Optimal Streaming Approximations for all Boolean Max-2CSPs and Max-kSAT

Chi-Ning Chou
John A. Paulson School of Engineering and Applied Sciences
Harvard University
Cambridge, MA USA
chiningchou@g.harvard.edu

Alexander Golovnev
John A. Paulson School of Engineering and Applied Sciences
Harvard University
Cambridge, MA USA
alexgolovnev@gmail.com

Santhoshini Velusamy
John A. Paulson School of Engineering and Applied Sciences
Harvard University
Cambridge, MA USA
svelusamy@g.harvard.edu

***Abstract*—We prove tight upper and lower bounds on approximation ratios of all Boolean Max-2CSP problems in the streaming model. Specifically, for every type of Max-2CSP problem, we give an explicit constant α, s.t. for any $\epsilon > 0$ (i) there is an $(\alpha - \epsilon)$-streaming approximation using space $O(\log n)$; and (ii) any $(\alpha + \epsilon)$-streaming approximation requires space $\Omega(\sqrt{n})$. This generalizes the celebrated work of [Kapralov, Khanna, Sudan SODA 2015; Kapralov, Krachun STOC 2019], who showed that the optimal approximation ratio for Max-CUT was $1/2$.**

Prior to this work, the problem of determining this ratio was open for all other Max-2CSPs. Our results are quite surprising for some specific Max-2CSPs. For the Max-DICUT problem, there was a gap between an upper bound of $1/2$ and a lower bound of $2/5$ [Guruswami, Velingker, Velusamy APPROX 2017]. We show that neither of these bounds is tight, and the optimal ratio for Max-DICUT is $4/9$. We also establish that the tight approximation for Max-2SAT is $\sqrt{2}/2$, and for Exact Max-2SAT it is $3/4$. As a byproduct, our result gives a separation between space-efficient approximations for Max-2SAT and Exact Max-2SAT. This is in sharp contrast to the setting of polynomial-time algorithms with polynomial space, where the two problems are known to be equally hard to approximate. Finally, we prove that the tight streaming approximation for Max-kSAT is $\sqrt{2}/2$ for every $k \geq 2$.

***Keywords*-Streaming Algorithms; Approximation Algorithms; Constraint Satisfaction.**

I. Introduction

Maximum Boolean Constraint Satisfaction Problems, or Max-CSPs, are a central class of optimization problems, including as special cases problems such as Max-CUT, 3SAT, Graph Coloring, and Vertex Cover [2]. Given a set of allowed predicates $\mathcal{F}$, Max-CSP($\mathcal{F}$) is the optimization problem defined as follows. Every instance Ψ of the problem consists of a set of Boolean variables $\mathcal{X}$, and a set of constraints applied to them. Each constraint is a predicate from $\mathcal{F}$ applied to the variables from $\mathcal{X}$ or their negations. The goal is to compute the maximum number of simultaneously satisfiable constraints. For example, Max-kSAT is Max-CSP $\left(\mathcal{F}_{\mathsf{OR}_{\leq k}}\right)$ where $\mathcal{F}_{\mathsf{OR}_{\leq k}}$ is the set of OR predicates on at most k variables.

All proofs are deferred to the full version of the paper [1]

Schaefer's famous dichotomy theorem [3], [4] states that for any set of allowed predicates $\mathcal{F}$, solving Max-CSP($\mathcal{F}$) *exactly* is either in P or NP-hard. However, the landscape of *approximation* algorithms for Max-CSPs is much more complex (see [5] and references therein).

The Max-2CSP problem—Max-CSP where all constraints have length at most 2—is the most studied case of Max-CSP, and it generalizes many optimization problems on graphs. Starting with the seminal work of Goemans and Williamson [6], a series of works [7], [8], [9] developed a 0.87401-approximation algorithm for all Max-2CSPs, while under the P $\neq$ NP and Unique Games conjectures some Max-2CSPs do not admit 0.9001- and 0.87435-approximations, respectively [10], [11], [12].

In this paper, we follow the line of work [13], [14], [15], [16], [17], [18] that studies the *unconditional hardness* of approximating Max-2CSP through the lens of *streaming algorithms*. Over the last decade, there has been a lot of interest in designing algorithms for processing large streams of data using limited space (see [19], [20] and references therein). The streaming model was formally defined in [21], [22].

A streaming algorithm for a Max-2CSP problem makes one pass through the list of constraints and uses space that is sub-linear (ideally, poly-logarithmic) in the input size.[1] Since the algorithm is space bounded, it cannot even store an assignment to the input variables. Thus, a streaming algorithm is required to output an estimate of the maximum number of simultaneously satisfiable constraints. Specifically, for $\alpha \in [0, 1]$, an α-approximate streaming algorithm outputs a value v for which the following two conditions hold with probability $3/4$: (i) there exists an assignment σ satisfying at least v constraints, and (ii) $v \geq \alpha \cdot \mathsf{val}$, where val is the maximum number of simultaneously satisfiable constraints.

Prior to this work, the only Max-2CSP for which we knew the optimal streaming approximation factor was Max-CUT. Max-CUT asks us to find a bipartition of the n vertices of an undirected graph that maximizes the number

[1] In this work we focus on randomized streaming algorithms that make one pass over the input in a fixed (adversarial) order, and return the correct answer with probability $3/4$.

of edges crossing the partition—called the "cut". Note that Max-CUT corresponds to the Max-CSP($\mathcal{F}$) where $\mathcal{F}$ contains the binary XOR predicate.[2] [23] shows that exact streaming algorithms for Max-CUT require quadratic space $\Omega(n^2)$. Since a random partition of a graph with m edges has cut of expected size $m/2$, a trivial streaming algorithm 1/2-approximates Max-CUT with $O(\log m)$ space. It is also easy to see that for every $\epsilon > 0$, it suffices to store $\widetilde{O}(n)$ random edges of the graph to compute a $(1-\epsilon)$-approximation of Max-CUT. A recent line of work [14], [13], [15], [17] shows that these two trivial bounds are optimal, *i.e.*, any $(1/2+\epsilon)$-approximation algorithm requires linear space $\Omega(n)$.

However, the case for directed graphs is not nearly so well understood. In the Max-DICUT problem (another special case of Max-2CSP), given a directed graph, one needs to compute the maximum number of edges going from the first to the second part of the graph under any bipartition. While [17], [14] rules out a $(1/2+\epsilon)$-approximation for Max-DICUT too, the trivial algorithm gives only a 1/4-approximation here. [16] gives a 2/5-approximation for Max-DICUT, still leaving a gap between the upper and lower bounds.

Even the hardness of Max-2SAT is not known in the streaming setting. Recall that in Max-2SAT the only allowed predicates are variables and pairwise ORs. A random assignment gives a 1/2-approximation, and the classical $(\sqrt{5}-1)/2 \approx 0.61$-approximate algorithm of [24] can be implemented in $O(\log n)$ space using ℓ_1-sketching [25], [26]. No non-trivial upper bounds are known for Max-2SAT.

A. Our contribution

In this work, we resolve a natural question about the approximation guarantees of streaming algorithms for *every* Max-2CSP problem.

Before presenting our results, we need a way to classify Boolean functions of two variables. Let $f\colon \{0,1\}^2 \to \{0,1\}$ be a function, then

- f is of TR-type, or trivial, if f depends on at most one of its inputs (trivial functions are the two constant functions, and the four functions which depend on one of the inputs);
- f is of OR-type if the truth table of f has exactly one 0 and three 1s;
- f is of XOR-type if f depends on both inputs, and the truth table of f has exactly two 0s and two 1s;
- f is of AND-type if the truth table of f has exactly three 0s and one 1.

If a set of allowed predicates $\mathcal{F}$ contains only constraints of a type $\Lambda \in \{\mathsf{OR}, \mathsf{XOR}, \mathsf{AND}\}$, then the corresponding Max-2CSP problem is called Max-2EΛ (2-Exact-Λ, meaning that all constraints have length exactly 2). If $\mathcal{F}$ contains only Λ-type constraints and trivial constraints, then the corresponding Max-2CSP problem is called Max-2Λ.

We abuse notation by identifying a set of allowed predicates $\mathcal{F}$ with the set of types of its predicates. Also, for a set $\mathcal{F} = \{\Lambda\}$ containing one element, we write $\mathcal{F} = \Lambda$. Therefore, a Max-CSP($\mathcal{F}$) problem is defined by $\mathcal{F} \subseteq \{\mathsf{TR}, \mathsf{OR}, \mathsf{XOR}, \mathsf{AND}\}$. Note that every Max-2CSP problem corresponds to one such $\mathcal{F}$.

For every Max-CSP($\mathcal{F}$) problem, we give an explicit constant $\alpha_{\mathcal{F}}$ such that $(\alpha_{\mathcal{F}} - \epsilon)$-approximation can be computed in $O(\log n)$ space, while $(\alpha_{\mathcal{F}} + \epsilon)$-approximation requires space $\Omega(\sqrt{n})$, for every $\epsilon > 0$.

Theorem 1. *Let $\mathcal{F} \subseteq \{\mathsf{TR}, \mathsf{OR}, \mathsf{XOR}, \mathsf{AND}\}$ be a set of allowed binary predicates. Let $\alpha_{\mathcal{F}} = \min_{\mathcal{G} \subseteq \mathcal{F}} \alpha_{\mathcal{G}}$, where $\alpha_{\mathcal{G}}$ is given in Table I.*

For every $\epsilon > 0$, there exists an $(\alpha_{\mathcal{F}} - \epsilon)$-approximate streaming algorithm for Max-CSP($\mathcal{F}$) that uses space $O(\epsilon^{-2} \log n)$. On the other hand, any $(\alpha_{\mathcal{F}} + \epsilon)$-approximate streaming algorithm for Max-CSP($\mathcal{F}$) requires space $\Omega(\sqrt{n})$.

