Electronic Colloquium on Computational Complexity, Report No. 166 (2022)

Characterizing the Multi-Pass Streaming Complexity for Solving Boolean CSPs Exactly

Gillat Kol*
Princeton University

Dmitry Paramonov†
Princeton University

Raghuvansh R. Saxena‡
Microsoft

Huacheng Yu§
Princeton University

Abstract

We study boolean *constraint satisfaction problems* (CSPs) $\mathsf{Max\text{-}CSP}_n^f$ for all predicates $f : \{0,1\}^k \to \{0,1\}$. In these problems, given an integer v and a list of constraints over n boolean variables, each obtained by applying f to a sequence of literals, we wish to decide if there is an assignment to the variables that satisfies at least v constraints. We consider these problems in the *streaming* model, where the algorithm makes a small number of passes over the list of constraints.

Our first and main result is the following complete characterization: For every predicate f, the streaming space complexity of the $\mathsf{Max\text{-}CSP}_n^f$ problem is $\tilde{\Theta}(n^{\mathsf{deg}(f)})$, where $\mathsf{deg}(f)$ is the degree of f when viewed as a multilinear polynomial. While the upper bound is obtained by a (very simple) one-pass streaming algorithm, our lower bound shows that a better space complexity is impossible even with *constant-pass* streaming algorithms.

Building on our techniques, we are also able to get an optimal $\Omega(n^2)$ lower bound on the space complexity of constant-pass streaming algorithms for the well studied $\mathsf{Max\text{-}CUT}$ problem, even though it is not technically a $\mathsf{Max\text{-}CSP}_n^f$ problem as, *e.g.*, negations of variables and repeated constraints are not allowed.

*gillat.kol@gmail.com
†dp20@princeton.edu
‡raghuvansh.saxena@gmail.com
§yuhch123@gmail.com

ISSN 1433-8092

Contents

1 Introduction

Constraint satisfaction problems (CSPs) are used extensively in mathematics as they give a unified framework that allows the expression of a wide variety of computational optimization problems. An instance of a (boolean) CSP is a list of *constraints* (or *clauses*) $\Psi = (\mathbf{C}_1, \ldots, \mathbf{C}_m)$ over n boolean variables $x_1, \ldots, x_n$. Here, each constraint $\mathbf{C}_i$ is obtained by applying a boolean function to a sequence of variables. The value of Ψ is the maximum number of constraints that can be satisfied by an assignment to the variables.

CSPs received a lot of attention in the computational setting, where the holy grail is to classify all CSPs according to their hardness. A surprising classical result from the 1970's, known as the *dichotomy theorem*, shows that the problem of deciding if all the constraints of a given CSP can be satisfied is either in P or is NP-complete [Sch78, FV98, Bul17, Zhu20]. Another very successful line of research studies the hardness of *approximating* the value of a CSP instance (or, equivalently, solving the corresponding gap problems), culminating in a complete characterization of "approximation-resistant" CSPs, at least under the unique games conjecture [Rag08] (also see [Mos10, Aus07, Aus10] and the survey of [Kho10]).

The space complexity required to solve general CSPs was only recently studied in the context of *streaming algorithms* [GT19, CGV20, CGSV21, CGS+22, BHP+21, SSV21]. Streaming algorithms are a restricted set of algorithms where the input is assumed to be given as a stream of objects that is only scanned once or a few times by the algorithm. In the framework of streaming CSPs, the objects in the stream are constraints (with repeated constraints allowed).

Recently, [CGSV21] showed that CSPs are never very easy in the streaming setting. In particular, they give a simple argument showing an $\Omega(n)$ lower bound on the space complexity of any streaming algorithm that solves $\mathsf{Max\text{-}CSP}^f_n$, for any non-constant f. Here, $\mathsf{Max\text{-}CSP}^f_n$ is the problem where on input (Ψ, v), we need to *decide* whether or not the value of Ψ is at least v, where $v \in \mathbb{N}$ and Ψ is a CSP instance over n variables with constraints that are applications of the predicate $f : \{0,1\}^k \to \{0,1\}$ to a sequence of literals (variables and negations of variables) and constants[1][2].

