for each $0 < p \leqslant 1/((1+c)r_c)$, there is a partial solution f which completely satisfies all the clauses in block i (by local distributions), and with following properties,*

$$\begin{aligned} \|\boldsymbol{v}_{x_{2i}}(f)\|^2 = \|\boldsymbol{v}_{y_{2i}}(f)\|^2 &= 1-p, \\ \boldsymbol{v}_{x_{2i}}(f)\cdot\boldsymbol{v}_{y_{2i}}(f) &= 1-(1+c)p, \\ \|\boldsymbol{v}_{x_{2i+2}}(f)\|^2 = \|\boldsymbol{v}_{y_{2i+2}}(f)\|^2 &= 1-r_c p, \\ \boldsymbol{v}_{x_{2i+2}}(f)\cdot\boldsymbol{v}_{y_{2i+2}}(f) &= 1-(1+c)r_c p. \end{aligned}$$

As explained in Section 3.2, in the first step (the step to decrease norms), to make $\|\boldsymbol{v}_{x_{2i+2}}(f)\|^2$ and $\|\boldsymbol{v}_{y_{2i+2}}(f)\|^2$ much smaller than $\|\boldsymbol{v}_{x_{2i}}(f)\|^2$ and $\|\boldsymbol{v}_{y_{2i}}(f)\|^2$, we need the inner product $\boldsymbol{v}_{x_{2i}}(f)\cdot\boldsymbol{v}_{y_{2i}}(f)$ to be small. This is why we introduce c, and require that $\boldsymbol{v}_{x_{2i}}(f)\cdot\boldsymbol{v}_{y_{2i}}(f) = 1-(1+c)p$. The larger c is, the faster the norms decrease. But due to technical reasons, in the second step (the step to decrease the inner product), we are not able to decrease the inner product fast when it is much smaller than the norms. So we put a upper bound $c \leqslant 0.2$ in the lemma.

Using Lemma 4.4 together with Claim 4.3, we immediately get the following corollary.

Corollary 4.5. *For the union of block* 0 *to block* k' $(0 \leqslant k' \leqslant k-1)$, *given parameters*

$$0 < c \leqslant 0.2 \quad \textit{and} \quad 0 < p \leqslant \frac{1}{(1+c)r_c^{k'+1}},$$

there is a partial solution g *which completely satisfies all the clauses, and with following properties,*

$$\begin{aligned}\|\boldsymbol{v}_{x_0}(g)\|^2 = \|\boldsymbol{v}_{y_0}(g)\|^2 &= 1-p,\\ \boldsymbol{v}_{x_0}(g)\cdot\boldsymbol{v}_{y_0}(g) &= 1-(1+c)p,\\ \|\boldsymbol{v}_{x_{2k'+2}}(g)\|^2 = \|\boldsymbol{v}_{y_{2k'+2}}(g)\|^2 &= 1-r_c^{k'+1}p,\\ \boldsymbol{v}_{x_{2k'+2}}(g)\cdot\boldsymbol{v}_{y_{2k'+2}}(g) &= 1-(1+c)r_c^{k'+1}p.\end{aligned}$$

Proof. Apply induction on k'. The basis case $k'=0$ is exactly Lemma 4.4. For $k'>0$, by induction hypothesis there is a partial solution g' satisfying all the clauses of the union of blocks 0 to $k'-1$ with the same parameter c,p. By Lemma 4.4, there is a partial solution f satisfying all the clauses of block k' with parameter $c, r_c^{k'}p$. Since g' and f agree on pairwise inner-products over the definition of $\{\boldsymbol{v}_{x_{2k'}}, \boldsymbol{v}_{y_{2k'}}\}$, by Claim 4.3, there is a partial solution g on the union of blocks 0 to k' completely satisfying all the clauses. □

With the above pieces in place, we now come to the final SDP solution.

Lemma 4.6. *The optimal SDP solution for the instance* $\mathcal{I}_k^{\mathrm{Horn}}$ *has value at least*

$$1-\frac{1}{(2k+3)1.05^k}.$$

Proof. By Corollary 4.5, for any $0<c\leqslant 0.2$, by setting $p = 1/((1+c)r_c^k)$, there is a partial solution g completely satisfying all the clauses of all the blocks, with

$$\begin{aligned}\|\boldsymbol{v}_{x_0}(g)\|^2 = \|\boldsymbol{v}_{y_0}(g)\|^2 &= 1-\frac{1}{(1+c)r_c^k},\\ \|\boldsymbol{v}_{x_{2k}}(g)\|^2 = \|\boldsymbol{v}_{y_{2k}}(g)\|^2 &= c/(1+c),\\ \boldsymbol{v}_{x_{2k}}(g)\cdot\boldsymbol{v}_{y_{2k}}(g) &= 0.\end{aligned}$$

Based on g, we define a local distribution on two "Start point" clauses by making x_0 (or y_0) equal 1 with probability $1-p$. At "End point," we define the local distribution on clause $x_{2k}\wedge y_{2k}\to x_{2k+1}$ as

$$\begin{aligned}\mathbf{Pr}_\pi[x_{2k}=1\wedge y_{2k}=0\wedge x_{2k+1}=0] &= c/(1+c),\\ \mathbf{Pr}_\pi[x_{2k}=0\wedge y_{2k}=1\wedge x_{2k+1}=0] &= c/(1+c),\\ \mathbf{Pr}_\pi[x_{2k}=0\wedge y_{2k}=0\wedge x_{2k+1}=0] &= (1-c)/(1+c).\end{aligned}$$

And a similar distribution for the clause $x_{2k}\wedge y_{2k}\to y_{2k+1}$ can be defined (by replacing x_{2k+1} by y_{2k+1} in the equations above). The distribution on clauses $\overline{x_{2k+1}}$ and $\overline{y_{2k+1}}$ never picks the corresponding variable to be 1. By defining $\boldsymbol{v}_{x_{2k+1}}$ and $\boldsymbol{v}_{y_{2k+1}}$ to be zero vectors, we note that the distributions are consistent with vectors. Thus the solution we construct is valid.

On the other hand, note that all the distributions locally satisfy the clauses, except for the distributions at "Start point" satisfy the corresponding clause with probability $1-1/((1+c)r_c^k)$. Thus the SDP solution has value

$$1-\frac{2}{(4k+6)(1+c)r_c^k}=1\geqslant 1-\frac{1}{(2k+3)r_c^k}.$$

By setting $c=0.2$, we get $r_c\geqslant 1.05$. Thus the best SDP solution has value at least $1-1/((2k+3)1.05^k)$. □

Combining Lemma 4.1 and Lemma 4.6, we get the following theorem.

Theorem 4.7. *$\mathcal{I}_k^{\mathrm{Horn}}$ is a $(1-\varepsilon)$ vs. $(1-\Omega(1/\log(1/\varepsilon)))$ gap instance of Max Horn-3SAT for the canonical SDP relaxation.*

Together with Theorem 3.1, Theorem 4.7 implies our main result, Theorem 2.1, on Max Horn-SAT.

4.1.3 Proof of the key Lemma 4.4

For block i, denote the clauses in Step $i.1$ by C_{1x} and C_{1y}, and the clauses in Step $i.2$ by C_{2x} and C_{2y}. We first construct partial solutions on Step $i.1$ and Step $i.2$ separately, as follows.

Partial solution on Step $i.1$ We first define a local distribution on satisfying assignments for C_{1x} as follows, and C_{1y} in a similar way (by replacing x_{2i+1} by y_{2i+1} in following equations).

$$\begin{aligned}
\mathbf{Pr}_{\pi_{C_{1x}}}[x_{2i}=1\wedge y_{2i}=1\wedge x_{2i+1}=1] &= 1-(1+c)p,\\
\mathbf{Pr}_{\pi_{C_{1x}}}[x_{2i}=1\wedge y_{2i}=0\wedge x_{2i+1}=0] &= cp,\\
\mathbf{Pr}_{\pi_{C_{1x}}}[x_{2i}=0\wedge y_{2i}=1\wedge x_{2i+1}=0] &= cp,\\
\mathbf{Pr}_{\pi_{C_{1x}}}[x_{2i}=0\wedge y_{2i}=0\wedge x_{2i+1}=1] &= (1+c-r_c)p=\frac{(1+c)c}{1.5+c}\cdot p,\\
\mathbf{Pr}_{\pi_{C_{1x}}}[x_{2i}=0\wedge y_{2i}=0\wedge x_{2i+1}=0] &= (r_c-2c)p=\frac{1.5-1.5c-2c^2}{1.5+c}\cdot p.
\end{aligned}$$

Recall $r_c=1.5(1+c)/(1.5+c)$. Note that all the probabilities are defined to be non-negative values by the range of c and p, and they sum up to 1.