Type $\mathcal{G}$	Tight bound	Previous bound	
	$\alpha_{\mathcal{G}}$	$\alpha_{\mathcal{G}}^{\text{pr}}$	Reference
TR	1	1	Folklore
OR	$\frac{3}{4}$	$[\frac{3}{4}, 1]$	Folklore
{TR, OR}	$\frac{\sqrt{2}}{2}$	$[\frac{\sqrt{5}-1}{2}, 1]$	[24]
XOR	$\frac{1}{2}$	$\frac{1}{2}$	[17]
AND	$\frac{4}{9}$	$[\frac{2}{5}, \frac{1}{2}]$	[16]

Table I
SUMMARY OF KNOWN AND NEW APPROXIMATION FACTORS $\alpha_{\mathcal{G}}$ FOR MAX-CSP($\mathcal{G}$). WE HAVE SUPPRESSED $(1 \pm \epsilon)$ MULTIPLICATIVE FACTORS.

Discussion: Interestingly, Theorem 1 identifies five Max-2CSP problems which completely characterize the hardness of any Max-2CSP problem. Namely, we show that Max-CSP($\mathcal{F}$) is precisely as hard to approximate as the hardest of the problems from Table I expressible by predicates from $\mathcal{F}$.

In particular, Theorem 1 closes the gap between 2/5 [16] and 1/2 [14] for the streaming approximation ratio of Max-DICUT. We prove that neither of these bounds is tight, and that the correct bound is 4/9.[3] Similarly, it shows that the $(\sqrt{5}-1)/2$-approximate algorithm of [24] for Max-2SAT can be improved further, and that the optimal approximation ratio is $\sqrt{2}/2$.

Many streaming problems have space-accuracy trade-offs allowing for better approximations with more space (*e.g.*, [20], [27]). Curiously, Theorem 1 shows that every

[2]Although formally Max-CUT is a special case of Max-2XOR where all constraints are of the form $x_i \oplus x_j = 1$, it can be shown that these two problems are equivalent.

[3]While Theorem 1 states the bound for the Max-2AND problem only, it is easy to see that the proof in Section IV-C gives the same bound even for Max-DICUT.

$\mathsf{Max\text{-}2CSP}(\mathcal{F})$ problem exhibits sharp threshold behavior: it needs only logarithmic space to be approximated up to some constant $\alpha_{\mathcal{F}}$, and it requires polynomial space for every larger approximation factor.

In the classical setting, approximation algorithms for Max-CSPs use space-inefficient techniques including semidefinite and linear programming, and network flow computations [28], [29], [6], [30], [31], [32], [5]. On the other hand, the best streaming algorithms for Max-CSPs (except for the work [16]) used only random assignments to the variables of the instance, including Max-CUT, Max-2SAT, and Unique Games problems. We design streaming algorithms for the Max-2AND and Max-2OR problems (*i.e.*, $\mathcal{F} = \{\mathsf{TR}, \mathsf{AND}\}$ and $\mathcal{F} = \{\mathsf{TR}, \mathsf{OR}\}$) which significantly improve on the approximation ratios guaranteed by a random assignment to the variables.

Additionally, Theorem 1 reveals a curious difference between streaming approximation of the cases $\mathcal{G} = \mathsf{OR}$ and $\mathcal{G} = \{\mathsf{TR}, \mathsf{OR}\}$ (*i.e.*, Exact Max-2SAT and Max-2SAT). The former problem can be $3/4$-approximated, while the latter does not admit better than $\sqrt{2}/2$-approximations. This shows that adding trivial constraints to Exact Max-2SAT actually makes the problem harder to approximate. This is in sharp contrast to the classical setting of polynomial-time algorithms with polynomial space, where approximation-preserving reductions between the two problems are known [28]. While $3/4$-approximation for Exact Max-2SAT is trivial, many $3/4$-approximation algorithms for Max-2SAT use non-efficient (though polynomial) linear programming routines. This led Williamson to pose a question in 1998 whether there exists an algorithm for Max-2SAT which does not use linear programming and at least matches the trivial $3/4$-approximation guarantee for Exact Max-2SAT [33]. The affirmative answer to this question was given by Poloczek and Schnitger in 2011 [34], [35], [36], [37]. Theorem 1 complements this result by showing that there is no $\sqrt{2}/2 < 3/4$-approximation for Max-2SAT in the streaming setting, thus, separating space-efficient approximations for Max-2SAT and Exact Max-2SAT.

Our final contribution is a tight bound on the approximation ratio of streaming algorithms for all Max-kSAT problems. We generalize the $\sqrt{2}/2$-approximation algorithm for Max-2SAT from Theorem 1 to an algorithm for Max-SAT, and a matching hardness result trivially follows from the hardness of Max-2SAT.

Theorem 2. *For every $\epsilon > 0$, there exists an $(\sqrt{2}/2 - \epsilon)$-approximate streaming algorithm for Max-SAT that uses space $O(\epsilon^{-2} \log n)$. On the other hand, for any $k \geq 2, \epsilon > 0$ any $(\sqrt{2}/2 + \epsilon)$-approximate streaming algorithm for Max-kSAT requires space $\Omega(\sqrt{n})$.*

B. Related Work

Classical setting: For every $\mathsf{Max\text{-}2CSP}(\mathcal{F})$ problem, a random assignment satisfies in expectation a constant fraction $\alpha^{\mathrm{tr}}_{\mathcal{F}}$ of the constraints (this algorithm can be easily derandomized via the method of conditional expectations). In particular, this algorithm gives $1/2$- and $1/4$-approximations for Max-CUT and Max-2CSP. On one hand, Håstad [10] used the PCP theorem to show that some Max-CSP problems, *e.g.*, MAX-E3SAT, do not admit better than $\alpha^{\mathrm{tr}}_{\mathcal{F}}$-approximations unless $\mathsf{P} = \mathsf{NP}$. On the other hand, Goemans and Williamson [6] used semidefinite programming (SDP) to significantly improve the bounds for Max-CUT and Max-2CSP to 0.87856 and 0.79607. Håstad [30] proved that there is an SDP-based approximation algorithm with a better than $\alpha^{\mathrm{tr}}_{\mathcal{F}}$ approximation guarantee for every Max-2CSP. Many of the SDP-based approximation algorithms are optimal under the Unique Games Conjecture [38], [39]. We refer the reader to [5] for an up-to-date overview of the literature.

Streaming setting: While there is a trivial $1/2$-approximation for Max-CUT using space $O(\log n)$, Kapralov *et al.* [14] showed that for any constant $\epsilon > 0$, a $(1/2 + \epsilon)$-approximation requires space $\tilde{\Omega}(\sqrt{n})$. Independently, Kogan and Krauthgamer [13] showed that (i) $(1 - \epsilon)$-approximation requires space $\Omega(n^{1-\epsilon})$ and (ii) $4/5$-approximation requires $\Omega(n^{\tau})$ space for some constant $\tau > 0$. In a subsequent work, [15] showed that $(1 - \epsilon)$-approximation requires $\Omega(n)$ space. This line of work culminated in a recent result by Kapralov and Krachun [17] showing that any $(1/2 + \epsilon)$-approximation for Max-CUT requires $\Omega(n)$ space.

Recently Guruswami *et al.* [16] gave a $(2/5 - \epsilon)$-approximate algorithm for Max-DICUT for any constant $\epsilon > 0$, significantly improving on the trivial $1/4$-approximation. For k-SAT, Kogan and Krauthgamer [13] showed that there is a $(1 - \epsilon)$-approximation using $\tilde{O}(\epsilon^{-2} kn)$ space. The hardness side has been widely open prior to this work and, to the best of our knowledge, the only other hardness result is by Guruswami and Tao [18] showing that $(1/p + \epsilon)$-approximation for Unique Games with alphabet size p requires $\tilde{\Omega}(\sqrt{n})$ space for any constant $\epsilon > 0$.

C. Techniques

Streaming algorithms: The first step of our proof of Theorem 1 is two new algorithms for Max-2OR and Max-2AND that improve on the naive approximations for these problems. For these algorithms, we generalize the notion of bias [16] to all Max-2CSP problems, and prove a series of bounds on the value of Max-2CSP w.r.t. the bias (and the numbers of trivial and non-trivial constraints in the instance). This results in log-space streaming algorithms that sketch the bias (and some additional information about the instance), and compute good estimates of the value of the instance.

It is not hard to see that Max-2AND is the "hardest" Max-2CSP problem, *i.e.*, an α-approximation for Max-2AND implies α-approximations for all Max-2CSPs (see the full version of the paper). Therefore, the hardness result of [17] for Max-CUT holds for Max-2AND as well, ruling out the possibility of $(1/2+\epsilon)$-approximations. On the other hand, a random assignment for Max-2AND formulas only guarantees a $1/4$-approximation. A recent work [16] improves the approximation ratio to $(2/5-\epsilon)$ as follows.

Let Ψ be a Max-2EAND instance with m constraints, and val be the maximum number of simultaneously satisfiable constraints in Ψ. [16] defines the bias of a variable x as the absolute difference between the number of positive and negative occurrences of x, and the bias of the instance as the sum of biases of its variables. It is easy to see that for every instance, $\mathsf{val} \leq (m+\mathsf{bias})/2$. [16] proves that the assignment of the input variables according to their biases satisfies at least bias constraints (see Lemma 10). Then they conclude that $\max(\mathsf{bias}, m/4)$ is a $2/5$-approximation of val:

$$\frac{\max(\mathsf{bias}, m/4)}{\mathsf{val}} \geq \frac{\mathsf{bias}/5 + (m/4)(4/5)}{(m+\mathsf{bias})/2} = 2/5\,.$$

The upper and lower bounds of [16] are shown in red and blue in Figure 1, and the gap between the bounds indeed achieves $2/5$ when $\mathsf{bias} = m/4$. While both lower bounds $\mathsf{val} \geq \max(\mathsf{bias}, m/4)$ are tight as functions of bias and m, we show that in the important regime of low $\mathsf{bias} \in [0, m/3]$, these bounds can be improved to

$$\mathsf{val} \geq \frac{m}{4} + \frac{\mathsf{bias}^2}{4(m-2\mathsf{bias})}\,. \tag{3}$$

Unlike the lower bound of $\mathsf{val} \geq \mathsf{bias}$ from [16], our lower bound cannot be achieved by a greedy assignment to the input variables. Instead, we design a distribution of assignments, whose expected value is at least (3). This improved lower bound on val (shown in green in Figure 1) leads to a $4/9$-approximation by a sketch for the expression (3). Namely, we give a $O(\log n)$-space streaming algorithm that approximates the green and red bounds in Figure 1, and returns their maximum.