Are there $\mathsf{Max\text{-}CSP}$ problems that require substantially more than linear space? We mention that for other streaming problems where the size of the input is potentially much larger than n, *e.g.*, graph streaming problems, linear or almost linear space algorithms are often considered efficient ("semi-streaming"), and $\Omega(n^2)$ lower bounds are desired[3].

This paper. In this paper we give a *characterization* of the space complexity of *multi-pass* streaming algorithms that solve $\mathsf{Max\text{-}CSP}^f_n$, for arbitrary f. For the rest of this section,

[1] *E.g.*, the constraint $\mathbf{C}_i$ can be $f(1, \bar{x}_5, \bar{x}_8, x_2)$.

[2] We mention that the setting of [CGSV21] is more general: it does not allow the constraints to use negation of variables, but does allow them to apply any predicate out of a set of predicates $\mathcal{F}$.

[3] For instance, an $\Omega(n)$ multi-pass lower bound for directed reachability and related graph problems is simple, and recent work focused on improving the bound to $\Omega(n^{2-\epsilon})$ [GO16, AR20, CKP+21].

assume that the length of the stream is at most polynomial in n. It is easy to see that for every f, the $\mathsf{Max\text{-}CSP}_n^f$ problem can be solved by a one-pass streaming algorithm with at most $\tilde{O}(n^k)$ space: Observe that the number of different constraints is only $O(n^k)$.[4] By counting the number of appearances of each clause in the stream, which only requires storing $O(n^k)$ counters, we essentially store the entire input and can even compute the exact value of the instance.

Is $\Omega(n^k)$ space always required? Clearly no, as f may not even depend on all k of its variables. So, what exactly determines the space complexity of $\mathsf{Max\text{-}CSP}_n^f$?

1.1 Our Results

We start by observing that, in fact, the $\mathsf{Max\text{-}CSP}_n^f$ problem admits an $\tilde{O}(n^d)$-space, one-pass streaming algorithm, where $d = \mathsf{deg}(f) \leq k$ is the *degree* of f when written as a multilinear polynomial *over the reals*[5]. This follows because, for any instance Ψ with n variables, there exists a degree d polynomial P over the same variables such that the values of Ψ and P on any assignment $\mathbf{x} \in \{0,1\}^n$ are the same. Moreover, this polynomial can easily be maintained using an $\tilde{O}(n^d)$-space streaming algorithm, as it has at most $O(n^d)$ coefficients and is just the sum of the multilinear polynomials corresponding to each individual clause[6]. Thus, an algorithm that maintains this polynomial using $\tilde{O}(n^d)$-space and outputs its largest value (over all $\mathbf{x}$) also solves $\mathsf{Max\text{-}CSP}_n^f$.

However, is there yet another, better, streaming algorithm for $\mathsf{Max\text{-}CSP}_n^f$, for any f?

1.1.1 Lower Bounds for Max-CSP

Our main result answers this question in the negative, showing that the above algorithm is essentially optimal, even if constantly many passes are allowed. This means that the degree of a predicate fully characterizes the streaming space complexity of the associated Max-CSP problem.

Theorem 1.1 (*cf.* Theorem 4.1)**.** *Let $k \in \mathbb{N}$ be a constant and let $f : \{0,1\}^k \to \{0,1\}$. For $n, p \in \mathbb{N}$, the p-pass streaming space complexity of $\mathsf{Max\text{-}CSP}_n^f$ is at least $\Omega\big(n^{\mathsf{deg}(f)}/p\big)$.*

We mention that with 2^n passes, the space complexity of $\mathsf{Max\text{-}CSP}_n^f$ drops down to $\tilde{O}(\log n)$, for every f. The reason is that, in each pass, the algorithm can count the number of constraints satisfied by a certain assignment. We also mention that the formal version (see Theorem 4.1) of Theorem 1.1 shows a lower bound on the *communication complexity* of $\mathsf{Max\text{-}CSP}_n^f$, and is therefore stronger. The same holds for the stronger version (see Theorem 5.1) of Theorem 1.2 below.

[4] A constraint corresponds to an element of $\mathcal{X}^k$, where $\mathcal{X} = \{x_1, \bar{x}_1, \ldots, x_n, \bar{x}_n, 0, 1\}^k$, and $|\mathcal{X}| = 2n + 2$.

[5] For instance, if $f(y_1, y_2, y_3) = y_1 \wedge y_2 \wedge \bar{y}_3$, then the corresponding polynomial is $y_1 y_2 (1 - y_3)$.