We observe the following inner-product matrix A over $\boldsymbol{I},\boldsymbol{v}_{x_{2i}},\boldsymbol{v}_{y_{2i}},\boldsymbol{v}_{x_{2i+1}},\boldsymbol{v}_{y_{2i+1}}$ is consistent with the local distributions on satisfying assignments for C_{1x} and C_{1y}.

$$A=\begin{bmatrix}
1 & 1-p & 1-p & 1-r_cp & 1-r_cp\\
1-p & 1-p & 1-(1+c)p & 1-(1+c)p & 1-(1+c)p\\
1-p & 1-(1+c)p & 1-p & 1-(1+c)p & 1-(1+c)p\\
1-r_cp & 1-(1+c)p & 1-(1+c)p & 1-r_cp & 1-(1+c)p\\
1-r_cp & 1-(1+c)p & 1-(1+c)p & 1-(1+c)p & 1-r_cp
\end{bmatrix}.$$

By Claim 4.8 (at the end of this section) we know that A is positive semidefinite, and therefore there is a set of vectors consistent with our local distributions, i. e., we get a partial solution on Step $i.1$.

Partial solution on Step $i.2$ We define the local distribution on satisfying assignments for C_{2x} as follows. The distribution for C_{2y} is defined in a similar way (by replacing x_{2i+2} with y_{2i+2} in the following equations). Let $q = r_c p$ and $\varepsilon = c/1.5$.

$$\begin{aligned}
\mathbf{Pr}_{\pi_{C_{2x}}}[x_{2i+1}=1 \wedge y_{2i+1}=1 \wedge x_{2i+2}=1] &= 1-(1+\varepsilon)q\,,\\
\mathbf{Pr}_{\pi_{C_{2x}}}[x_{2i+1}=1 \wedge y_{2i+1}=0 \wedge x_{2i+2}=0] &= \varepsilon q\,,\\
\mathbf{Pr}_{\pi_{C_{2x}}}[x_{2i+1}=0 \wedge y_{2i+1}=1 \wedge x_{2i+2}=0] &= \varepsilon q\,,\\
\mathbf{Pr}_{\pi_{C_{2x}}}[x_{2i+1}=0 \wedge y_{2i+1}=0 \wedge x_{2i+2}=1] &= \varepsilon q\,,\\
\mathbf{Pr}_{\pi_{C_{2x}}}[x_{2i+1}=0 \wedge y_{2i+1}=0 \wedge x_{2i+2}=0] &= (1-2\varepsilon)q\,.
\end{aligned}$$

Note that all the probabilities are defined to be non-negative values by the range of c and p, and they sum up to 1.

Then note that the following inner-product matrix B over $\boldsymbol{I}, \boldsymbol{v}_{x_{2i+1}}, \boldsymbol{v}_{y_{2i+1}}, \boldsymbol{v}_{x_{2i+2}}, \boldsymbol{v}_{y_{2i+2}}$ is consistent with the local distribution.

$$B = \begin{bmatrix}
1 & 1-q & 1-q & 1-q & 1-q\\
1-q & 1-q & 1-(1+\varepsilon)q & 1-(1+\varepsilon)q & 1-(1+\varepsilon)q\\
1-q & 1-(1+\varepsilon)q & 1-q & 1-(1+\varepsilon)q & 1-(1+\varepsilon)q\\
1-q & 1-(1+\varepsilon)q & 1-(1+\varepsilon)q & 1-q & 1-(1+1.5\varepsilon)q\\
1-q & 1-(1+\varepsilon)q & 1-(1+\varepsilon)q & 1-(1+1.5\varepsilon)p & 1-q
\end{bmatrix}.$$

Again by Claim 4.8, B is positive semidefinite, and therefore there is a set of vectors consistent with local distributions—we have constructed a partial solution on Step $i.2$.

Combining the two partial solutions We first check that under our parameter setting, partial solutions on Step $i.1$ and Step $i.2$ agree on pairwise inner-products between their shared vectors $\boldsymbol{I}, \boldsymbol{v}_{x_{2i+1}}, \boldsymbol{v}_{y_{2i+1}}$.

- For $\langle \boldsymbol{I}, \boldsymbol{I} \rangle$, we have $1 = 1$.
- For $\langle \boldsymbol{I}, \boldsymbol{v}_{x_{2i+1}} \rangle$, we have $1 - r_c p = 1 - q$.
- For $\langle \boldsymbol{I}, \boldsymbol{v}_{y_{2i+1}} \rangle$, we have $1 - r_c p = 1 - q$.
- For $\langle \boldsymbol{v}_{x_{2i+1}}, \boldsymbol{v}_{x_{2i+1}} \rangle$, we have $1 - r_c p = 1 - q$.
- For $\langle \boldsymbol{v}_{x_{2i+1}}, \boldsymbol{v}_{y_{2i+1}} \rangle$, we have $1-(1+c)p = 1-(1+c/1.5)r_c p = 1-(1+\varepsilon)q$.
- For $\langle \boldsymbol{v}_{y_{2i+1}}, \boldsymbol{v}_{y_{2i+1}} \rangle$, we have $1 - r_c p = 1 - q$.

Thus, there is a partial solution on block i, with

$$\begin{aligned}
\|\boldsymbol{v}_{x_{2i}}(f)\|^2 = \|\boldsymbol{v}_{y_{2i}}(f)\|^2 &= 1-p\,,\\
\boldsymbol{v}_{x_{2i}}(f) \cdot \boldsymbol{v}_{y_{2i}}(f) &= 1-(1+c)p\,,\\
\|\boldsymbol{v}_{x_{2i+2}}(f)\|^2 = \|\boldsymbol{v}_{y_{2i+2}}(f)\|^2 &= 1-q = 1-r_c p\,,\\
\boldsymbol{v}_{x_{2i+2}}(f) \cdot \boldsymbol{v}_{y_{2i+2}}(f) &= 1-(1+1.5\varepsilon)q = 1-(1+c)r_c p\,. \quad \square
\end{aligned}$$

Finally, we establish the two positive semidefinite matrices used in the proof above.

Claim 4.8. *Given $0 < c \leqslant 0.2$, $0 < p \leqslant r_c/(1+c)$, $q = r_c p$, $\varepsilon = c/1.5$, the following two matrices are positive semidefinite:*

$$A = \begin{bmatrix} 1 & 1-p & 1-p & 1-r_c p & 1-r_c p \\ 1-p & 1-p & 1-(1+c)p & 1-(1+c)p & 1-(1+c)p \\ 1-p & 1-(1+c)p & 1-p & 1-(1+c)p & 1-(1+c)p \\ 1-r_c p & 1-(1+c)p & 1-(1+c)p & 1-r_c p & 1-(1+c)p \\ 1-r_c p & 1-(1+c)p & 1-(1+c)p & 1-(1+c)p & 1-r_c p \end{bmatrix},$$

$$B = \begin{bmatrix} 1 & 1-q & 1-q & 1-q & 1-q \\ 1-q & 1-q & 1-(1+\varepsilon)q & 1-(1+\varepsilon)q & 1-(1+\varepsilon)q \\ 1-q & 1-(1+\varepsilon)q & 1-q & 1-(1+\varepsilon)q & 1-(1+\varepsilon)q \\ 1-q & 1-(1+\varepsilon)q & 1-(1+\varepsilon)q & 1-q & 1-(1+1.5\varepsilon)q \\ 1-q & 1-(1+\varepsilon)q & 1-(1+\varepsilon)q & 1-(1+1.5\varepsilon)p & 1-q \end{bmatrix}.$$

Proof. Let J be the all 1 matrix, E_1 be the matrix with 1 in entry $(1,1)$ as the only one non-zero entry. We also define $E_{i,j}, F_{i,j}$ and $G_{i,j}$ as matrices with only four non-zero entries located in the intersections of Column i, j and Row i, j. The sub-matrices of $E_{i,j}, F_{i,j}$ and $G_{i,j}$ on Column i, j and Row i, j are defined as

$$(\text{for } E_{i,j}) \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad (\text{for } F_{i,j}) \begin{bmatrix} 2 & 1 \\ 1 & 0.5 \end{bmatrix} \quad \text{and} \quad (\text{for } G_{i,j}) \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}.$$

Clearly, all of J, E_1, $E_{i,j}$, $F_{i,j}$ and $G_{i,j}$ are positive semidefinite matrices.