Perhaps surprisingly, the optimal approximation ratio for Max-2OR significantly differs from both the $3/4$-approximation for Max-2EOR, and the trivial $1/2$-approximation. The classical algorithm of [24] can be implemented in the streaming setting, but it only provides a $(\sqrt{5}-1)/2 \approx 0.61$-approximation. We prove that the tight bound for Max-2OR is even larger—$\sqrt{2}/2$. Proofs of these upper and lower bounds are perhaps the most technical parts of this work. It can be shown that various naive random assignments to the variables used in $1/2$- and $(\sqrt{5}-1)/2$-approximations cannot lead to better bounds. Instead we construct a family of distributions of assignments which depend on individual biases of the variables. We use these distributions to prove the existence of assignments of some high value v, and finally we show a way to approximate v in logarithmic space. We remark that it is not always possible to satisfy $m\sqrt{2}/2$ constraints, thus, we also prove non-trivial

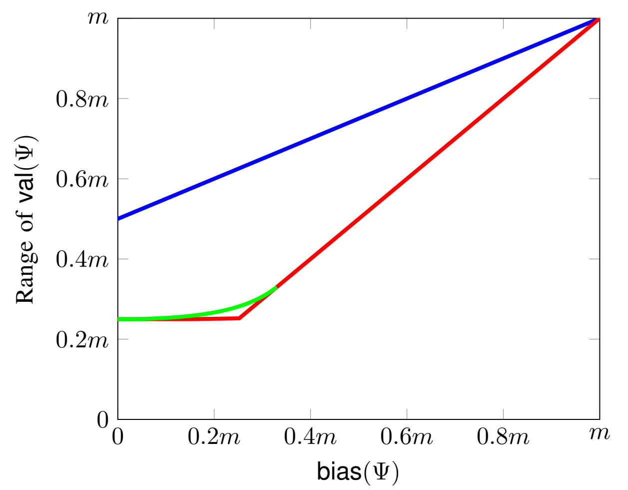

Figure 1. Upper and lower bounds on the maximum number val of simultaneously satisfiable constraints of Max-2AND as a function of bias. The blue line is the upper bound $\frac{m+\mathsf{bias}}{2}$, and the red line is the lower bound $\max\left(\frac{m}{4}, \mathsf{bias}\right)$ from [16] (see Lemma 10). The green line is the new lower bound $\frac{m}{4} + \frac{\mathsf{bias}^2}{4(m-2\mathsf{bias})}$ from Lemma 11 in the interval $\mathsf{bias} \in [0, m/3]$.

upper bounds on val for the case when our estimate v is low $v < m\sqrt{2}/2$. (See Lemma 13 and Lemma 14 for formal statements of these results.)

Hardness results: We develop a framework for proving hardness results for various Max-2CSP problems, and use it to establish tight bounds for every Max-2CSP. This framework is based on the communication complexity lower bound of [14] for the Distributional Boolean Hidden Partition problem (DBHP) (which, in turn, extends the results of [40], [41] for Boolean Hidden Matching and Boolean Hidden Hypermatching). In DBHP, Alice holds a random bipartition of $[n]$, and Bob has a (random) graph G on n vertices with some edges marked. Their goal is to use minimal communication to distinguish between the following two cases: in the **YES** case, the set of Bob's marked edges is exactly the edges of G that cross Alice's bipartition; while in the **NO** case, a random subset of the edges is marked. [14] proved a lower bound of $\Omega(\sqrt{n})$ on the randomized one-way communication complexity of DBHP.

For a set of allowed predicates $\mathcal{G}$, we construct a reduction from DBHP to Max-CSP($\mathcal{G}$), which naturally induces distributions $\mathcal{D}^Y$ and $\mathcal{D}^N$ of Max-CSP($\mathcal{G}$) instances. Then by a careful analysis we show that the gap between the optimal solutions of instances from $\mathcal{D}^Y$ and $\mathcal{D}^N$ achieves $\alpha_{\mathcal{G}} + \epsilon$ with high probability. This, amplified by a series of repetitions, lets us conclude that a space-efficient $(\alpha_{\mathcal{G}}+\epsilon)$-approximate algorithm would contradict the lower bound on the communication complexity of DBHP.

In our framework, we give separate reductions from DBHP to Max-2EAND and Max-2OR with approximation ratios $4/9+\epsilon$ and $\sqrt{2}/2+\epsilon$, respectively. For the Max-2EOR problem, we give an efficient streaming reduction from Max-CUT to Max-2EOR which asserts that an α-approximation for Max-2EOR implies an $\alpha/(3-2\alpha)$-approximation for Max-CUT. This, equipped with the lower bound from [17], proves a *linear* lower bound $\Omega(n)$ on the space complexity of $(3/4+\epsilon)$-approximations of Max-2EOR.

Putting it all together: Finally, we show that our algorithms for the five problems from Table I can be combined together to handle every Max-2CSP problem. Similarly, we prove that the established lower bounds for these five problems cover all possible Max-2CSPs. This implies that every Max-2CSP problem Max-CSP($\mathcal{F}$) is precisely as hard to approximate as the hardest problem from Max-TR, Max-2EOR, Max-2OR, Max-2EXOR, Max-2EAND expressible in $\mathcal{F}$, and finishes the proof of Theorem 1.

D. Structure

In Section II, we review some necessary background knowledge. In Section III, we provide streaming algorithms with optimal approximation ratios for all Max-2CSP problems. Sections IV and V are devoted to proving tight bounds on the approximation ratios of streaming algorithms from Section III. In particular, Section IV contains the general framework for our lower bounds, and the reductions from Distributional Boolean Hidden Partition to Max-2CSP problems. Section V provides a tight analysis of the approximation ratios resulting from these reductions. We defer all the proofs (including the proofs of Theorem 1 and Theorem 2) to the full version of the paper [1].

II. Preliminaries

Let $\mathbb{N}=\{1,2,\dots,\}$ be the set of natural numbers, and $[n]=\{1,2,\dots,n\}$ for any $n\in\mathbb{N}$. We use $\sqcup$ for the disjoint union of two sets. For an $0<\epsilon<1$, $B\in(1\pm\epsilon)$ is shorthand for $1-\epsilon\le B\le 1+\epsilon$. For ease of exposition we will abuse notation and associate a vector $X\in\{0,1\}^n$ with the set $X\subseteq[n], X=\{i\colon X_i=1\}$.

As we explained in Section I, we will primarily consider Max-CSP($\mathcal{G}$) where $\mathcal{G}\in\{\mathsf{TR},\mathsf{OR},\{\mathsf{TR},\mathsf{OR}\},\mathsf{XOR},\mathsf{AND}\}$. In order to get familiar with these problems, we provide several examples in Table II.

type $\mathcal{G}$	problem name	special case
OR	Max-2EOR	Exact Max-2SAT
{TR, OR}	Max-2OR	Max-2SAT
XOR	Max-2EXOR	Max-CUT
AND	Max-2EAND	Max-DICUT

Table II
For each case $\mathcal{G}\in\{\mathsf{OR},\{\mathsf{TR},\mathsf{OR}\},\mathsf{XOR},\mathsf{AND}\}$, we give the name of the Max-CSP($\mathcal{G}$) problem, as well as one well-studied special case/alternative name of the problem.

For an instance Ψ of a Max-2CSP problem, we denote the number of clauses (constraints) in Ψ by $m=|\Psi|$. We denote the set of Boolean variables of Ψ by $\mathcal{X}=\{x_1,\dots,x_n\}$. A literal ℓ is called positive if $\ell=x_i$, and negative if $\ell=\neg x_i$ for some variable x_i. A 1-clause is a clause (constraint) which depends only on one variable. We use $\mathrm{pos}_i^{(1)}(\Psi)$ and $\mathrm{pos}_i^{(2)}(\Psi)$ for the number of 1- and 2-clauses where the variable x_i appears positively. Similarly, $\mathrm{neg}_i^{(1)}(\Psi)$ and $\mathrm{neg}_i^{(2)}(\Psi)$ denote the number of 1- and 2-clauses containing $\neg x_i$.

For an assignment $\sigma:\mathcal{X}\to\{0,1\}$ of the variables of Ψ, we denote the number of clauses of Ψ satisfied by σ as $\mathsf{val}_\Psi(\sigma)$. We denote the maximum number of simultaneously satisfiable clauses in Ψ as val_Ψ:

$$\mathsf{val}_\Psi=\max_\sigma \mathsf{val}_\Psi(\sigma)\,.$$

For $\alpha\in[0,1]$ and a set of allowed predicates $\mathcal{F}$, an algorithm $\mathcal{A}$ is an α-approximation to the Max-CSP($\mathcal{F}$) problem if on any input Ψ, $\mathcal{A}$ outputs v, such that with probability $3/4$, it holds that $\mathsf{val}_\Psi\ge v\ge\alpha\cdot\mathsf{val}_\Psi$. For example, when $\alpha=1$, the algorithm solves Max-CSP($\mathcal{F}$) exactly (with probability $3/4$).