[6] For instance, if $f(y_1, y_2, y_3) = y_1 y_2 (1 - y_3) = y_1 y_2 - y_1 y_2 y_3$ and $\mathbf{C}_i = f(\bar{x}_5, 1, x_2)$, then multilinear polynomial corresponding to $\mathbf{C}_i$ is $(1 - x_5) \cdot 1 \cdot (1 - x_2) = 1 - x_2 - x_5 - x_2 x_5$.

The proof of Theorem 1.1 consists of two key results. The first result, given in Theorem 3.1, shows that any instance of $\mathsf{Max\text{-}CSP}_n^{\mathsf{AND}_d}$, where AND_d is the d-bit conjunction function, can be expressed as a $\mathsf{Max\text{-}CSP}_n^f$ instance for any f that has $\mathsf{deg}(f) = d$.[7] Therefore, to prove Theorem 1.1, it suffices to show an $\Omega(n^d)$ lower bound on the streaming complexity of $\mathsf{Max\text{-}CSP}_n^{\mathsf{AND}_d}$, which is done by our second key result, Lemma 4.2. Lemma 4.2, in turn, is proved using a novel communication complexity reduction from set disjointness. We mention that our proofs are generally quite simple.

Theorem 3.1 may be of independent interest, as it gives a general way of converting lower bounds for $\mathsf{Max\text{-}CSP}_n^{\mathsf{AND}_d}$ to lower bounds for $\mathsf{Max\text{-}CSP}_n^f$. Indeed, in Appendix A, we show that it can also be used to obtain a lower bound on the space complexity of *multi-pass* streaming algorithms that *approximate* Max-CSP problems *arbitrarily well*. We mention that the space complexity of streaming and sketching algorithms that approximate, within *any constant factor*, the value of a given CSP instance was the main interest of [CGSV21] (also see [CGS+22]), and that they prove beautiful dichotomy (or partial dichotomy) results.[8] See [Sud22] for a recent and great survey.

1.1.2 Lower Bound for Max-CUT

One of the most studied CSPs in the streaming literature is the Max-CUT problem, corresponding to the XOR predicate [KK15, KKS15, KKSV17, BDV18, KK19, AKSY20, AV21]. Note that $\mathsf{Max\text{-}CUT}_n$ is not a proper $\mathsf{Max\text{-}CSP}_n^f$ problem, as constraints cannot be repeated nor use constants or negations of variables. Nevertheless, our techniques can be used to prove an $\Omega(n^2)$ lower bound on the space complexity of multi-pass $\mathsf{Max\text{-}CUT}_n$ streaming algorithms for *unweighted graphs*. Observe that, indeed, $\mathsf{deg}(\mathsf{XOR}) = 2$.

Theorem 1.2 (*cf.* Theorem 5.1)**.** *For $n, p \in \mathbb{N}$, the p-pass streaming space complexity of $\mathsf{Max\text{-}CUT}_n$ is at least $\Omega(n^2/p)$.*

Prior to our work, an $\Omega(n^2)$ lower bound was only known for one-pass streaming algorithms that solve $\mathsf{Max\text{-}CUT}_n$ [Zel11] and for weighted graphs [BCHD+19][9]. Multi-pass streaming lower bounds were recently shown for the much more general case of approximation algorithms, but these only obtained sub-linear lower bounds on the space [AKSY20, AV21]. Quadratic multi-pass lower bounds for other graph problems are shown in [ACHKP21].

[7]Reductions of this form were used in the study of CSPs in the computational setting. For instance, the XOR of two bits can be expressed using a set of f-clauses, for many different functions f, see *e.g.* Lemma 5.36 in [CKS01]. However, such reductions do not preserve the degree (reducing it to 2), and would not give us better than quadratic bounds. Indeed, our proofs are very different from theirs and preserve the degree of f.

[8]We note that the space regime in their dichotomies is different than the one we consider in Theorem 4.1: As the value of any CSP instance can be approximated within any constant factor by a one-pass $\tilde{O}(n)$-space streaming algorithm, an "easy" CSP for [CGSV21] admits an $O(\mathrm{poly}\log n)$-space one-pass streaming algorithm, and a "hard" CSP requires $\Omega(n^\alpha)$ space ($\alpha \leq 1$), also see [CGS+22]. In the exact version, however, an $\Omega(n)$ lower bound is known [CGSV21] and so, our main result (Theorem 4.1) concerns super-linear space complexities.