Then we can write A as

$$\begin{aligned} A &= (1-(1+c)p)J + cp(E_{1,2}+E_{1,3}) + (1+c-r_c)p(E_{1,4}+E_{1,5}) + (2r_c - 1 - 3c)pE_1 \\ &= (1-(1+c)p)J + cp(E_{1,2}+E_{1,3}) + \frac{(1+c)c}{1.5+c} \cdot p(E_{1,4}+E_{1,5}) + \frac{1.5-2.5c-3c^2}{1.5+c} \cdot pE_1\,. \end{aligned}$$

Note that all the coefficients before matrices are non-negative within the range of c. Since A can be written as the sum of several positive semidefinite matrices, A is positive semidefinite.

For matrix B, note that

$$B = (1-(1+\varepsilon)q)J + \varepsilon q(E_{1,2}+E_{1,3}+F_{1,4}+F_{1,5}) + 0.5\varepsilon q G_{4,5} + (1-5\varepsilon)qE_1\,.$$

Clearly, as long as $5\varepsilon = 5c/1.5 < 1$, B can be expressed as sum of positive semidefinite matrices, and hence B is positive semidefinite. ☐

4.2 Algorithm for Min Horn-2SAT Deletion and Max Horn-2SAT

In the Min Horn-2SAT Deletion problem, we are given a Horn-2SAT instance, and the goal is to find a subset of clauses of minimum total weight whose deletion makes the instance satisfiable. A factor 3 approximation algorithm for Min Horn-2SAT Deletion is given in [12]. Here we improve the approximation ratio to 2. Note that following reduction from vertex cover is approximation preserving: given a graph with n vertices, for each edge (i, j), introduce a clause $\bar{x}_i \vee \bar{x}_j$ of weight n; for each vertex i, introduce a clause x_i of weight 1. Therefore, by the inapproximability result for vertex cover [15], our approximation algorithm for Min Horn-2SAT Deletion is optimal under the UGC.

Our motivation to study Min Horn-2SAT Deletion in the context of this paper is to pin down the fraction of clauses one can satisfy in a $(1-\varepsilon)$-satisfiable instance of Horn-2SAT: we can satisfy a fraction $(1-2\varepsilon)$ of clauses (even in the weighted case), and satisfying a $(1-c\varepsilon)$ fraction is hard for $c<2$ assuming that vertex cover does not admit a c-approximation for any constant $c<2$.

In this section, we prove the following theorem by showing half-integrality of a natural LP relaxation for the problem.

Theorem 4.9. *There is a polynomial-time 2-approximation algorithm for Min Horn-2SAT Deletion problem.*

Theorem 2.2 (on approximating near-satisfiable instances of Max Horn-2SAT) is a direct corollary of Theorem 4.9. □

4.2.1 LP formulation

We find it slightly more convenient to present the algorithm for dual Horn-2SAT where each clause has at most one negated literal. (So the clauses are of the form x, $\bar{x}$, $x\vee y$, or $x\to y$, for variables x,y.) Let $w_{ij}^{(D)}>0$ be the weight imposed on the disjunction constraint $x_i\vee x_j$ (for each pair of i, j such that $i<j$), and $w_{ij}^{(I)}>0$ be the weight imposed on the implication constraint $x_i\to x_j$ (for each pair of i, j such that $i\neq j$). For each variable x_i, let $w_i^{(T)}$ be the weight on x_i being true (i. e., $x_i=1$), and $w_i^{(F)}$ be the weight on x_i being false (i. e., $x_i=0$). Then we write the following LP relaxation, where each real variable y_i corresponds to the integer variable x_i:

$$\begin{array}{lll}
\text{Minimize} & \multicolumn{2}{l}{\sum_{i\in V} w_i^{(T)}(1-y_i)+\sum_{i\in V} w_i^{(F)}y_i+\sum_{i<j} w_{ij}^{(D)}z_{ij}^{(D)}+\sum_{i\neq j} w_{ij}^{(I)}z_{ij}^{(I)}} \\
\text{subject to} & z_{ij}^{(D)} \geqslant 1-y_i-y_j & \forall i<j\,, \\
& z_{ij}^{(I)} \geqslant y_i-y_j & \forall i\neq j\,, \\
& z_{ij}^{(D)} \geqslant 0 & \forall i<j\,, \\
& z_{ij}^{(I)} \geqslant 0 & \forall i\neq j\,, \\
& y_i \in [0,1] & \forall i\in V\,.
\end{array}$$

Let OPT be the optimal value of the integral solution, and $\mathrm{OPT_{LP}}$ be the optimal value of the LP solution. We have $\mathrm{OPT_{LP}}\leqslant\mathrm{OPT}$.

4.2.2 Half-integrality and rounding

Given a LP solution $f=\{z_{ij}^{(D)},z_{ij}^{(I)},y_i\}$, we can assume $z_{ij}^{(D)}=\max\{1-y_i-y_j,0\}$ and $z_{ij}^{(I)}=\max\{y_i-y_j,0\}$ to minimize $\mathsf{Val}(f)$. Thus, we only need $f=\{y_i\}$ to characterize a solution, and we have

$$\mathsf{Val}(f)=\sum_{i\in V} w_i^{(T)}(1-y_i)+\sum_{i\in V} w_i^{(F)}y_i+\sum_{i<j} w_{ij}^{(D)}\max\{1-y_i-y_j,0\}+\sum_{i\neq j} w_{ij}^{(I)}\max\{y_i-y_j,0\}\,.$$

Lemma 4.10. *There is a polynomial-time algorithm that, given a solution $f = \{y_i\}$ to the above LP, converts f into another solution $f^* = \{y_i^*\}$ such that each y_i^* is half-integral, i. e., $y_i^* \in \{0, 1, 1/2\}$, and* $\mathsf{Val}(f^*) \leqslant \mathsf{Val}(f)$.

Proof. We run Algorithm 1 whose input is the LP formulation and one of the solutions $f = \{y_i\}$, and whose output is the desired f^*. At a high level, the algorithm iteratively moves the LP variables that are not half integral to half integral values (according to some strategy), and we need to prove that at each step of the iteration, the algorithm creates a new valid LP solution whose objective value is no greater than the previous one.

Algorithm 1 Round any LP solution $f = \{y_i\}$ to a half-integral solution f^*, with $\mathsf{Val}(f^*) \leqslant \mathsf{Val}(f)$

while $\exists i \in V : y_i \notin \{0, 1, 1/2\}$ **do**
    choose $k \in V$, such that $y_k \notin \{0, 1, 1/2\}$ (arbitrarily)
    if $y_k < 1/2$ **then**
        $p \leftarrow y_k$
    else
        $p \leftarrow 1 - y_k$
    end if
    $S \leftarrow \{i : y_i = p\}$, $S' \leftarrow \{i : y_i = 1 - p\}$
    $a \leftarrow \max\{y_i : y_i < p, 1 - y_i : y_i > 1 - p, 0\}$, $b \leftarrow \min\{y_i : y_i > p, 1 - y_i : y_i < 1 - p, 1/2\}$
    $f^{(a)} \leftarrow \{y_i^{(a)} = a\}_{i \in S} \cup \{y_i^{(a)} = 1 - a\}_{i \in S'} \cup \{y_i^{(a)} = y_i\}_{i \in V \setminus (S \cup S')}$
    $f^{(b)} \leftarrow \{y_i^{(b)} = b\}_{i \in S} \cup \{y_i^{(b)} = 1 - b\}_{i \in S'} \cup \{y_i^{(b)} = y_i\}_{i \in V \setminus (S \cup S')}$
    if $\mathsf{Val}(f^{(a)}) \leqslant \mathsf{Val}(f^{(b)})$ **then**
        $f \leftarrow f^{(a)}$
    else
        $f \leftarrow f^{(b)}$
    end if
end while
return f (as f^*)

It's easy to see that Algorithm 1 always maintains a valid solution f to the LP (i. e., all variables y_i's are within the $[0, 1]$ range). Then we only need to prove the following two things to show the correctness of Algorithm 1: (1) the while loop terminates (in a linear number of steps), (2) in each loop, $\min\{\mathsf{Val}(f^{(a)}), \mathsf{Val}(f^{(b)})\} \leqslant \mathsf{Val}(f)$, so that $\mathsf{Val}(f)$ never increases in the whole algorithm.

To prove the first point, we consider the set

$$W_f = \{0 < y < 1/2 : \exists i \in V, s.t.\ y = y_i \vee y = 1 - y_i\}\,.$$

In each loop, the algorithm picks a p from W_f. At the end of the loop, we see that p is wiped from W_f while no new elements are added. Thus, after linear steps of the loop, W_f becomes $\emptyset$ and the loop terminates.