We will use the following definition of the bias of Ψ, which generalizes the definition from [16] to all Max-2CSPs with clauses of length 1 or 2.[4]

Definition 4 (Bias)**.** *The* bias *of a variable x_i of an instance Ψ is defined as*

$$\textit{bias}_i(\Psi)=\frac{1}{2}\cdot\left|2\mathrm{pos}_i^{(1)}(\Psi)+\mathrm{pos}_i^{(2)}(\Psi)-2\mathrm{neg}_i^{(1)}(\Psi)-\mathrm{neg}_i^{(2)}(\Psi)\right|.$$

The bias vector of Ψ is a vector $\boldsymbol{b}\in\mathbb{R}^n$, where $\boldsymbol{b}_i=\textit{bias}_i(\Psi)$. Finally, the bias *of the formula Ψ is defined as the sum of biases of its variables:*

$$\textit{bias}(\Psi)=\sum_{i=1}^n \textit{bias}_i(\Psi)$$

Note that for every formula Ψ with $|\Psi|=m$ clauses, $0\le\mathsf{bias}(\Psi)\le m$.

In order to approximate the bias of a formula Ψ, we will use a streaming algorithm for approximating the ℓ_1 norm of the bias vector of Ψ.

Theorem 5 ([25], [26])**.** *Given a stream S of $\mathsf{poly}(n)$ updates $(i,v)\in[n]\times\{1,-1\}$, let $x_i=\sum_{(i,v)\in S}v$ for $i\in[n]$. There exists a 1-pass streaming algorithm, which uses $O(\log n/\epsilon^2)$ bits of memory and outputs a $(1\pm\epsilon)$-approximation to the value $\ell_1(x)=\sum_i|x_i|$ with probability $3/4$.*

[4]For uniformity reasons, our definition of bias differs from the definition in [16] by a multiplicative factor of 2.

Finally, we will use the lower bound on the space complexity of streaming algorithms for approximate Max-CUT from [17].

Theorem 6. *For any constant $\epsilon > 0$, any streaming algorithm that $(1/2 + \epsilon)$-approximates Max-CUT with success probability at least $3/4$ requires $\Omega(n)$ space.*

A. Total variation distance

Definition 7 (Total variation distance of discrete random variables)**.** *Let Ω be a finite probability space and X, Y be random variables with support Ω. The total variation distance between X and Y is defined as follows.*

$$\|X - Y\|_{tvd} := \frac{1}{2} \sum_{\omega \in \Omega} |\Pr[X = \omega] - \Pr[Y = \omega]| \ .$$

We will use the two following properties of the total variation distance .

Proposition 8 ([14, Claim 6.5])**.** *Let Ω be a finite probability space and X, Y be random variables with support Ω.*

1) *(Triangle inequality) Let W be an arbitrary random variable, then we have $\|X - Y\|_{tvd} \geq \|X - W\|_{tvd} - \|Y - W\|_{tvd}$.*
2) *(Data processing inequality) Let W be a random variable that is independent of both X and Y, and f be a function, then we have $\|f(X, W) - f(Y, W)\|_{tvd} \leq \|X - Y\|_{tvd}$.*

III. Streaming Algorithms

In this section, we present optimal approximation algorithms for Max-2CSPs using $O(\log n)$ space. In Theorem 1 we will prove that it is actually sufficient to design optimal algorithms for Max-CSP($\mathcal{G}$) in the following five cases $\mathcal{G} \in \{\mathsf{TR}, \mathsf{OR}, \{\mathsf{TR}, \mathsf{OR}\}, \mathsf{XOR}, \mathsf{AND}\}$. In Section III-A, we present the trivial algorithm for Max-2CSPs, this algorithm turns our to be optimal for $\mathcal{G} \in \{\mathsf{TR}, \mathsf{OR}, \mathsf{XOR}\}$. Then we develop and analyze optimal algorithms for the cases $\mathcal{G} = \mathsf{AND}$ and $\mathcal{G} = \{\mathsf{TR}, \mathsf{OR}\}$ in Sections III-B and III-C, respectively.

For ease of exposition, we will assume that input instances never contain unsatisfiable and tautological clauses (*e.g.*, $(x \wedge \neg x), (x \vee \neg x)$). This assumption is without loss of generality, because a streaming algorithm can ignore unsatisfiable clauses and have a separate counter for tautological clauses.

A. Trivial Algorithm

First we present the trivial algorithm: this algorithm takes a Max-2CSP instance Ψ, counts the number of clauses $m = |\Psi|$ in it, and outputs the expected number of clauses satisfied by a uniform random assignment to the variables of Ψ. In Section IV we will show that this algorithm gives the best streaming approximation not only in the case of Max-2XOR (the Max-CUT problem), but also in the case of Max-2EOR.

Proposition 9 (Folklore)**.** *For a function $f : \{0,1\}^2 \to \{0,1\}$, let $\alpha_f \in [0,1]$ denote the fraction of 1s in its truth table. Then for a set of allowed predicates $\mathcal{F}$, we define $\alpha^{tr}_{\mathcal{F}} = \min_{f \in \mathcal{F}} \alpha_f$. There exists a streaming algorithm that uses $O(\log n)$ space, and computes $\alpha^{tr}_{\mathcal{F}}$-approximation for Max-CSP($\mathcal{F}$) with success probability 1.*

For example, for the problem Max-2EOR (*i.e.*, $\mathcal{F} = \{\mathsf{OR}\}$), we have $\alpha_{\mathsf{OR}} = 3/4$, as every clause is satisfied by 3 out of 4 possible assignments to its variables. Since the problem Max-2OR (*i.e.*, $\mathcal{F} = \{\mathsf{TR}, \mathsf{OR}\}$) also allows clauses of length 1 (which are satisfied by 1 out of 2 possible assignments to the variable), we have $\alpha_{\{\mathsf{TR},\mathsf{OR}\}} = 1/2$.

Algorithm 1 $\alpha^{\text{tr}}_{\mathcal{F}}$-approximation streaming algorithm for Max-CSP($\mathcal{F}$)

Input: Ψ—an instance of Max-CSP($\mathcal{F}$).
1: Use $O(\log n)$ bits to compute $m = |\Psi|$.
Output: $v = \alpha^{\text{tr}}_{\mathcal{F}} \cdot m$.

Remark. *For an $(\alpha^{tr}_{\mathcal{F}} - \epsilon)$-approximation, one can reduce the space usage of Algorithm 1 to $O(\log \log n + \log(1/\epsilon))$ bits by using the approximate counting algorithm of Morris [42], [43].*

Remark. *Formally, Algorithm 1 only guarantees a $1/2$-approximation for the problem Max-CSP(TR),* i.e.*, the problem where all clauses have length 1. In this case, in order to achieve a $(1 - \epsilon)$-approximation using $O(\log n)$ space for arbitrary constant $\epsilon > 0$, one can use an ℓ_1-sketch (Theorem 5) to approximate the bias vector of the input formula. Indeed, it is easy to see that for an instance Ψ of Max-CSP(TR) with m clauses, $\mathsf{val}_\Psi = (m + \mathsf{bias}(\Psi))/2$.*

We give $\alpha^{\text{tr}}_{\mathcal{G}}$ for relevant sets of predicates in Table III.

Type $\mathcal{G}$	TR	OR	{TR, OR}	XOR	AND
$\alpha^{\text{tr}}_{\mathcal{G}}$	1	$\frac{3}{4}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{4}$
$\alpha^{\text{opt}}_{\mathcal{G}}$	1	$\frac{3}{4}$	$\frac{\sqrt{2}}{2}$	$\frac{1}{2}$	$\frac{4}{9}$

Table III
For various sets of predicates $\mathcal{G}$, the table presents (i) $\alpha^{\text{TR}}_{\mathcal{G}}$—the approximation ratio guaranteed by the trivial algorithm for Max-CSP($\mathcal{G}$), and (ii) $\alpha^{\text{OPT}}_{\mathcal{G}}$—the optimal approximation ratio of streaming algorithms, proven in Sections III and IV for Max-CSP($\mathcal{G}$). We have suppressed $(1 - \epsilon)$ multiplicative factors for the case $\mathcal{G} = \mathsf{TR}$.

As we show in the following sections, this trivial approximation algorithm can be improved for the Max-2AND and Max-2OR problems.

B. Algorithm for Max-2AND and Max-2EAND

Consider a Max-2AND instance Ψ' where all clauses are of length 1 or 2. Note that Ψ' can be written as an equivalent Max-2AND instance Ψ, where 1-clauses of Ψ' are replaced with 2-clauses containing the same literal twice.[5] In this section, we will consider such representation of every instance of Max-2AND, *i.e.*, we assume that all clauses have exactly 2 (not necessarily distinct) literals. Note that in this case, the bias (see Definition 4) of Ψ is simply

$$\mathsf{bias}(\Psi) = \frac{1}{2}\sum_{i=1}^{n} |\mathrm{pos}_i^{(2)}(\Psi) - \mathrm{neg}_i^{(2)}(\Psi)|\,,$$

where $\mathrm{pos}_i^{(2)}(\Psi)$ and $\mathrm{neg}_i^{(2)}(\Psi)$ are the numbers of occurrences of x_i and $\neg x_i$ in 2-clauses.

[16] gave lower and upper bounds for the maximum number of satisfied clauses val_Ψ in terms of $\mathsf{bias}(\Psi)$ and m (the number of clauses in Ψ). For the sake of being self contained, and to verify that these bounds hold for our slightly more general case where 2-clauses may contain repeated literals, we present the proofs of these bounds in the full version of the paper.

Lemma 10 ([16])**.** *Let Ψ by a Max-2AND instance with m clauses. Then*

$$\mathsf{bias}(\Psi) \le \mathsf{val}_\Psi \le \frac{m + \mathsf{bias}(\Psi)}{2}\,.$$

We improve the lower bound of [16] in the important regime of $\mathsf{bias}(\Psi) \le m/3$ in the following lemma.