[9]We thank the anonymous reviewer for telling us that this theorem follows from [BCHD+19].

2 Models and Preliminaries

2.1 Notation

We use $\mathbb{N} = \{1, 2, 3, \ldots\}$ to denote the set of natural numbers (note that $0 \notin \mathbb{N}$). We denote vectors in bold letters (*e.g.*, $\mathbf{x}$ and $\mathbf{C}$). Let $\ell \geq 1$ and let $\mathbf{x}$ be a vector with ℓ coordinates. For $i \in [\ell]$, we use the notation x_i to address coordinate i of $\mathbf{x}$. Let $S \subseteq [\ell]$, we use the notation $\mathbf{x}_S$ to address the vector with $|S|$ coordinates obtained by deleting from $\mathbf{x}$ coordinates that are not in S. We often use the notation $(\cdot, \cdot)$ to denote vector concatenation, *e.g.*, if each of $\mathbf{x}$ and $\mathbf{y}$ is either a vector or an element, then $(\mathbf{x}, \mathbf{y})$ denotes the vector obtained by concatenating $\mathbf{y}$ to $\mathbf{x}$.

Let $\ell \geq 0$. We use $\mathbf{0}^\ell$ and $\mathbf{1}^\ell$ to denote the all-0s and all-1s vectors (respectively) of ℓ coordinates. For a vector $\mathbf{x} \in \{0,1\}^\ell$, we denote the Hamming weight of $\mathbf{x}$ by $\|\mathbf{x}\|$. That is, $\|\mathbf{x}\| = \sum_{i \in [\ell]} x_i$.

2.2 Constraint Satisfaction Problems

CSPs. Let $k \in \mathbb{N}$ be a natural number and $f : \{0,1\}^k \to \{0,1\}$ be a boolean function. Let $n \in \mathbb{N}$ and consider n boolean variables $x_1, \ldots, x_n$. Let $\mathcal{X}_n = \{0, 1, x_1, \bar{x}_1, \ldots, x_n, \bar{x}_n\}$ be the set of all literals and constants. An instance of the $\mathsf{Max\text{-}CSP}_n^f$ problem is defined as a list of clauses $\Psi = (\mathbf{C}_1, \ldots, \mathbf{C}_m)$, for some $m \in \mathbb{N}$, where $\mathbf{C}_i \in \mathcal{X}_n^k$ for all $i \in [m]$.

Observe that if $\mathbf{C} \in \mathcal{X}_n^k$, then an assignment $\mathbf{x} \in \{0,1\}^n$, fixes the value of $f(\mathbf{C}_i)$. We define the value of Ψ on an assignment $\mathbf{x} \in \{0,1\}^n$ to be the number of clauses that it satisfies:

$$\Psi(\mathbf{x}) = \sum_{i=1}^{m} f(\mathbf{C}_i).$$

The value of Ψ is defined as the maximum number of clauses that are satisfied by a single assignment:

$$\mathsf{Max\text{-}CSP}_n^f(\Psi) = \max_{\mathbf{x} \in \{0,1\}^n} \Psi(\mathbf{x}). \tag{1}$$

The problem of $\mathsf{Max\text{-}CSP}_n^f$ is a decision problem that on input (Ψ, v), where Ψ is as above and $v \in \mathbb{N}$, outputs 1 if $\mathsf{Max\text{-}CSP}_n^f(\Psi) \geq v$ and 0 otherwise.