For the second point, we define

$$f^{(t)} = \{y_i^{(t)} = t\}_{i \in S} \cup \{y_i^{(t)} = 1 - t\}_{i \in S'} \cup \{y_i^{(t)} = y_i\}_{i \in V \setminus (S \cup S')}$$

for $t \in [a,b]$ at Line 9 in the algorithm. Then if we can show $\mathsf{Val}(f^{(t)})$ is a linear function within $t \in [a,b]$, together with the fact $p \in [a,b]$, we shall conclude that $\min\{\mathsf{Val}(f^{(a)}), \mathsf{Val}(f^{(b)})\} \leqslant \mathsf{Val}(f^{(p)}) = \mathsf{Val}(f)$. To prove the linearity of $\mathsf{Val}(f^{(t)})$, we only need to show that

$$g_1(t) = \max\{1 - y_i^{(t)} - y_j^{(t)}, 0\} \quad \text{and} \quad g_2(t) = \max\{y_i^{(t)} - y_j^{(t)}, 0\}$$

are linear with the respect to $t \in [a,b]$, for any possible i, j. Thus we discuss the following five cases.

- $i, j \in V \setminus (S \cup S')$. In this case, g_1 and g_2 are constant functions.
- $i \in V \setminus (S \cup S')$, $j \in S \cup S'$. In this case, the only "non-linear point" is at $t = 1 - y_i$ for g_1 and $t = y_i$ for g_2. But these two points are away from $[a,b]$.
- $i \in S \cup S'$, $j \in V \setminus (S \cup S')$. Similar argument works as the previous case.
- $i \in S$, $j \in S'$ (or $i \in S'$, $j \in S$). In this case, $1 - y_i^{(t)} - y_j^{(t)} = 0$ always holds for $t \in [a,b]$ and therefore g_1 is constant function. On the other hand, since $y_i^{(t)} \leqslant y_j^{(t)}$ (or $y_i^{(t)} \geqslant y_j^{(t)}$) , we also have $g_2(t) = 0$ (or $g_2(t) = y_i^{(t)} - y_j^{(t)} = 1 - 2t$) being linear.
- $i, j \in S$ (or $i, j \in S'$). In this case, $y_i^{(t)} = y_j^{(t)}$ always holds for $t \in [a,b]$ and therefore g_2 is constant function. On the other hand, since $y_i^{(t)} + y_j^{(t)} \leqslant 1$ (or $y_i^{(t)} + y_j^{(t)} \geqslant 1$), we also have $g_1(t) = 1 - y_i^{(t)} - y_j^{(t)} = 1 - 2t$ (or $g_1(t) = 0$) being linear.

$\square$

A direct corollary of Lemma 4.10 is the following.

Corollary 4.11. *There is a polynomial-time algorithm to get a solution f such that $\mathsf{Val}(f) = \mathrm{OPT}_{\mathrm{LP}}$ and the variables in f are half-integral (i. e., being one of 0, 1, and $1/2$).*

Now we are ready for the proof of Theorem 4.9.

Proof of Theorem 4.9. Apply Corollary 4.11 to get an optimal LP solution $f = \{y_i\}$ which has half-integral values. Then define $f_{\mathrm{int}} = \{x_i\}$ as follows. For each $i \in V$, let $x_i = 1$ when $y_i \geqslant 1/2$, and $x_i = 0$ when $y_i = 0$. We observe that

- $x_i \leqslant 2y_i$ and $1 - x_i \leqslant 1 - y_i$ for each $i \in V$.
- For each $i < j$, we have $\max\{1 - x_i - x_j, 0\} \leqslant \max\{1 - y_i - y_j, 0\}$ since $x_i \geqslant y_i$ and $x_j \geqslant y_j$.
- For each $i \neq j$, we see that when $\max\{y_i - y_j, 0\} = 0$ (in which case we have $y_i \leqslant y_j$), we always have $x_i \leqslant x_j$, therefore $\max\{x_i - x_j, 0\} = 0$. On the other hand, when $\max\{y_i - y_j, 0\}$ (in which case by half-integrality we have$\max\{y_i - y_j, 0\} \geqslant 1/2$), we have $\max\{x_i - x_j, 0\} \leqslant 1 \leqslant 2\max\{y_i - y_j, 0\}$.

Altogether, we have

$$\begin{aligned}\mathsf{Val}(f_{\text{int}}) &= \sum_{i\in V} w_i^{(T)}(1-x_i)+\sum_{i\in V} w_i^{(F)}x_i+\sum_{i<j} w_{ij}^{(D)}\max\{1-x_i-x_j,0\}+\sum_{i\neq j} w_{ij}^{(I)}\max\{x_i-x_j,0\}\\ &\leqslant \sum_{i\in V} w_i^{(T)}(1-y_i)+\sum_{i\in V} w_i^{(F)}2y_i+\sum_{i<j} w_{ij}^{(D)}\max\{1-y_i-y_j,0\}+\sum_{i\neq j} w_{ij}^{(I)}2\max\{y_i-y_j,0\}\\ &\leqslant 2\mathsf{Val}(f)=2\mathrm{OPT}_{\mathrm{LP}}\leqslant 2\mathrm{OPT}\,.\end{aligned}$$

□

5 Inapproximability and approximation algorithm for Max 1-in-k-HS

5.1 SDP gap and UG-hardness for Max 1-in-k-HS

In this section, we construct an SDP gap for Max 1-in-k-HS. Recall that the problem is to find a subset that intersects a maximum number of sets (of size at most k) from an input family at exactly one element. We are going to prove Theorem 2.4, which is restated as follows.

Theorem 2.4 (restated). *For some absolute constant $C'>0$, for every $\alpha>0$, given a $(1-1/k^{1-\alpha})$-satisfiable instance of Max 1-in-k-HS, it is UG-hard to find a subset intersecting more than a fraction $C'/(\alpha\log k)$ of the sets exactly once.*

We start by constructing the gap instance. Roughly speaking, the instance includes all subsets of the universe with size at most k and we put uniform weights on sets of geometrically varying sizes. Here is the construction in details.

Instance We define the (weighted) instance of Max 1-in-k-HS, denoted $\mathcal{I}^{\mathrm{EHS}}(m,n,\varepsilon)$, parameter $0<\varepsilon<1$, $m\geqslant 2$ and $n\geqslant \varepsilon m\cdot 2^{2\lceil m(1+\varepsilon)\rceil}$ as follows:

- The universe $U=[n]=\{1,2,\ldots,n\}$.
- Let the weight of a set $S\in\mathcal{C}$ be the probability under the following measure: choose $t\in m+1,m+2,\ldots,\lceil m(1+\varepsilon)\rceil$ uniformly at random, and output a uniformly random subset of U of size 2^t.

Note that in such an instance, the size of S_i is at most $k=2^{\lceil m(1+\varepsilon)\rceil}$.

5.1.1 Upper bound of optimal integral solution

In this section, we prove the following lemma showing that the above instance does not have a good exact hitting set.

Lemma 5.1. *There is a constant C_1 such that for all $0<\varepsilon<1$, $m\geqslant 2$ and $n\geqslant \varepsilon m\cdot 2^{2\lceil m(1+\varepsilon)\rceil}$, the optimal solution to $\mathcal{I}^{\mathrm{EHS}}(m,n,\varepsilon)$ has value at most $C_1/(\varepsilon\log k)$.*

We begin with the following two statements that will be useful in bounding the value of any integral solution to $\mathcal{I}^{\mathrm{EHS}}(m,n,\varepsilon)$.

Lemma 5.2. *Suppose the hitting set $V \subseteq U$ is of size ℓ. Then the probability that a size-z ($2 \leqslant z \leqslant \ell/2$) set is hit exactly once by V, is at most $(2z/n)\cdot\ell\cdot(1/\mathrm{e})^{z\ell/4n}$.*

Proof.

$$\begin{aligned}
\mathbf{Pr}_{S\in\mathcal{C}}[|S\cap V|=1 \mid |S|=z] &= \frac{\ell\binom{n-\ell}{z-1}}{\binom{n}{z}} \\
&= \ell\cdot\frac{(n-\ell)!(n-z)!z!}{(n-\ell-z+1)!(z-1)!n!} \\
&= z\ell\cdot\frac{(n-\ell)!}{(n-\ell-z+1)!}\cdot\frac{(n-z)!}{n!} \\
&\leqslant z\ell\cdot\frac{(n-\ell)^{z-1}}{(n-z)^{z}} \\
&= \frac{z}{n-z}\cdot\ell\cdot\left(1-\frac{\ell-z}{n-z}\right)^{z-1} \\
&\leqslant \frac{z}{n-z}\cdot\ell\cdot\left(\frac{1}{\mathrm{e}}\right)^{(z-1)(\ell-z)/(n-z)} \\
&\leqslant \frac{z}{n-z}\cdot\ell\cdot\left(\frac{1}{\mathrm{e}}\right)^{z\ell/4(n-z)} && (2\leqslant z\leqslant \ell/2) \\
&\leqslant \frac{2z}{n}\cdot\ell\cdot\left(\frac{1}{\mathrm{e}}\right)^{z\ell/4n} && (z\leqslant \ell/2\leqslant n/2)\,.
\end{aligned}$$