Lemma 11. *biaslb Let Ψ by a Max-2AND instance with m clauses and $\mathsf{bias}(\Psi) \le m/3$. Then*

$$\mathsf{val}_\Psi \ge \frac{m}{4} + \frac{\mathsf{bias}(\Psi)^2}{4(m - 2\mathsf{bias}(\Psi))} \ge \frac{2(m + \mathsf{bias}(\Psi))}{9}\,.$$

The proof of Lemma 11 is based on *biased random sampling*, and appears in the full version of the paper. For a pictorial view of this improvement, see Figure 1.

We are now ready to present a streaming algorithm that $(4/9)$-approximates Max-2AND and Max-2EAND.

Theorem 12 ($\frac{4}{9}$–approximation for Max-2AND and Max-2EAND)**.** *For any $\epsilon \in (0, 0.01)$, there exists a streaming algorithm that uses space $O(\epsilon^{-2}\log n)$ and computes $\left(\frac{4}{9} - \epsilon\right)$-approximation for Max-2AND and Max-2EAND with success probability at least $3/4$.*

C. Algorithm for Max-2OR

For the case of Max-2OR, it is crucial to distinguish 1- and 2-clauses. Therefore, we treat clauses containing two identical literals as 1-clauses. We denote the number of 1-clauses of Ψ by m_1, and the number of 2-clauses by m_2. In particular, the total number of clauses is $m = m_1 + m_2$.

In Lemmas 13 and 14 we give upper and lower bounds on val_Ψ in terms of m_1, m_2, and $\mathsf{bias}(\Psi)$. In this section we prove that the ratio between the presented lower and upper bounds is bounded by $\frac{\sqrt{2}}{2}$, and that there is a $O(\log n)$-space algorithm that sketches the lower bounds of Lemma 14 on val_Ψ.

When the bias of Ψ is large (say, $\mathsf{bias}(\Psi) = m$), it might be possible to satisfy all m clauses of Ψ, so no non-trivial upper bounds on val_Ψ can be proven in terms of bias in this case. Even if the bias is low (say, $\mathsf{bias}(\Psi) = 0$), but the formula does not contain 1-clauses, it might still be possible to satisfy all clauses of Ψ. (*E.g.*, if all clauses of Ψ contain one positive and one negative literal.) It turns out that for the optimal approximation ratio, we need to bound from above val_Ψ in the case of low bias and large number of 1-clauses.

Lemma 13. *Let Ψ be a Max-2OR instance with m_1 1-clauses, and m_2 2-clauses. Then*

$$\mathsf{val}(\Psi) \le \min\left\{m_1 + m_2, \frac{m_1 + 2m_2 + \mathsf{bias}(\Psi)}{2}\right\}.$$

The trivial algorithm guarantees that for every Max-2OR instance Ψ, $\mathsf{val}_\Psi \ge m_1/2 + 3m_2/4$. While this bound is tight in terms of m_1 and m_2, for instances with high $\mathsf{bias} > m_2/2$, we prove a better lower bound of $\mathsf{val} \ge (m_1 + m_2 + \mathsf{bias}(\Psi))/2$. Clearly, this bound is not sufficient for a better than $1/2$-approximation in the case of low $\mathsf{bias}(\Psi) = 0$. In order to handle this case, we design a distribution of assignments which in expectation satisfy a large number of clauses in formulas with low bias.

Lemma 14. *Let Ψ be a Max-2OR instance with m_1 1-clauses, and m_2 2-clauses. Then leftmargin=**

1) $\mathsf{val}_\Psi \ge \frac{m_1 + m_2 + \mathsf{bias}(\Psi)}{2}$;
2) *if $\mathsf{bias}(\Psi) \le m_2$, then*

$$\mathsf{val}_\Psi \ge \frac{m_1}{2} + \frac{3m_2}{4} + \frac{\mathsf{bias}(\Psi)^2}{4m_2}\,.$$

Algorithm 2 $\left(\frac{4}{9} - \epsilon\right)$-approximation streaming algorithm for Max-2AND

Input: Ψ—an instance of Max-2AND. Error parameter $\epsilon \in (0, 0.01)$.
1: Approximate the ℓ_1-norm of the bias vector with error $\delta = \epsilon/2$ (Theorem 5):
Compute $B \in (1 \pm \delta)\,\mathsf{bias}(\Psi)$.
2: Count the number of clauses $m = |\Psi|$.
3: **if** $B \in \left[0, \frac{m}{3}(1-\delta)\right]$ **then**
Output: $v = \frac{2(m+B)}{9(1+\delta)}$.
4: **else**
Output: $v = \frac{B}{(1+\delta)}$.

[5]We only apply this transformation to Max-2AND instances, because here it plays in our favor. For example, an AND clause with repeated literals is satisfied by a uniform random assignment with probability $1/2$, while an AND clause with distinct variables is satisfied with probability only $1/4$. For the case of OR, a clause with repeated literals would be satisfied only with probability $1/2$, while an OR clause with distinct variables would be satisfied with probability $3/4$.

We will also use the following simple claim.

Claim 15. *For every $x \geq 0, y > 0$:*

$$\frac{2x + 3y + x^2/y}{4(x+y)} \geq \frac{\sqrt{2}}{2}\,.$$

Now we are ready to present an approximation algorithm for the Max-2OR problem.

Theorem 16 ($\frac{\sqrt{2}}{2}$–approximation for Max-2OR)**.** *For any $\epsilon \in (0, 0.01)$, there exists a streaming algorithm that uses space $O(\epsilon^{-2}\log n)$ and computes $\left(\frac{\sqrt{2}}{2} - \epsilon\right)$-approximation for Max-2OR with success probability at least $3/4$.*

Algorithm 3 $\left(\frac{\sqrt{2}}{2} - \epsilon\right)$-approximation streaming algorithm for Max-2OR

Input: Ψ—an instance of Max-2OR. Error parameter $\epsilon \in (0, 0.01)$.

1: Approximate the ℓ_1-norm of the bias vector with error $\delta = \epsilon/4$ (Theorem 5):
Compute $B \in (1 \pm \delta)\,\mathsf{bias}(\Psi)$.

2: Count the number of 1- and 2- clauses m_1 and m_2.

3: **if** $B \in [0, (1-\delta)m_2]$ **then**
Output: $v = \frac{(1-\delta)^2(2m_1+3m_2+B^2/m_2)}{4}$.

4: **else**
Output: $v = \frac{(1-\delta)(m_1+m_2+B)}{2}$.

D. Algorithm for for Max-kSAT

We first extend the notion of bias which we defined in Section II to Max-kSAT instances. Similarly, $\mathrm{neg}_i^{(r)}(\Psi)$ denotes the number of r-clauses containing $\neg x_i$.

Definition 17. *The* bias *of a variable x_i of an instance Ψ of Max-kSAT is defined as*

$$\mathsf{bias}_i(\Psi) = \left|\sum_r \frac{1}{2^{r-1}}\left(\mathrm{pos}_i^{(r)}(\Psi) - \mathrm{neg}_i^{(r)}(\Psi)\right)\right|\,.$$

The bias vector of Ψ is a vector $\boldsymbol{b} \in \mathbb{R}^n$, where $\boldsymbol{b}_i = \mathsf{bias}_i(\Psi)$. Finally, the bias *of the formula Ψ is defined as the sum of biases of its variables: $\mathsf{bias}(\Psi) = \sum_{i=1}^n \mathsf{bias}_i(\Psi)$.*

In the full version of the paper we show that Algorithm 4 finds a $(\sqrt{2}/2 - \epsilon)$-approximation for Max-kSAT. This, together with the $(\sqrt{2}/2 + \epsilon)$ lower bound for Max-2SAT from Theorem 1, finishes the proof of Theorem 2.

IV. Space Lower Bounds for Approximating Boolean Max-2CSP

In this section, we establish space lower bounds for streaming approximations for all Max-2CSPs. In Theorem 1 we will show that it suffices to prove lower bounds for Max-CSP($\mathcal{G}$) for the following four cases $\mathcal{G} \in \{\mathsf{OR}, \{\mathsf{TR}, \mathsf{OR}\}, \mathsf{XOR}, \mathsf{AND}\}$. A linear space lower

---

Algorithm 4 $(\sqrt{2}/2-\epsilon)$-approximation streaming algorithm for Max-kSAT)

Input: Ψ—an instance of Max-kSAT. Error parameter $\epsilon \in (0, 0.01)$.

1: Approximate the ℓ_1-norm of the bias vector with error $\delta = \epsilon/8$ (Theorem 5):
Compute $B \in (1 \pm \delta)\,\mathsf{bias}(\Psi)$.

2: Use three counters to count the number of 1-clauses, m_1, the number of 2-clauses, m_2, and the number of clauses that depend on at least three variables, $m_{\geq 3}$.

3: **if** $B \in \left[0, (1-\delta)(m_2 \cdot \frac{\sqrt{2}}{2} + m_{\geq 3} \cdot \frac{5\sqrt{2}}{8})\right]$ **then**
Output: $v = \frac{(1-\delta)^2(4m_1+6m_2+7m_{\geq 3})}{8} + \frac{(1-\delta)^2B^2}{4(m_2+2.5m_{\geq 3})}$.

4: **else**
Output: $v = \frac{(1-\delta)(m_1+m_2+m_{\geq 3}+B)}{2}$.

---

bound for the case $\mathcal{G} = \mathsf{XOR}$ is proven by Kapralov and Krachun [17]. We use this result to prove a linear lower bound for the case $\mathcal{F} = \mathsf{OR}$ in Section IV-A. We prove the two remaining lower bounds by reductions from the communication complexity problem DBHP [14]. In Section IV-B, we present a general framework for proving such lower bounds, while in Sections IV-C and IV-D we give specific reductions for the Max-2AND and Max-2OR problems. Finally, some technical results used in the framework in are deferred to the full version of the paper.