Approximate CSPs. We will also be interested in the approximation version of $\mathsf{Max\text{-}CSP}_n^f$. For $\epsilon \geq 0$, the problem of $\mathsf{Max\text{-}CSP}_{n,\epsilon}^f$ on instance Ψ is to output a value v that satisfies

$$(1 - \epsilon) \cdot \mathsf{Max\text{-}CSP}_n^f(\Psi) \leq v \leq \mathsf{Max\text{-}CSP}_n^f(\Psi). \tag{2}$$

Positive CSPs. It will be useful to consider CSPs with a restricted set of possible clauses, where variables are only used positively (meaning that the negations of variables cannot be used). Formally, as before, we define an instance of the $\mathsf{Max\text{-}Pos\text{-}CSP}_n^f$ problem as a list of clauses $\Psi = (\mathbf{C}_1, \ldots, \mathbf{C}_m)$. However, now each $\mathbf{C}_i$ is in the set $\{0, 1, x_1, \ldots, x_n\}^k$.[10]

Predicate degree. Let $k \in \mathbb{N}$ and $f : \{0,1\}^k \to \{0,1\}$ be a boolean function. We define the degree of f, denoted $\mathsf{deg}(f)$, to be the minimum degree of a (multilinear) polynomial $g : \mathbb{R}^k \to \mathbb{R}$ that satisfies $\forall x \in \{0,1\}^k : \ f(x) = g(x)$. We mention that it can be assumed, without loss of generality, that the coefficients of g are integers. Indeed, if not, fixing the smallest degree term with a non-integer coefficient and setting all the variables in this term to 1 and all other variables to 0 results in a non-integral value.

Max-AND. Let $k \in \mathbb{N}$. We denote $\mathsf{AND}_k(x_1, \ldots, x_k) = \bigwedge_{i \in [k]} x_i$. We use $\mathsf{Max\text{-}AND}_n^k$ to denote the $\mathsf{Max\text{-}CSP}_n^{\mathsf{AND}_k}$ problem.

Max-CUT. Let $n \in \mathbb{N}$ and consider a *simple*, undirected graph G on n vertices. We define $\mathsf{Max\text{-}CUT}_n(G)$ to be the maximum size of a cut (partitioning of the vertices) in G. Here, the size of a cut is the number of edges in G that cross the cut. Let $v \in \mathbb{N}$. We define $\mathsf{Max\text{-}CUT}_n(G, v) = 1$ if $\mathsf{Max\text{-}CUT}_n(G) \geq v$, and otherwise $\mathsf{Max\text{-}CUT}_n(G, v) = 0$.

2.3 Communication Complexity

For a two-party communication task $T(x, y)$, we use $\mathsf{CC}(T)$ to denote the randomized communication complexity of T with success probability at least $2/3$.

Max-CSP as a communication task. We denote by $\mathsf{CC}(\mathsf{Max\text{-}CSP}_n^f)$ the communication complexity of solving $\mathsf{Max\text{-}CSP}_n^f$ instances where the clauses are partitioned between two parties. Formally, the input to the communication task is $(\Psi, v) = \big((\Psi^A, \Psi^B), v\big)$, where Alice gets as input Ψ^A and Bob gets as input Ψ^B, and v is known to both parties. We will assume throughout that Ψ^A and Ψ^B are of the same size. This technical assumption will be useful for us as it implies that both Alice and Bob know the total number of clauses. We define $\mathsf{CC}(\mathsf{Max\text{-}Pos\text{-}CSP}_n^f)$ similarly.

Max-CUT as a communication task. We denote by $\mathsf{CC}(\mathsf{Max\text{-}CUT}_n)$ the communication complexity of solving $\mathsf{Max\text{-}CUT}_n$ instances where the edges of the graph are partitioned between two parties. Formally, there is a set V of n vertices and both Alice and Bob are given disjoint sets of edges E_A and E_B over the vertices in V. Both of them also know a value v and need to determine whether or not the maximum cut in the graph $G = (V, E_A \cup E_B)$ is at least v.

[10] We note that we do want to allow constants: consider, for example, the case where $f(x_1, x_2, x_3) = x_1 \oplus x_2 \oplus x_3$. When not allowing constants, any instance of $\mathsf{Max\text{-}Pos\text{-}CSP}_n^f$ is trivially maximized by the all-1s vector.

Set disjointness. We will use a lower bound on the communication complexity of the following version of the *set disjointness problem*: For $u, m \in \mathbb{N}$ with $u \geq m$, an instance of the $\mathsf{DISJ}_{u,m}$ problem is a pair $(\mathbf{y}, \mathbf{z})$, where $\mathbf{y}, \mathbf{z} \in \{0,1\}^u$ with $\|\mathbf{y}\| = \|\mathbf{z}\| = m$. The problem is to compute whether or not the sets indicated by $\mathbf{y}$ and $\mathbf{z}$ intersect or not, *i.e.*, $\mathsf{DISJ}_{u,m}(\mathbf{y}, \mathbf{z}) = \mathbb{1}(\forall i \in [u] : y_i \cdot z_i = 0)$.