□

Claim 5.3. *For all $x>0$ and $m\in\mathbb{N}^+$, we have*

$$\sum_{i=1}^{m} 2^i x\mathrm{e}^{-2^i x} \leqslant 2/\ln 2.$$

Proof.

$$\begin{aligned}
\sum_{i=1}^{m} 2^i x\mathrm{e}^{-2^i x} &\leqslant \sum_{i=-\infty}^{+\infty} 2^i x\mathrm{e}^{-2^i x} \\
&= \sum_{i=-\infty}^{[\log_2 1/x]} 2^i x\mathrm{e}^{-2^i x} + \sum_{i=[\log_2 1/x]+1}^{+\infty} 2^i x\mathrm{e}^{-2^i x} \\
&\leqslant \int_{-\infty}^{[\log_2 1/x]+1} 2^y x\mathrm{e}^{-2^y x}dy + \int_{[\log_2 1/x]}^{+\infty} 2^y x\mathrm{e}^{-2^y x}dy && \text{(monotonicity)} \\
&\leqslant 2\int_{-\infty}^{+\infty} 2^y x\mathrm{e}^{-2^y x}dy \\
&= \frac{2}{\ln 2}\,.
\end{aligned}$$

□

We can now prove Lemma 5.1.

Proof of Lemma 5.1. Set $C_1 = \max\{32/\ln 2, 12\}$. Given a solution V, let $\ell = |V|$. If $\ell \geqslant 2 \cdot 2^{\lceil m(1+\varepsilon) \rceil}$, then $\ell \geqslant 2|S|, \forall S \in \mathcal{C}$. In this case, the probability that $S \in \mathcal{C}$ is hit exactly once, is

$$\begin{aligned}
\mathbf{Pr}_{S\in\mathcal{C}}[|S\cap V| = 1] &= \sum_{t=m+1}^{\lceil m(1+\varepsilon)\rceil} \mathbf{Pr}_{S\in\mathcal{C}}[|S| = 2^t] \cdot \mathbf{Pr}_{S\in\mathcal{C}}[|S\cap V| = 1 \bigm| |S| = 2^t] \\
&= \frac{1}{\varepsilon m} \sum_{t=m+1}^{\lceil m(1+\varepsilon)\rceil} \mathbf{Pr}_{S\in\mathcal{C}}[|S\cap V| = 1 \bigm| |S| = 2^t] \\
&\leqslant \frac{1}{\varepsilon m} \sum_{t=m+1}^{\lceil m(1+\varepsilon)\rceil} \frac{2\cdot 2^t}{n} \cdot \ell \cdot \left(\frac{1}{\mathrm{e}}\right)^{2^t \ell/4n} && \text{(by Lemma 5.2)} \\
&\leqslant \frac{1}{\varepsilon m} \cdot \frac{16}{\ln 2} && \text{(by Claim 5.3)} \\
&\leqslant C_1/(\varepsilon \log k).
\end{aligned}$$

On the other hand, if $\ell < 2 \cdot 2^{\lceil m(1+\varepsilon) \rceil}$, then

$$\begin{aligned}
&\mathbf{Pr}_{S\in\mathcal{C}}[|S\cap V| = 1] \\
\leqslant\ &\mathbf{Pr}_{S\in\mathcal{C}}[|S\cap V| \geqslant 1] \\
\leqslant\ &\mathbf{Pr}_{S\in\mathcal{C}}\left[|S\cap V| \geqslant 1 \,\middle|\, |S| = 2^{\lceil m(1+\varepsilon)\rceil}\right] && \text{(since the sets in } \mathcal{C} \text{ have size at most } 2^{\lceil m(1+\varepsilon)\rceil}) \\
=\ &1 - \binom{n-\ell}{2^{\lceil m(1+\varepsilon)\rceil}} \Big/ \binom{n}{2^{\lceil m(1+\varepsilon)\rceil}} \\
=\ &1 - \frac{(n-\ell)!}{(n-\ell-2^{\lceil m(1+\varepsilon)\rceil})!} \cdot \frac{(n-2^{\lceil m(1+\varepsilon)\rceil})!}{n!} \\
\leqslant\ &1 - \left(\frac{n-\ell-2^{\lceil m(1+\varepsilon)\rceil}}{n}\right)^{\ell} \\
\leqslant\ &1 - \left(\frac{n-3\cdot 2^{\lceil m(1+\varepsilon)\rceil}}{n}\right)^{2\cdot 2^{\lceil m(1+\varepsilon)\rceil}} && (\ell < 2\cdot 2^{\lceil m(1+\varepsilon)\rceil}) \\
\leqslant\ &\frac{6\cdot 2^{2\lceil m(1+\varepsilon)\rceil}}{n} && (\forall 0 \leqslant x \leqslant 1, y \geqslant 0, (1-x)^y \geqslant 1-xy) \\
\leqslant\ &6/\varepsilon m && (n \geqslant \varepsilon m \cdot 2^{2\lceil m(1+\varepsilon)\rceil}) \\
\leqslant\ &C_1/(\varepsilon \log k)\,.
\end{aligned}$$

And this proves the lemma. □

5.1.2 Construction of good SDP solution

We prove that the canonical SDP has a solution with value close to 1.

Lemma 5.4. *For the* Max 1*-in-*k*-HS instance* $\mathcal{I}^{\mathrm{EHS}}(m,n,\varepsilon)$*, the optimal solution to the canonical SDP has value at least* $1-4/2^m \geqslant 1-4/k^{1-\varepsilon}$.

To prove Lemma 5.4, recall the canonical SDP for Max 1-in-k-HS as follows:

$$\begin{array}{lrll}
\text{Maximize} & \mathbf{E}_{S\in\mathcal{C}}[\mathbf{Pr}_{\sigma\in\pi_S}[|\sigma^{-1}(1)|=1]] & & \\
\text{subject to} & \boldsymbol{v}_i\cdot\boldsymbol{I} & = \|\boldsymbol{v}_i\|^2 & \forall i,j\in U\,, \\
 & \|\boldsymbol{I}\|^2 & = 1 & \forall i\in U\,, \\
 & \mathbf{Pr}_{\sigma\in\pi_S}[\sigma(i)=1] & = \|\boldsymbol{v}_i\|^2 & \forall S\in\mathcal{C}, i\in S\,, \\
 & \mathbf{Pr}_{\sigma\in\pi_S}[\sigma(i)=1\wedge\sigma(j)=1] & = \boldsymbol{v}_i\cdot\boldsymbol{v}_j & \forall S\in\mathcal{C}, i\neq j\in S\,.
\end{array}$$

Here are some high level ideas about the our construction of the SDP solution. To make the consistency between local distributions, we need to find local distributions where every element has the same marginal probability p and every pair of elements has the same marginal probability q. This is relatively simple because of the symmetry of the sets in $\mathcal{C}$. We can define the local distribution in a symmetric way, where all the assignments choosing the same number of elements have the same probability mass. The goal is to find the local distributions that have a lot of mass on singletons (the assignments choosing exactly 1 element). Ideally we want to put all the mass on singletons, but this would violate the consistency check and positive semidefiniteness. We can resolve this issue by assigning a small fraction of probability mass to the assignments choosing all the elements or choosing half of the elements, and giving more flexibility to our construction.

Now, we exhibit an SDP solution for the instance $\mathcal{I}^{\mathrm{EHS}}(m,n,\varepsilon)$ that has value close to 1. We first construct the scalars, and then the vectors.

Constructing the solution—scalars Let $M=2^m$, $p=2/M$, $q=1/M$. We would like to define local probability distributions so that p is the marginal probability for a singleton being chosen, and q is the marginal probability for a pair of distinct elements being chosen. For each $S\in\mathcal{C}$, and each $\sigma: S\to\{0,1\}$, define the local distribution π_S as follows:

$$\pi_S(\sigma)=\begin{cases}
\frac{|S|}{|S|-2}\cdot\left(\frac{|S|}{|S|-1}-\frac{3|S|-2}{|S|-1}\cdot p+2q\right)\Big/\binom{|S|}{1} & \text{if } |\sigma^{-1}(1)|=1, \\
\frac{4}{|S|-2}\cdot\Big((|S|-1)(p-q)-(1-p)\Big)\Big/\binom{|S|}{|S|/2} & \text{if } |\sigma^{-1}(1)|=\frac{|S|}{2}, \\
1-\binom{|S|}{1}\pi_S(\sigma)|_{|\sigma|=1}-\binom{|S|}{|S|/2}\pi_S(\sigma)|_{|\sigma^{-1}(1)|=|S|/2} & \\
\qquad=\frac{1}{|S|-1}-\frac{|S|}{|S|-1}\cdot p+2q & \text{if } |\sigma^{-1}(1)|=|S|, \\
0 & \text{otherwise.}
\end{cases}$$

Given $M<|S|$ for all $S\in\mathcal{C}$, it is easy to check π_S is always non-negative. And it can be checked that $\sum_{\sigma\subseteq S}\pi_S(\sigma)=1$. Thus, π_S is a valid probability distribution.