A. From Max-2EXOR to Max-2EOR

In this section, we give a simple streaming reduction from Max-CUT to Max-2EOR, which asserts that a better than trivial $3/4$-approximation for Max-2EOR would lead to a better then trivial $1/2$-approximation for Max-CUT. Since the latter is known to require linear space [17], we get a linear lower bound on the space complexity of $(3/4+\epsilon)$-approximations of Max-2EOR.

Lemma 18 (Folklore)**.** *Let Ψ_{XOR} be a Max-2EXOR instance with m clauses. Consider the following reduction from Ψ_{XOR} to Ψ_{OR}, a Max-2EOR instance: For every clause $(x \oplus y)$ in Ψ_{XOR}, we add clauses $(x \vee y)$ and $(\neg x \vee \neg y)$ to Ψ_{OR}. Then*

$$\mathsf{val}_{\Psi_{\mathsf{OR}}} = m + \mathsf{val}_{\Psi_{\mathsf{XOR}}}\,.$$

Corollary 19. *For any constant $\epsilon > 0$, any streaming algorithm that $(3/4+\epsilon)$-approximates Max-2EOR with success probability at least $3/4$ requires $\Omega(n)$ space.*

B. Distributional Boolean Hidden Partition (DBHP) Problem

We prove lower bounds for Max-2EAND and Max-2OR in two steps. Recall that the goal of the players in DBHP is to distinguish between two distributions **YES** and **NO**. First, we show a reduction from DBHP to Max-CSP($\mathcal{G}$). This induces a **YES** and a **NO** distributions of instances of $\mathsf{Max\text{-}CSP}(\mathcal{G})$, corresponding to the **YES** and **NO** cases of DBHP. Next, we show that with high probability there is a gap between the optimal value of instances from the **YES** and **NO** distributions. The ratio α between these optimal values will be the upper bound on the approximation ratio of space-efficient streaming algorithms. Informally, any $(\alpha+\epsilon)$-approximate streaming algorithm with space s distinguishes the distributions **YES** and **NO**, and, therefore, can be converted into a communication protocol for DBHP that uses s bits of communication. Since Kapralov, Khanna, and Sudan [14] proved that any communication protocol for DBHP requires at least $\Omega(\sqrt{n})$ bits of communication, the corresponding space lower bound for streaming algorithms follows.

Before presenting the framework for streaming lower bounds, we need to define DBHP and adjust it to our setting.

For $n \in \mathbb{N}$ and $p \in [0,1]$, by $G(n,p)$ we denote the Erdös-Rényi distribution of undirected graphs with n vertices, where each edge is chosen independently with probability p.

Definition 20 (DBHP)**.** *Let $n \in \mathbb{N}$, $\beta \in (0, 1/16)$ be parameters. Let $X^* \in \{0,1\}^n$ be a uniformly random vector, and G be a random graph sampled from $G(n, 2\beta/n)$. Let r be the number of edges in G, and $M \in \{0,1\}^{r\times n}$ be the edge-vertex incidence matrix of G. We will consider the following three distributions of a vector $w \in \{0,1\}^r$.*

- *(**YES** distribution) $w = MX^* \in \{0,1\}^r$, where the arithmetic is over $\mathbb{F}_2$;*
- *(**NO** distribution) $w = \mathbf{1} + MX^* \in \{0,1\}^r$, where $\mathbf{1} \in \mathbb{F}_2^r$ is the all 1s vector, and the arithmetic is over $\mathbb{F}_2^r$;*
- *($\overline{\textbf{NO}}$ distribution) w be uniformly sampled from $\{0,1\}^r$.*

For a pair of distinct distributions $\mathcal{D} \neq \mathcal{D}' \in \{\textbf{YES}, \textbf{NO}, \overline{\textbf{NO}}\}$, we consider the following decisional 2-player one-way communication problem $\mathsf{DBHP}_{\mathcal{D},\mathcal{D}'}(n,\beta)$. Alice receives $X^* \in \{0,1\}^n$, and Bob receives (M,w) as their private inputs, where w is sampled from $\mathcal{D}$ or $\mathcal{D}'$ with probability $1/2$. A communication protocol Π for $\mathsf{DBHP}_{\mathcal{D},\mathcal{D}'}(n,\beta)$ consists of a message m sent from Alice to Bob. The complexity of the protocol Π is the length of the message m: $|\Pi| := |m|$. The goal of the players is to distinguish between the distributions $\mathcal{D}$ and $\mathcal{D}'$, and the success probability of Π is defined as $\Pr_{(M,w)\sim\mathcal{D}}[$Bob outputs $\mathcal{D}]/2 + \Pr_{(M,w)\sim\mathcal{D}'}[$Bob outputs $\mathcal{D}']/2$.

[14] showed that for any constant $\delta > 0$, any protocol that solves $\mathsf{DBHP}_{\mathbf{YES},\overline{\mathbf{NO}}}(n,\beta)$ with success probability $(1/2+\delta)$ requires $\Omega(\beta^{3/2}\sqrt{n})$ bits of communication. The next lemma shows that the same lower bound extends to the $\mathsf{DBHP}_{\mathbf{YES},\mathbf{NO}}$ problem by an application of the triangle inequality.

Lemma 21 (A modification of [14, Lemma 5.1])**.** *Let $\beta \in (n^{-1/10}, 1/16)$ and $s \in (n^{-1/10}, 1)$ be parameters. Any protocol Π for $\mathsf{DBHP}_{\mathbf{YES},\mathbf{NO}}(n,\beta)$ that uses $s\sqrt{n}$ bits of communication cannot distinguish between the **YES** and **NO** distributions with success probability more than $1/2 + c\cdot(\beta^{3/2}+s)$ for some constant $c > 0$ and all large enough n.*

For ease of exposition, now we will use $\mathsf{DBHP}(n,\beta)$ to denote $\mathsf{DBHP}_{\mathbf{YES},\mathbf{NO}}(n,\beta)$.

Finally, note that the graph G in the definition of DBHP is extremely sparse (in expectation it has $r \approx \beta n < 0.1n$ edges), and, thus, it is not immediately useful for designing hard instances of Max-2CSP problems. In order to overcome this issue, [14] used DBHP where Bob receives a collection of T messages all sampled either from the **YES** or **NO** distribution. Now the union of the T sparse graphs received by Bob can be used in reductions to Max-2CSPs.

Definition 22 (DBHP with T messages)**.** *For any $\beta \in (0, 1/16)$ and $n, T \in \mathbb{N}$, we define $\mathsf{DBHP}(n,\beta,T)$ as follows. Let $X^* \in \{0,1\}^n$ be a uniformly random vector, and for $1 \le t \le T$, let G_i be a random graph sampled from $G(n, 2\beta/n)$, and M_i be the edge-vertex incidence matrix of G_i. Alice receives X^*, and Bob receives a list $(M_1, w_1), \dots, (M_T, w_T)$, where with probability $1/2$ all $w_t = M_t X^*$ (**YES** case), and with probability $1/2$ all $w_t = \mathbf{1} + M_t X^*$ (**NO** case). The goal of the players is to have a non-trivial advantage over a random guess in distinguishing between the two distributions, while only communication from Alice to Bob is allowed.*

Reduction from DBHP: A reduction from $\mathsf{DBHP}(n,\beta,T)$ to $\mathsf{Max\text{-}CSP}(\mathcal{G})$ is defined by a pair of algorithms, $\mathcal{A}$ and $\mathcal{B}$. Alice receives her input vector $X^* \in \{0,1\}^n$, runs $\mathcal{A}$ on the input X^*, and outputs a set of $\mathsf{Max\text{-}CSP}(\mathcal{G})$-clauses. Bob receives a collection of T pairs (M_t, w_t), applies $\mathcal{B}$ to each of them, and outputs T sets of $\mathsf{Max\text{-}CSP}(\mathcal{G})$-clauses. Finally, the resulting instance of the $\mathsf{Max\text{-}CSP}(\mathcal{G})$ problem is the union of clauses from $\mathcal{A}(X^*), \mathcal{B}(M_1,w_1), \dots, \mathcal{B}(M_T,w_T)$.

The reduction above naturally induces two distributions $\mathcal{D}^Y(\beta,T,\mathcal{A},\mathcal{B})$ and $\mathcal{D}^N(\beta,T,\mathcal{A},\mathcal{B})$ of $\mathsf{Max\text{-}CSP}(\mathcal{G})$ instances, corresponding to the **YES** and **NO** distributions of (M_t,w_t). Let us pick some v^Y and v^N, such that $\Pr_{\Psi\sim\mathcal{D}^Y}[\mathsf{val}_\Psi \ge v^Y] > 1 - o(1)$ and $\Pr_{\Psi\sim\mathcal{D}^N}[\mathsf{val}_\Psi \le v^N] > 1 - o(1)$. Note that for any $\alpha > v^N/v^Y$, an α-approximate streaming algorithm for $\mathsf{Max\text{-}CSP}(\mathcal{G})$ distinguishes the two distributions $\mathcal{D}^Y(\beta,T,\mathcal{A},\mathcal{B})$ and $\mathcal{D}^N(\beta,T,\mathcal{A},\mathcal{B})$ with high probability. The following theorem states that any streaming algorithm that distinguishes these two distributions, requires space $\Omega(\sqrt{n})$. In particular, any streaming α-approximation for $\mathsf{Max\text{-}CSP}(\mathcal{G})$ requires space at least $\Omega(\sqrt{n})$.

Theorem 23 (Reduction from DBHP with T messages)**.** *Let $c > 0$ be the constant from Lemma 21. For every $T \in \mathbb{N}$, $0 < \beta \le 1/(10cT)^{2/3}$, and reduction $(\mathcal{A},\mathcal{B})$*

from DBHP *to* Max-CSP$(\mathcal{G})$*, any streaming algorithm that distinguishes* $\mathcal{D}^Y(\beta, T, \mathcal{A}, \mathcal{B})$ *and* $\mathcal{D}^N(\beta, T, \mathcal{A}, \mathcal{B})$ *with success probability at least* $3/4$ *requires space at least* $\frac{1}{40cT} \cdot \sqrt{n}$.