Lemma 2.1 ([Raz90])**.** *Let $m \in \mathbb{N}$. We have that $\mathsf{CC}(\mathsf{DISJ}_{4m+1,m}) \geq \Omega(m)$.*

2.4 Streaming Algorithms

We say that p-pass streaming algorithm solves a streaming task if it scans the input p times and outputs a correct solution with probability at least 2/3. The problems Max-CSP, Max-CUT have a natural streaming task associated with where the list of clauses/edges are given in a stream and the target value v is hard-coded in the algorithm.

3 Reducing Max-AND to Max-CSP

The goal of this section is to show the following theorem:

Theorem 3.1. *Let $k \geq d \in \mathbb{N}$. Let $f : \{0,1\}^k \to \{0,1\}$ be such that $\mathsf{deg}(f) = d$. There exist non-negative rational numbers $\{\alpha_{\mathbf{C}}\}_{\mathbf{C} \in \mathcal{X}_d^k}$ and α, such that for every $\mathbf{x} \in \{0,1\}^d$ it holds that*

$$\mathsf{AND}_d(\mathbf{x}) = \sum_{\mathbf{C} \in \mathcal{X}_d^k} \alpha_{\mathbf{C}} f(\mathbf{C}) - \alpha.$$

Proof. To start, note that the non-negativity of α is without loss of generality (given the other claims), as can be seen by setting $\mathbf{x} = \mathbf{0}^d$. We use the following notation: Given a function $j : \{0,1\}^\ell \to \{0,1\}$, we write it as the polynomial $j(\mathbf{x}) = \sum_{S \subseteq [\ell]} j_S T_S$, where $T_S = \prod_{i \in S} x_i$.

Let $S \subseteq [k]$ be a set of size d with $f_S \neq 0$. We assume without loss of generality that $S = [d]$. We define the function $h : \{0,1\}^d \to \{0,1\}$ by $h(\mathbf{x}) = f(\mathbf{x}, \mathbf{0}^{k-d})$ if $f_{[d]} > 0$, and by $f(\bar{x}_1, x_2, \ldots, x_d, \mathbf{0}^{k-d})$ if $f_{[d]} < 0$. Observe that $h_{[d]} = |f_{[d]}| > 0$.

If h is of the form $h(\mathbf{x}) = h_{[d]} \cdot T_{[d]} + h_\emptyset$, we are done, as this implies $T_{[d]} = \mathsf{AND}_d(\mathbf{x}) = \frac{1}{h_{[d]}}(h(\mathbf{x}) - h_\emptyset)$ and as $h_{[d]} > 0$ (also recall that, for every $S \subseteq [d]$, the coefficient h_S can be assumed to be an integer). Otherwise, let $0 < d^* < d$ be the maximum size of a set S such that $h_S \neq 0$, and assume without loss of generality that $h_{[d^*]} \neq 0$.

Let $h', g : \{0,1\}^d \to \{0,1\}$ be given by $h'(\mathbf{x}) = h(\bar{x}_1, x_2, \ldots, x_{d^*}, \mathbf{0}^{d-d^*})$ and $g(\mathbf{x}) = h(\mathbf{x}) + h'(\mathbf{x})$. We next prove the following three properties about the coefficients of g:

1. $g_{[d]} = h_{[d]} > 0$.

2. $g_{[d^*]} = 0$.

3. Let $\mathcal{S}$ be the set of subsets $S \subsetneq [d]$ with $|S| \geq d^*$ and $h_S = 0$. Then, for every $S \in \mathcal{S}$, it holds that $g_S = 0$.

Before proving the above three properties, we show that they suffice in order to prove the theorem. We use the following observation that is implied by the second and third properties: Recall that d^* is the maximum size of a set $S \subsetneq [d]$ with $h_S \neq 0$, and let t be the number of sets $S \subsetneq [d]$ of size d^* with $h_S \neq 0$. Then, either the maximum size of a set $S \subsetneq [d]$ with $g_S \neq 0$ is strictly smaller than d^*, or the maximum size of such set is d^* but there are strictly less than t sets S of size d^* with $g_S \neq 0$.