Then we calculate the following values which are related to the SDP.

- For all $i \in S \in \mathcal{C}$,

$$\begin{aligned}\mathbf{Pr}_{\sigma\in\pi_S}[\sigma(i)=1] &= 1-\frac{|S|-1}{|S|}\cdot\frac{|S|}{|S|-2}\cdot\Big(\frac{|S|}{|S|-1}-\frac{3|S|-2}{|S|-1}\cdot p+2q\Big)\\ &\quad -\frac{1}{2}\cdot\frac{4}{|S|-2}\cdot\Big((|S|-1)(p-q)-(1-p)\Big)\\ &= p\,.\end{aligned}$$

- For all $i \neq j \in S \in \mathcal{C}$,

$$\begin{aligned}\mathbf{Pr}_{\sigma\in\pi_S}[\sigma(i)=1\wedge\sigma(j)=1] &= \Big(\frac{1}{|S|-1}-\frac{|S|}{|S|-1}\cdot p+2q\Big)\\ &\quad +\frac{|S|/2-1}{2(|S|-1)}\cdot\frac{4}{|S|-2}\cdot\Big((|S|-1)(p-q)-(1-p)\Big)\\ &= q\,.\end{aligned}$$

Constructing the solution—vectors Now we need to show there exists a set of vectors passing the consistency check on local distributions we defined above. In fact, we show there exists set of vectors satisfying even stricter requirements, where the inner-product between every pair of vectors is defined, as follows:

$$\begin{aligned}&\|\boldsymbol{v}_i\|^2 = p && \forall i\in U\,,\\ &\boldsymbol{v}_i\cdot\boldsymbol{v}_j = q && \forall i\neq j\in U\,,\\ &\boldsymbol{v}_i\cdot\boldsymbol{I} = \|\boldsymbol{v}_i\|^2 && \forall i\in U\,,\\ &\|\boldsymbol{I}\|^2 = 1\,.\end{aligned}$$

Thus we only need to show the corresponding inner-product matrix is positive semidefinite. The matrix is in the form of

$$A=\begin{bmatrix} 1 & p\boldsymbol{b}^T \\ p\boldsymbol{b} & (p-q)I+qJ\end{bmatrix}$$

where $\boldsymbol{b}$ is $n\times 1$ all-one vector, J is the $n\times n$ all-one matrix, and I is the identity matrix.

Given $\boldsymbol{x}=(x_0,x_1,\dots,x_n)\in\mathbb{R}^n$,

$$\begin{aligned}\boldsymbol{x}^TA\boldsymbol{x} &= (x_0,x_1,\dots,x_n)\begin{bmatrix} 1 & p\boldsymbol{b}^T \\ p\boldsymbol{b} & (p-q)I+qJ\end{bmatrix}(x_0,x_1,\dots,x_n)^T\\ &= x_0^2+2px_0(\sum_{i=1}^n x_i)+q(\sum_{i=1}^n x_i)^2+(p-q)\sum_{i=1}^n x_i^2\,.\end{aligned}$$

Note that this quadratic form is always non-negative when $p\geqslant q$ and $4p^2-4q\leqslant 0$ (which is equivalent to $q\geqslant p^2$). Our $p=2/M$ and $q=1/M$ satisfies these conditions. Therefore the inner-product matrix is positive semidefinite and the vectors exist.

Now we can prove Lemma 5.4, which says the optimal SDP solution has value close to 1.

Proof of Lemma 5.4. The value of the solution we exhibited above is

$$\begin{aligned}
\mathbf{E}_{S\in\mathcal{C}}[\mathbf{Pr}_{\sigma\in\pi_S}[|\sigma^{-1}(1)|=1]] &= \mathbf{E}_{S\in\mathcal{C}}\Big[\sum_{\sigma:S\to\{0,1\},|\sigma|^{-1}(1)=1}\pi_S(\sigma)\Big] \\
&= \mathbf{E}_{S\in\mathcal{C}}\Big[\frac{|S|}{|S|-2}\cdot\Big(\frac{|S|}{|S|-1}-\frac{3|S|-2}{|S|-1}\cdot p+2q\Big)\Big] \\
&\geqslant \mathbf{E}_{S\in\mathcal{C}}\Big[\frac{|S|}{|S|-1}-\frac{3|S|-2}{|S|-1}\cdot p+2q\Big] \\
&= \mathbf{E}_{S\in\mathcal{C}}[1-3p+2q+(1-p)/(|S|-1)] \\
&\geqslant \mathbf{E}_{S\in\mathcal{C}}[1-3p+2q] \\
&= 1-3p+2q = 1-4/M\,.
\end{aligned}$$

□

Together with Theorem 3.1, Lemmas 5.1 and 5.4 imply Theorem 2.4. □

5.2 A robust algorithm for almost-satisfiable Max 1-in-k-SAT

In this section, we prove Theorem 2.5.

Theorem 2.5 (restated). *For every constant $B>1$, the following holds. There is a polynomial time algorithm that given a $(1-1/(Bk))$-satisfiable instance of Max 1-in-k-SAT, finds a truth-assignment on variables satisfying exactly one term for a fraction λ of the clauses, where $\lambda=(1-1/\sqrt{B})^2/\mathrm{e}$.*

The algorithm is based on rounding an LP relaxation for the problem, and gives a robust version of the algorithm in [8] which achieved a factor $1/\mathrm{e}$-approximation for (perfectly) satisfiable instances.

Given a truth-assignment σ and a clause C, we denote $\sigma\cap C$ by the set of terms in C satisfied by σ. Our algorithm first solves the following LP relaxation of the problem. (The relaxation is a linear programming since we can expand the expectation and probability operators into linear combinations of entries of π_C.)

$$\begin{array}{lll}
\text{Maximize} & \mathbf{E}_{C\in\mathcal{C}}[\mathbf{Pr}_{\sigma\in\pi_C}[|\sigma\cap C|=1]] & \\
\text{subject to} & \mathbf{Pr}_{\sigma\in\pi_C}[\sigma(i)=1]=x_i & \forall C\in\mathcal{C}, i\in C\,.
\end{array}$$

Given a solution $\{\pi_C\}$ and $\{x_i\}$ to the LP, we generate an assignment τ by for each $i\in U$ letting $\tau(x_i)=1$ with probability x_i. Then we prove the following lemma which directly implies Theorem 2.5.

Lemma 5.5. *For every constant $B>1$, when $\mathrm{OPT}_{\mathrm{LP}}>1-1/(Bk)$, we have*

$$\mathbf{E}_\tau[\mathbf{Pr}_{C\in\mathcal{C}}[|\tau\cap C|=1]]\geqslant\frac{(1-1/\sqrt{B})^2}{\mathrm{e}}\,.$$

Proof. Given $\mathbf{E}_{C\in\mathcal{C}}[\mathbf{Pr}_{\sigma\in\pi_C}[|\sigma\cap C|=1]]>1-1/(Bk)$, by an averaging argument, we know that for at least $(1-1/\sqrt{B})$ fraction of $C\in\mathcal{C}$ are "good," i. e., for these C clauses, we have

$$\mathbf{Pr}_{\sigma\in\pi_C}[|\sigma\cap C|=1]>1-\frac{1}{\sqrt{B}k}.$$

For each good $C\in\mathcal{C}$, and for each term $t\in C$, let $p(t)=x_i$ if $t=x_i$, or $p(t)=1-x_i$ if $t=\overline{x_i}$, i. e., $p(t)$ is the probability that t is satisfied by τ. Then we know that

$$\sum_{t\in C}p(t)=\mathbf{E}_{\sigma\in\pi_C}[|\sigma\cap C|]\geqslant\mathbf{Pr}_{\sigma\in\pi_C}[|\sigma\cap C|=1]\geqslant 1-\frac{1}{\sqrt{B}k}.$$

On the other hand,

$$\sum_{t\in C}p(t)=\mathbf{E}_{\sigma\in\pi_C}[|\sigma\cap C|]\leqslant\mathbf{Pr}_{\sigma\in\pi_C}[|\sigma\cap C|=1]+(1-\mathbf{Pr}_{\sigma\in\pi_C}[|\sigma\cap C|=1])|C|\leqslant 1+1/\sqrt{B}. \quad (5.1)$$

We now lower bound the probability that τ satisfies C, using the Lemma 5.6 proved at the end of the section. We discuss the following two cases to establish the lower bound.