The proof of Theorem 23 follows the proofs in [14] by using the standard hybrid argument as well as the data processing inequality for total variation. First we describe reductions from DBHP to Max-2EAND and Max-2OR in Sections IV-C and IV-D, respectively.

C. From DBHP to Max-2EAND

Now, we describe the reduction from DBHP to Max-2EAND. To describe the reduction, it suffices to specify the parameters β and T, and the algorithms $\mathcal{A}^{\mathsf{EAND}}$ and $\mathcal{B}^{\mathsf{EAND}}$. Recall that we associate a vector $X^* \in \{0,1\}^n$ with the set of its ones: $X \subseteq [n], X = \{i\colon X_i = 1\}$. Also, recall that the input of Bob, (M, w), consists of an edge-vertex incidence matrix $M \in \{0,1\}^{r \times n}$ and a vector $w \in \{0,1\}^r$. In particular, every row of M has exactly two ones.

Reduction from DBHP to Max-2EAND

- Let $c > 0$ be the constant from Lemma 21. For a given error parameter $\epsilon \in (0,1)$, let $T = (10000/\epsilon^2)^3 \cdot (10c)^2$ and $\beta = \frac{1}{(10cT)^{2/3}}$ such that $\beta T = 10000/\epsilon^2$.
- $\mathcal{A}^{\mathsf{EAND}}(X^*)$: Sample $\beta n T/4$ independent pairs $(i,j) \in X^* \times \overline{X^*}$, and for each of them output the clause $(x_i \wedge \neg x_j)$.
- $\mathcal{B}^{\mathsf{EAND}}(M, w)$: Let r be the number of rows in M. For each $1 \le k \le r$ with $w_k = 1$, let the 1s in the k^{th} row of M be at the i^{th} and j^{th} positions, then output two clauses: $(x_i \wedge \neg x_j)$ and $(\neg x_i \wedge x_j)$.

Lemma 24. *lemma For any* $\epsilon \in (0,1)$*, let* $(\beta, T, \mathcal{A}^{\mathsf{EAND}}, \mathcal{B}^{\mathsf{EAND}})$ *be the parameters described in the above reduction. For a* Max-2EAND *instance* Ψ*, let* m_Ψ *denote the number of clauses in* Ψ*. Then*

$$\Pr_{\Psi \sim \mathcal{D}^Y(\beta, T, \mathcal{A}^{\mathsf{EAND}}, \mathcal{B}^{\mathsf{EAND}})}\left[\mathsf{val}_\Psi < \left(\frac{3}{5} - \epsilon\right) \cdot m_\Psi\right] = o(1)$$

and

$$\Pr_{\Psi \sim \mathcal{D}^N(\beta, T, \mathcal{A}^{\mathsf{EAND}}, \mathcal{B}^{\mathsf{EAND}})}\left[\mathsf{val}_\Psi > \left(\frac{4}{15} + \epsilon\right) \cdot m_\Psi\right] = o(1)\,.$$

An immediate corollary of Theorem 23 and Lemma 24 is the desired lower bound for streaming approximation of Max-2EAND.

Corollary 25. *For any constant* $\epsilon \in (0,1)$*, any streaming algorithm that* $(4/9 + \epsilon)$*-approximates* Max-2EAND *with success probability at least* $3/4$ *requires* $\Omega(\sqrt{n})$ *space.*

D. From DBHP to Max-2OR

Now, we describe the reduction from DBHP to Max-2OR. Again, it suffices to specify the parameters β and T, and the algorithms $\mathcal{A}^{\mathsf{OR}}$ and $\mathcal{B}^{\mathsf{OR}}$.

Reduction from DBHP to OR

- Let $c > 0$ be the constant from Lemma 21. For a given error parameter $\epsilon \in (0,1)$, let $T = (10000/\epsilon^2)^3 \cdot (10c)^2$ and $\beta = \frac{1}{(10cT)^{2/3}}$ such that $\beta T = 10000/\epsilon^2$.
- $\mathcal{A}^{\mathsf{OR}}(X^*)$: Sample $\frac{\sqrt{2}-1}{2} \cdot \beta n T$ independent copies of $i \in X^*$, and for each of them output the 1-clause (x_i). Sample another $\frac{\sqrt{2}-1}{2} \cdot \beta n T$ independent copies of $j \in \overline{X^*}$, and for each of them output the 1-clause $(\neg x_j)$.
- $\mathcal{B}^{\mathsf{OR}}(M, w)$: Let r be the number of rows in M. For each $1 \le k \le r$ with $w_k = 1$, let the the 1s in the k^{th} row of M be at the i^{th} and j^{th} positions, then output two clauses: $(x_i \vee x_j)$ and $(\neg x_i \vee \neg x_j)$.

Lemma 26. *For any* $\epsilon \in (0,1)$*, let* $(\beta, T, \mathcal{A}^{\mathsf{OR}}, \mathcal{B}^{\mathsf{OR}})$ *be the parameters described in the above reduction. For a* Max-2OR *instance* Ψ*, let* m_Ψ *denote the number of clauses in* Ψ*. Then*

$$\Pr_{\Psi \sim \mathcal{D}^Y(\beta, T, \mathcal{A}^{\mathsf{OR}}, \mathcal{B}^{\mathsf{OR}})}[\mathsf{val}_\Psi = m_\Psi] = 1$$

and

$$\Pr_{\Psi \sim \mathcal{D}^N(\beta, T, \mathcal{A}^{\mathsf{OR}}, \mathcal{B}^{\mathsf{OR}})}\left[\mathsf{val}_\Psi > \left(\frac{\sqrt{2}}{2} + \epsilon\right) \cdot m_\Psi\right] = o(1)\,.$$

Now, the desired lower bound for any streaming approximations for Max-2OR immediately follows from Theorem 23 and Lemma 26.

Corollary 27. *For any constant* $\epsilon \in (0,1)$*, any streaming algorithm that* $(\sqrt{2}/2 + \epsilon)$*-approximates* Max-2OR *with success probability at least* $3/4$ *requires* $\Omega(\sqrt{n})$ *space.*

V. Analysis for the gap of Max-2EAND and Max-2OR instances

In order to prove Lemma 24 and Lemma 26, we present an intuitive and graphical view of the reductions. In this section, we interchangeably work with one of the following representations for σ in order to simplify the presentation. Previously, σ was defined as a function that maps $\{x_1, x_2, \ldots, x_n\}$ to $\{0,1\}$. It can be represented by a vector in $\{0,1\}^n$ which has $\sigma(x_i)$ as its i^{th} coordinate. It can also be represented by the set $\{i \in [n] : \sigma(x_i) = 1\}$.

Recall that in DBHP, Bob has private inputs $M \in \{0,1\}^r$ and $w \in \{0,1\}^r$, where M is the edge-incidence matrix of an n-vertex graph G and w is an indicator vector. Specifically, M corresponds to a graph sampled from $G(n, 2\beta/n)$ and r denotes the number of edges in this graph. We focus on the subgraph $H \subseteq G$ that contains only those edges from M whose corresponding entry in w is 1. We examine the distributions of this subgraph H under different input distributions to DBHP. Recall that we are interested in two input distributions to DBHP: **YES** and **NO**. In both of these distributions, we first sample a hidden partition $X^* \in \{0,1\}^*$ and then sample T independent graphs from $G(n, 2\beta/n)$ where the edge-vertex incidence matrices of these graphs are denoted as $\{M_t\}_{t\in[T]}$. In the **YES** distribution, $w_t = M_t X^*$ and in the **NO** distribution, $w_t = \mathbf{1} - M_t X^*$. We will abuse notation and call the corresponding distributions of the subgraph H as **YES** and **NO** respectively. We summarize the properties of these distributions in the following lemma.

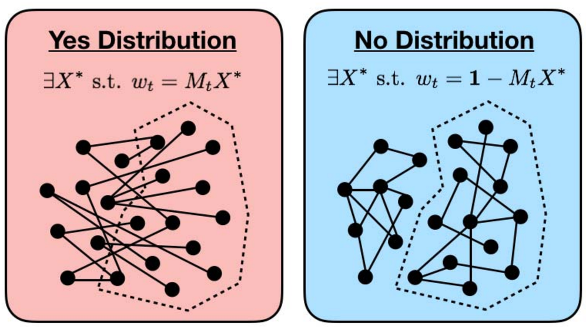

Figure 2. For a random graph on vertex set $[n]$, we partition the edges into two sets: (i) edges that lie across X^* and $\overline{X^*}$ and (ii) edges that lie in X^* or $\overline{X^*}$. In the **YES** distribution, only the (i) type edges are present in H. In the **NO** distribution, only the (ii) type edges are present in H.