The theorem follows from the observation by repeatedly "zeroing out" a leading coefficient. In more detail, consider the sequence of functions $h^1, h^2, \dots,$ where $h^1 = h$ and $h^{i+1} = h^i + (h^i)'$,[11] and where the sequence ends after the function h^m if and only if it is of the form $h^m(\mathbf{x}) = h^m_{[d]} \cdot T_{[d]} + h^m_{\emptyset}$. By the observation, the sequence indeed ends. Let h^m be the last function in the sequence. Observe that h^m is of the form $\sum_{\mathbf{C} \in \mathcal{X}^k_d} \alpha'_{\mathbf{C}} f(\mathbf{C})$ with the coefficients $\alpha'_{\mathbf{C}}$ being non-negative integers, and that, by the first property, $h^m_{[d]} > 0$. This concludes the proof as we have $T_{[d]} = \mathsf{AND}_d(\mathbf{x}) = \frac{1}{h^m_{[d]}}(h^m(\mathbf{x}) - h^m_{\emptyset})$.

It remains to prove the three above properties. We first calculate the coefficients of h':

$$\begin{aligned} h'(\mathbf{x}) = h(\bar{x}_1, x_2, \dots, x_{d^*}, \mathbf{0}^{d-d^*}) &= \sum_{S \subseteq \{2,\dots,d^*\}} h_S T_S + h_{S\cup\{1\}}(1-x_1)T_S \\ &= \sum_{S \subseteq [d^*]:\, 1 \notin S} (h_S + h_{S\cup\{1\}})T_S - \sum_{S \subseteq [d^*]:\, 1 \in S} h_S T_S. \end{aligned}$$

Therefore, for $S \subseteq [d^*]$, if $1 \in S$ then $h'_S = -h_S$, and if $1 \notin S$, then $h'_S = h_S + h_{S\cup\{1\}}$. Observe that if S is not a subset of $[d^*]$, it holds that $h'_S = 0$, and therefore $g_S = h_S + h'_S = h_S + 0 = h_S$. Since $d^* < d$, this implies $g_{[d]} = h_{[d]}$, proving the first property.

To prove the second property, note that for any set $S \subseteq [d^*]$ with $1 \in S$, we have $g_S = h_S + h'_S = h_S - h'_S = 0$. This implies $g_{[d^*]} = 0$.

To prove the third property, let $S \in \mathcal{S}$. Recall $h_S = 0$, and thus $S \neq [d^*]$. Also recall that $|S| \geq d^*$, and since $S \neq [d^*]$, this means that S is not contained in $[d^*]$. By the above, $g_S = h_S = 0$. $\square$

Our proofs use the following corollaries of Theorem 3.1 to communication complexity and streaming space complexity.

Corollary 3.2. *Let $k \in \mathbb{N}$ and $f : \{0,1\}^k \to \{0,1\}$. For all $n \in \mathbb{N}$, we have:*

$$\mathsf{CC}\Big(\mathsf{Max\text{-}AND}_n^{\mathsf{deg}(f)}\Big) \leq \mathsf{CC}\big(\mathsf{Max\text{-}CSP}_n^f\big).$$

Proof. Let $d = \mathsf{deg}(f)$. We prove the theorem by reduction. Given an input $(\Psi, v) = \big((\Psi^A, \Psi^B), v\big)$ for the $\mathsf{Max\text{-}AND}_n^d$ communication problem over variables $\mathbf{x} = (x_1, \dots, x_n)$,

[11] The function $(h^i)'$ is obtained from h^i by negating one of the variables. However, for a general i, the negated variable may not be x_1.

we construct an input $\Phi = \big((\Phi^A, \Phi^B), u\big)$ for $\mathsf{Max\text{-}CSP}^f_n$ over the same variables. To this end, we generate a set of f-clauses for every AND clause using Theorem 3.1.