Case 1. If all the terms in C depend on distinct variables, then

$$\mathbf{Pr}_\tau[|\tau\cap C|=1]=\sum_{t\in C}p(t)\prod_{t'\in C,t\neq t'}(1-p(t')). \quad (5.2)$$

For good C we know that $\sum_{t\in C}p(t)\in[1-1/(\sqrt{B}k),1+1/\sqrt{B}]\subseteq[1-1/\sqrt{B},1+1/\sqrt{B}]$. By Lemma 5.6 given right after this proof, we know that (5.2) $\geqslant(1-1/\sqrt{B})/\mathrm{e}$.

Case 2. If some terms in C depend on the same variable, i. e., $\exists i: x_i,\overline{x_i}\in C$, then by (5.1) we know that $\sum_{t\in C\setminus\{x_i,\overline{x_i}\}}p(t)\leqslant 1/\sqrt{B}<1$. Thus terms in $C\setminus\{x_i,\overline{x_i}\}$ depend on distinct variables, we have

$$\mathbf{Pr}_\tau[|\tau\cap C|=1]=1\cdot\prod_{t\in C\setminus\{x_i,\overline{x_i}\}}(1-p(t))\geqslant 1-\sum_{t\in C\setminus\{x_i,\overline{x_i}\}}p(t)\geqslant 1-1/\sqrt{B}.$$

Combining the two cases above, we get

$$\begin{aligned}\mathbf{E}_\tau[\mathbf{Pr}_{C\in\mathcal{C}}[|\tau\cap C|=1]] &= \mathbf{E}_{C\in\mathcal{C}}[\mathbf{Pr}_\tau[|\tau\cap C|=1]]\\ &\geqslant (1-1/\sqrt{B})\mathbf{E}_{C\in\mathcal{C}}[\mathbf{Pr}_\tau[|\tau\cap C|=1]\mid C\text{ is good}]\\ &\geqslant \frac{(1-1/\sqrt{B})^2}{\mathrm{e}}.\end{aligned}$$

□

It remains to prove the following inequality which was used in the above proof.

Lemma 5.6. *Given $x_1,x_2,\ldots,x_n\in[0,1]$, and $1-\varepsilon\leqslant\sum_i x_i\leqslant 1+\varepsilon$ where $\varepsilon<1$ then*

$$\sum_i x_i\prod_{j\neq i}(1-x_j)\geqslant\frac{1-\varepsilon}{\mathrm{e}}.$$

Proof. We use the following claim to prove this lemma.

Claim 5.7. *For $n \geqslant 2$, given a set of n numbers $\{x_i\}$ as described in the lemma, the objective function $\sum_i x_i \prod_{j \neq i}(1-x_j)$ is minimized when*

- *all the x_i's are the same, or*
- *$\exists i : x_i = 0$ or $\exists i : x_i = 1$.*

Proof. Suppose the first condition doesn't hold, we prove the second one holds. Without loss of generality assume that $x_1 \neq x_2$. Then rewrite the objective function as

$$\sum_i x_i \prod_{j \neq i}(1-x_j) = (1-x_1)(1-x_2)\Big(\sum_{i \geqslant 3} x_i \prod_{j \neq i, j \geqslant 3}(1-x_j)\Big) + \Big(x_1(1-x_2)+x_2(1-x_1)\Big)\prod_{j \geqslant 3}(1-x_j)\,.$$

Let $C_1 = \sum_{i \geqslant 3} x_i \prod_{j \neq i, j \geqslant 3}(1-x_j)$ and $C_2 = \prod_{j \geqslant 3}(1-x_j)$, we have

$$\sum_i x_i \prod_{j \neq i}(1-x_j) \quad = \quad C_1 + (C_2 - C_1)(x_1+x_2) + (C_1 - 2C_2)x_1x_2\,.$$

Note that when fixing the sum x_1+x_2, we can change individual values of x_1 and x_2 within $[0,1]$ while still $\{x_i\}$ still being a valid solution. By the perturbation, only the term $(C_1-2C_2)x_1x_2$ in the objective function might have value changed. Since $x_1 \neq x_2$, we know that $C_1 - 2C_2 > 0$ or making $x_1' = x_2' = (x_1+x_2)/2$ gets no larger objective function value. When $C_1 - 2C_2 > 0$, x_1 and x_2 should be "apart from" each other, thus one of x_1 and x_2 must touch their bound, i. e., 0 or 1. □

Now we use this claim and induction on n to prove the lemma. The lemma trivially holds in the base case when $n = 1$. When $n = k > 1$, supposing the lemma holds for all $n < k$, we discuss the three cases proposed by Claim 5.7 (splitting the second case in the claim into two).

- When all x_i's are the same, we know that $x_i = S/n$ where $S = \sum_i x_i$. Then

  $$\sum_i x_i \prod_{j \neq i}(1-x_j) = S\Big(1-\frac{S}{n}\Big)^{n-1} \geqslant S\mathrm{e}^{-S} \geqslant \min\{(1-\varepsilon)\mathrm{e}^{\varepsilon}, (1+\varepsilon)\mathrm{e}^{-\varepsilon}\}/e\,,$$

  where the last inequality is because that $S \in [1-\varepsilon, 1+\varepsilon]$ and that $S\mathrm{e}^{-S}$ is monotonically increasing when $S \in [1-\varepsilon, 1]$ and monotonically decreasing when $S \in [1, 1+\varepsilon]$.

  We have $(1-\varepsilon)\mathrm{e}^{\varepsilon} \geqslant 1-\varepsilon$. By $\mathrm{e}^{-\varepsilon} \geqslant 1-\varepsilon$, we know that $(1+\varepsilon)\mathrm{e}^{-\varepsilon} \geqslant 1-\varepsilon^2 \geqslant 1-\varepsilon$. In all, we have $\sum_i x_i \prod_{j \neq i}(1-x_j) \geqslant (1-\varepsilon)/e$.

- When $\exists i : x_i = 0$, with out loss of generality, suppose $x_1 = 0$. Then this reduces to the same problem with $(n-1)$ variables and the induction hypothesis gives us a $(1-\varepsilon)/\mathrm{e}$ lower bound.

- When $\exists i : x_i = 1$, again with out loss of generality, suppose $x_1 = 1$. Now the objective function becomes $\prod_{i \geqslant 2}(1-x_i)$ while $\sum_{i \geqslant 2} x_i$ is at most ε. It is easy to see the product is lower bounded by $(1-\varepsilon)$. (All but one of x_i are 0.)

□

6 Concluding remarks on finding almost-satisfying assignments for CSPs

In the world of "CSP dichotomy" (see [10] for a recent survey), the tractability of LIN-mod-2, 2-SAT, and Horn-SAT is explained by the existence of non-trivial *polymorphisms* which combine many satisfying assignments to produce a new satisfying assignment. The Boolean functions which are polymorphisms for LIN-mod-2, 2-SAT, and Horn-SAT are xor (of odd size), majority, and minimum respectively. The existence of algorithms to find almost-satisfying assignments to 2-SAT and Horn-SAT can be attributed to the "noise stability" of the majority and minimum functions. The xor function of many variables, on the other hand, is highly sensitive to noise. This distinction seems to underly the difficulty of solving near-satisfiable instances of LIN-mod-2 and Håstad's tight hardness result for the problem.

For Boolean CSPs, we understand the complexity of finding almost-satisfying assignments for all the cases where deciding satisfiability is tractable: it is possible in polynomial time for 2-SAT and Horn-SAT, and NP-hard for LIN-mod-2. Further, under the UGC, the exact approximation threshold as a function of the gap ε to perfect satisfiability is also pinned down for both 2-SAT and Horn-SAT. What about CSPs over larger domains? For any CSP Π that can "express linear equations" (this notion is formalized in the CSP dichotomy literature, but we can work with the intuitive meaning for this discussion), Håstad's strong inapproximability result for near-satisfiable linear equations over abelian groups [9] implies hardness of finding an almost satisfying assignment for $(1-\varepsilon)$-satisfiable instances of Π. A recent breakthrough [1] established that every other tractable CSP (i. e., a polynomial time decidable CSP that cannot express linear equations) must be of so-called "bounded width," which means that a natural *local* propagation algorithm correctly decides satisfiability of every instance of that CSP.