Lemma 28 (Graphical view of DBHP)**.** *For any $n \in \mathbb{N}$ large enough and $\epsilon \in (0, 0.25)$, let $T = (10000/\epsilon^2)^3 \cdot (10c)^2$ and $\beta = \frac{1}{(10cT)^{2/3}}$ such that $\beta T = 10000/\epsilon^2$. Let **YES** and **NO** be the distributions of the subgraph H induced from DBHP(n, β, T) as described above, and let m_{DBHP} denote the total number of edges in H. For every $X^*, \sigma \in \{0,1\}^n$, define $m_{cross}(\sigma)$ to be the number of edges (i, j) such that (i) $\sigma(x_i) \neq \sigma(x_j)$ and (ii) $X^*_i = X^*_j$. We have the following.*

- *(Size of X^*) For each distribution **YES**, **NO** and for any constant $\epsilon' \in (0,1)$ such that $\epsilon' \geq \epsilon/10$, we have*

$$\Pr\left[\left||X^*| - \frac{n}{2}\right| > \epsilon' \cdot n\right] = o(1)\,.$$

- *(Number of edges) For each distribution **YES**, **NO** and for any constant $\epsilon' \in (0,1)$ such that $\epsilon' \geq \epsilon/10$, we have*

$$\Pr\left[\left|m_{DBHP} - \frac{\beta n T}{2}\right| > \epsilon' \cdot \beta n T\right] = o(1)\,.$$

- *(**NO** distribution) For any constant $\epsilon' \in (0,1)$ such that $\epsilon' \geq \epsilon/10$, we have*

$$\Pr_{\textbf{NO}}\left[\exists \sigma \in \{0,1\},\ m_{cross}(\sigma) > M\right] = o(1)\,,$$

where

$$M = \left(\frac{|\sigma \cap X^*| \cdot |\overline{\sigma} \cap X^*| + |\sigma \cap \overline{X^*}| \cdot |\overline{\sigma} \cap \overline{X^*}|}{n^2}\right) \cdot 2\beta n T + \epsilon' \cdot \beta n T\,.$$

Open Questions

Our work gives optimal approximation ratios for all Boolean maximum constraint satisfaction problems with constraints of length at most two. It would be interesting to understand the complexity of constraint languages with *arity greater than two*, and *larger alphabet* sizes.

In terms of lower bounds, we show that better than $\frac{4}{9}$- and $\frac{\sqrt{2}}{2}$-approximations for Max-2-AND and Max-2-OR require space $\Omega(\sqrt{n})$. Can we improve these space lower bounds to $\Omega(n)$, matching the space requirements of standard algorithms that give $1 - \epsilon$ approximation?

Acknowledgment

We thank Madhu Sudan for fruitful discussions and for pointing out a mistake in an old proof, we are also grateful to the anonymous reviewers for their helpful comments.

The work of CC is supported by NSF awards CCF 1565264 and CNS 1618026, the work of AG is partially supported by a Rabin Fellowship, the work of SV is supported in part by a Simons Investigator Award and NSF Award CCF 1715187.

References

[1] C. Chou, A. Golovnev, and S. Velusamy, "Optimal streaming approximations for all Boolean Max-2CSPs," *CoRR*, vol. abs/2004.11796, 2020.

[2] N. Creignou and H. Vollmer, "Boolean constraint satisfaction problems: When does Post's lattice help?" in *Complexity of Constraints*. Springer, 2008, pp. 3–37.

[3] T. J. Schaefer, "The complexity of satisfiability problems," in *STOC 1978*. ACM, 1978, pp. 216–226.

[4] J. Thapper and S. Živný, "The complexity of finite-valued CSPs," *J. ACM*, vol. 63, no. 4, pp. 1–33, 2016.

[5] K. Makarychev and Y. Makarychev, "Approximation algorithms for CSPs," in *The Constraint Satisfaction Problem: Complexity and Approximability*. LIPIcs, 2017, vol. 7, pp. 287–325.

[6] M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *J. ACM*, vol. 42, no. 6, pp. 1115–1145, 1995.

[7] U. Feige and M. Goemans, "Approximating the value of two prover proof systems, with applications to MAX 2SAT and MAX DICUT," in *ISTCS 1995*. IEEE, 1995, pp. 182–189.

[8] U. Zwick, “Analyzing the MAX 2-SAT and MAX DI-CUT approximation algorithms of Feige and Goemans,” *Unpublished manuscript*, 2000.

[9] M. Lewin, D. Livnat, and U. Zwick, “Improved rounding techniques for the MAX 2-SAT and MAX DI-CUT problems,” in *IPCO 2002*. Springer, 2002, pp. 67–82.

[10] J. Håstad, “Some optimal inapproximability results,” *J. ACM*, vol. 48, no. 4, pp. 798–859, 2001.

[11] L. Trevisan, G. B. Sorkin, M. Sudan, and D. P. Williamson, “Gadgets, approximation, and linear programming,” *SIAM J. Comput.*, vol. 29, no. 6, pp. 2074–2097, 2000.

[12] P. Austrin, “Towards sharp inapproximability for any 2-CSP,” *SIAM J. Comput.*, vol. 39, no. 6, pp. 2430–2463, 2010.

[13] D. Kogan and R. Krauthgamer, “Sketching cuts in graphs and hypergraphs,” in *ITCS 2015*. ACM, 2015, pp. 367–376.

[14] M. Kapralov, S. Khanna, and M. Sudan, “Streaming lower bounds for approximating MAX-CUT,” in *SODA 2015*. SIAM, 2015, pp. 1263–1282.

[15] M. Kapralov, S. Khanna, M. Sudan, and A. Velingker, “$(1 + \omega(1))$-approximation to MAX-CUT requires linear space,” in *SODA 2017*. SIAM, 2017, pp. 1703–1722.

[16] V. Guruswami, A. Velingker, and S. Velusamy, “Streaming complexity of approximating Max 2CSP and Max Acyclic Subgraph,” in *APPROX 2017*. LIPIcs, 2017.

[17] M. Kapralov and D. Krachun, “An optimal space lower bound for approximating MAX-CUT,” in *STOC 2019*. ACM, 2019, pp. 277–288.

[18] V. Guruswami and R. Tao, “Streaming hardness of unique games,” in *APPROX 2019*. LIPIcs, 2019, pp. 5:1–5:12.

[19] A. McGregor, “Graph stream algorithms: a survey,” *SIGMOD Record*, vol. 43, no. 1, pp. 9–20, 2014.

[20] A. Chakrabarti, “Data stream algorithms,” *Lecture notes*, p. 79, 2015. [Online]. Available: https://www.cs.dartmouth.edu/ ac/Teach/CS49-Fall11/Notes/lecnotes.pdf

[21] N. Alon, Y. Matias, and M. Szegedy, “The space complexity of approximating the frequency moments,” *J. Comput. Syst. Sci.*, vol. 58, no. 1, pp. 137–147, 1999.

[22] M. R. Henzinger, P. Raghavan, and S. Rajagopalan, “Computing on data streams,” *External memory algorithms*, vol. 50, pp. 107–118, 1998.

[23] M. Zelke, “Intractability of Min- and Max-Cut in streaming graphs,” *Inf. Process. Lett.*, vol. 111, no. 3, pp. 145–150, 2011.

[24] K. Lieberherr and E. Specker, “Complexity of partial satisfaction,” in *SFCS 1979*. IEEE, 1979, pp. 132–139.

[25] P. Indyk, “Stable distributions, pseudorandom generators, embeddings and data stream computation,” in *FOCS 2000*. IEEE, 2000, pp. 189–197.

[26] D. M. Kane, J. Nelson, and D. P. Woodruff, “On the exact space complexity of sketching and streaming small norms,” in *SODA 2010*. SIAM, 2010, pp. 1161–1178.

[27] S. Assadi, S. Khanna, and Y. Li, “Tight bounds for single-pass streaming complexity of the set cover problem,” in *STOC 2016*, 2016, pp. 698–711.

[28] M. Yannakakis, “On the approximation of maximum satisfiability,” *J. Algorithms*, vol. 17, no. 3, pp. 475–502, 1994.

[29] M. X. Goemans and D. P. Williamson, “New $3/4$-approximation algorithms for the maximum satisfiability problem,” *SIAM J. Discrete Math.*, vol. 7, no. 4, pp. 656–666, 1994.

[30] J. Håstad, “Every 2-CSP allows nontrivial approximation,” *Comput. Complex.*, vol. 17, no. 4, pp. 549–566, 2008.

[31] P. Raghavendra, “Optimal algorithms and inapproximability results for every CSP?” in *STOC 2008*, 2008, pp. 245–254.

[32] P. Raghavendra and D. Steurer, “Graph expansion and the unique games conjecture,” in *STOC 2010*, 2010, pp. 755–764.

[33] D. P. Williamson, “Lecture notes on approximation algorithms,” Technical Report RC–21409, IBM, Tech. Rep., 1999.

[34] M. Poloczek and G. Schnitger, “Randomized variants of Johnson’s algorithm for MAX SAT,” in *SODA 2011*. SIAM, 2011, pp. 656–663.

[35] M. Poloczek, “Bounds on greedy algorithms for MAX SAT,” in *ESA 2011*. Springer, 2011, pp. 37–48.

[36] A. Van Zuylen, “Simpler $3/4$-approximation algorithms for MAX SAT,” in *WAOA 2011*. Springer, 2011, pp. 188–197.

[37] M. Poloczek, G. Schnitger, D. P. Williamson, and A. Van Zuylen, “Greedy algorithms for the maximum satisfiability problem: Simple algorithms and inapproximability bounds,” *SIAM J. Comput.*, vol. 46, no. 3, pp. 1029–1061, 2017.

[38] S. Khot and N. K. Vishnoi, “On the unique games conjecture,” in *FOCS 2005*, vol. 5, 2005, p. 3.

[39] S. Khot, G. Kindler, E. Mossel, and R. O’Donnell, “Optimal inapproximability results for MAX-CUT and other 2-variable CSPs?” *SIAM J. Comput.*, vol. 37, no. 1, pp. 319–357, 2007.

[40] D. Gavinsky, J. Kempe, I. Kerenidis, R. Raz, and R. De Wolf, “Exponential separations for one-way quantum communication complexity, with applications to cryptography,” in *STOC 2007*, 2007, pp. 516–525.

[41] E. Verbin and W. Yu, “The streaming complexity of cycle counting, sorting by reversals, and other problems,” in *SODA 2011*. SIAM, 2011, pp. 11–25.

[42] R. Morris, “Counting large numbers of events in small registers,” *Commun. ACM*, vol. 21, no. 10, pp. 840–842, 1978.

[43] A. Gronemeier and M. Sauerhoff, “Applying approximate counting for computing the frequency moments of long data streams,” *Theory Comput. Syst.*, vol. 44, no. 3, pp. 332–348, 2009.