We end this paper with the appealing conjecture that every bounded width CSP admits a robust satisfiability algorithm that can find a $(1-g(\varepsilon))$-satisfying assignment given a $(1-\varepsilon)$-satisfiable instance for some function $g()$ such that $g(\varepsilon) \to 0$ as $\varepsilon \to 0$. (We should clarify that in this context we always treat the domain size k of the CSP as fixed and let $\varepsilon \to 0$, so $g(\varepsilon)$ can have an arbitrary dependence on k. Note that Unique Games itself, which is a bounded width CSP, admits a robust satisfiability algorithm that satisfies a fraction $1-O(\sqrt{\varepsilon \log k})$ of constraints in a $(1-\varepsilon)$-satisfiable instance [3].) By the preceding discussion, this conjecture would imply that bounded width *characterizes* the existence of robust satisfiability algorithms for CSPs.

6.1 Subsequent work on our conjecture

There are several subsequent works regarding the conjecture proposed above, after the publication of the conference version of this paper. A partial answer to the conjecture for width-1 problems was independently given by [16] and [6], and the full conjecture was later confirmed by [2].

Acknowledgments

We thank Andrei Krokhin for a useful discussion on the relationship between the Unique Games conjecture and our "bounded width implies robust satisfiability" conjecture.

References

[1] LIBOR BARTO AND MARCIN KOZIK: Constraint satisfaction problems of bounded width. In *Proc. 50th FOCS*, pp. 595–603. IEEE Comp. Soc. Press, October 2009. [doi:10.1109/FOCS.2009.32] 264

[2] LIBOR BARTO AND MARCIN KOZIK: Robust satisfiability of constraint satisfaction problems. In *Proc. 44th STOC*, pp. 931–940. ACM Press, 2012. [doi:10.1145/2213977.2214061] 264

[3] MOSES CHARIKAR, KONSTANTIN MAKARYCHEV, AND YURY MAKARYCHEV: Near-optimal algorithms for unique games. In *Proc. 38th STOC*, pp. 205–214. ACM Press, 2006. [doi:10.1145/1132516.1132547] 264

[4] MOSES CHARIKAR, KONSTANTIN MAKARYCHEV, AND YURY MAKARYCHEV: Near-optimal algorithms for maximum constraints satisfaction problems. *ACM Trans. Algorithms*, 5(3), 2009. Preliminary version in SODA'07. [doi:10.1145/1541885.1541893] 241

[5] PIERLUIGI CRESCENZI, RICCARDO SILVESTRI, AND LUCA TREVISAN: On weighted vs unweighted versions of combinatorial optimization problems. *Inform. and Comput.*, 167(1):10–26, 2001. Preliminary version in ISTCS'96. [doi:10.1006/inco.2000.3011] 240

[6] VICTOR DALMAU AND ANDREI KROKHIN: Robust satisfiability for CSPs: Algorithmic and hardness results, 2011. In preparation. 264

[7] ERIK D. DEMAINE, URIEL FEIGE, MOHAMMAD TAGHI HAJIAGHAYI, AND MOHAMMAD R. SALAVATIPOUR: Combination can be hard: Approximability of the unique coverage problem. *SIAM J. Comput.*, 38(4):1464–1483, 2008. Preliminary version in SODA'06. [doi:10.1137/060656048] 243

[8] VENKATESAN GURUSWAMI AND LUCA TREVISAN: The complexity of making unique choices: Approximating 1-in-k sat. In *Proc. 8th Internat. Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX'05)*, pp. 99–110, 2005. [doi:10.1007/11538462_9] 241, 243, 247, 261

[9] JOHAN HÅSTAD: Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001. Preliminary version in STOC'97. [doi:10.1145/502090.502098] 241, 264

[10] PAVOL HELL AND JAROSLAV NEŠETŘIL: Colouring, constraint satisfaction, and complexity. *Comp. Sci. Rev.*, 2(3):143–163, 2008. [doi:10.1016/j.cosrev.2008.10.003] 264

[11] PETER JONSSON, ANDREI KROKHIN, AND FREDRIK KUIVINEN: Hard constraint satisfaction problems have hard gaps at location 1. *Theoret. Comput. Sci.*, 410(38-40):3856–3874, 2009. [doi:j.tcs.2009.05.022] 241

[12] SANJEEV KHANNA, MADHU SUDAN, LUCA TREVISAN, AND DAVID P. WILLIAMSON: The approximability of constraint satisfaction problems. *SIAM J. Comput.*, 30(6):1863–1920, 2001. [doi:10.1137/S0097539799349948] 240, 241, 242, 252

[13] SUBHASH KHOT: On the power of unique 2-prover 1-round games. In *Proc. 34th STOC*, pp. 767–775. ACM Press, 2002. [doi:10.1145/509907.510017] 241

[14] SUBHASH KHOT, GUY KINDLER, ELCHANAN MOSSEL, AND RYAN O'DONNELL: Optimal inapproximability results for MAX-CUT and other 2-variable CSPs? *SIAM J. Comput.*, 37(1):319–357, 2007. Preliminary version in FOCS'04. [doi:10.1137/S0097539705447372] 241

[15] SUBHASH KHOT AND ODED REGEV: Vertex cover might be hard to approximate to within 2-ε. *J. Comput. System Sci.*, 74(3):335–349, 2008. Preliminary version in CCC'03. [doi:10.1016/j.jcss.2007.06.019] 253

[16] GÁBOR KUN, RYAN O'DONNELL, SUGURU TAMAKI, YUICHI YOSHIDA, AND YUAN ZHOU: Linear programming, width-1 CSPs, and robust satisfaction. In *Proc. 3rd Innovations in Theoretical Computer Science Conf. (ITCS'12)*, pp. 484–495, New York, NY, USA, 2012. ACM Press. [doi:10.1145/2090236.2090274] 264

[17] PRASAD RAGHAVENDRA: Optimal algorithms and inapproximability results for every CSP? In *Proc. 40th STOC*, pp. 245–254. ACM Press, 2008. [doi:10.1145/1374376.1374414] 244, 245

[18] THOMAS J. SCHAEFER: The complexity of satisfiability problems. In *Proc. 10th STOC*, pp. 216–226. ACM Press, 1978. [doi:10.1145/800133.804350] 240

[19] URI ZWICK: Finding almost-satisfying assignments. In *Proc. 30th STOC*, pp. 551–560. ACM Press, 1998. [doi:10.1145/276698.276869] 240, 241, 245

AUTHORS

Venkatesan Guruswami
Associate professor
Carnegie Mellon University, Pittsburgh, PA
guruswami@cmu.edu
http://www.cs.cmu.edu/~venkatg/

Yuan Zhou
Ph. D. student
Carnegie Mellon University, Pittsburgh, PA
yuanzhou@cs.cmu.edu
http://www.cs.cmu.edu/~yuanzhou/

ABOUT THE AUTHORS

VENKATESAN GURUSWAMI, or Venkat, as his friends and colleagues call him, received his Bachelor's degree from the Indian Institute of Technology at Madras in 1997. He is grateful to his undergraduate research advisor, C. Pandu Rangan, for initiating him to the joys of theoretical computer science. Venkat received his Ph. D. from the Massachusetts Institute of Technology in 2001 under the excellent advisorship of Madhu Sudan.

Venkat is currently an Associate Professor in the Computer Science Department at Carnegie Mellon University. Earlier, during 2002-09, he was a faculty member at the University of Washington. Venkat was a Miller Research Fellow at UC Berkeley during 2001-02, and was a member of the School of Mathematics, Institute for Advanced Study during 2007-08.

Venkat's research interests span several topics including the theory of error-correcting codes, approximability of fundamental optimization problems, pseudorandomness, probabilistically checkable proofs, computational complexity theory, and algebraic algorithms.

Venkat currently serves on the editorial boards of the SIAM Journal on Computing, the IEEE Transactions on Information Theory, and the ACM Transactions on Computation Theory, and is program committee chair for the 2012 Computational Complexity conference. Venkat is a recipient of the Presburger award, the Packard Fellowship, the Sloan Fellowship, the NSF CAREER award, the ACM Doctoral Dissertation Award, and the IEEE Information Theory Society Paper Award.

In his (sadly limited) spare time, Venkat enjoys traveling, ethnic vegetarian food, racquet sports, hiking/running within his humble limits, and listening to Carnatic music.

YUAN ZHOU is a Ph. D. student in the Computer Science Department at Carnegie Mellon University. His advisors are Ryan O'Donnell and Venkatesan Guruswami. Yuan was an undergraduate student in the Microsoft Special Pilot CS Class supervised by Prof. Andrew Yao at Tsinghua University, and received his Bachelor's degree in 2009. His main research interests are in approximation algorithms, hardness of approximation, and analysis of Boolean functions. He is also interested in algorithmic mechanism design, algebraic dichotomy theory, coding theory, and their relations to the theory of approximation algorithms.