In more detail, Alice goes over all the clauses in Ψ^A. Suppose that clause i is $\mathbf{C} \in \mathcal{X}^d_n$. Alice generates the f-clauses corresponding to this clause as follows: View $\mathbf{C}$ as the vector of formal variables $(X_1, \ldots, X_d)$ and let $\mathcal{X}'_d = \{0, 1, X_1, \bar{X}_1, \ldots, X_d, \bar{X}_d\}$ be the corresponding set of formal literals and constants. By Theorem 3.1, there exist $w_{\mathbf{C}'} \in \mathbb{N} \cup \{0\}$ for every $\mathbf{C}' \in (\mathcal{X}'_d)^k$, an integer w, and $w' \in \mathbb{N}$, such that

$$w' \cdot \mathsf{AND}_d(\mathbf{C}) = \sum_{\mathbf{C}' \in (\mathcal{X}'_d)^k} w_{\mathbf{C}'} f(\mathbf{C}') - w.$$

For every $\mathbf{C}' \in (\mathcal{X}'{}_d)^k$, Alice adds $w_{\mathbf{C}'}$ copies of the clause $\mathbf{C}'$ to Φ^A. Here, we view $\mathbf{C}'$ as an element in $\mathcal{X}^k_n$, as each of its coordinates X'_i, $i \in [k]$, is either a bit or is of the form X_j or $\bar{X}_j$ for some $j \in [d]$, and X_j itself is either a bit or of the form x_ℓ or $\bar{x}_\ell$ for some $\ell \in [n]$ (*e.g.*, if $X'_i = \bar{X}_j$ and $X_j = \bar{x}_\ell$, then we identify X'_i with $X'_i = \bar{X}_j = \overline{(\bar{x}_\ell)} = x_\ell$). Bob constructs Φ^B similarly.

Observe that both Alice and Bob generate the same number of f-clauses for every AND clause in Ψ. Since we assume that Ψ^A and Ψ^B has the same number of clauses, Φ^A and Φ^B have the same number of clauses. Let m be the number of clauses in Ψ, *i.e.*, m is the sum of the lengths of Ψ^A and Ψ^B. Observe that since Alice and Bob have the same number of clauses, they both know m. Also observe that

$$w' \cdot \mathsf{Max\text{-}AND}^{\mathsf{deg}(f)}_n(\Psi) = \mathsf{Max\text{-}CSP}^f_n(\Phi) - w \cdot m.$$

Now, set $u = w' \cdot v + w \cdot m$, and note that both Alice and Bob can compute u. To finish the proof we observe that $\mathsf{Max\text{-}CSP}^f_n(\Phi) \geq u$ if and only if $\mathsf{Max\text{-}AND}^{\mathsf{deg}(f)}_n(\Psi) \geq v$.

□

Corollary 3.3. *Let $k \in \mathbb{N}$ and $f : \{0,1\}^k \to \{0,1\}$. For all $\epsilon' \geq 0$, there exists $\epsilon \geq 0$ such that for all $p, n \in \mathbb{N}$, any p-pass streaming algorithm for $\mathsf{Max\text{-}CSP}^f_{n,\epsilon'}$ implies a p-pass streaming algorithm for $\mathsf{Max\text{-}AND}^{\mathsf{deg}(f)}_{n,\epsilon}$ with the same space complexity, up to constant factors.*

Proof. Let $d = \mathsf{deg}(f)$. Given an instance Ψ of $\mathsf{Max\text{-}AND}^d_n$ over variables $\mathbf{x} = (x_1, \ldots, x_n)$, presented as a stream of clauses, we can use the same construction as in the proof of Corollary 3.2 to generate an instance Φ of $\mathsf{Max\text{-}CSP}^f_n$ over the same variables. Note that this construction can be implemented in a streaming manner.

Let $w_{\mathbf{C}'} \in \mathbb{N} \cup \{0\}$ for every $\mathbf{C}' \in (\mathcal{X}'_d)^k$, $w \in \mathbb{N} \cup \{0\}$, and $w' \in \mathbb{N}$ be such as in the proof of Corollary 3.2, let $\alpha = \frac{w}{w'}$ be as in Theorem 3.1, and let m be the number of clauses in Ψ. Note that as before,

$$w' \cdot \mathsf{Max\text{-}AND}^d_n(\Psi) = \mathsf{Max\text{-}CSP}^f_n(\Phi) - w \cdot m.$$

Now, let $\epsilon = 2^{k+1}(\alpha+1)\epsilon'$, and suppose that there existed a p-pass streaming algorithm $\mathcal{A}'$ which, given an instance Φ of $\textsf{Max-CSP}^f_{n,\epsilon'}$ returned a value v' such that $(1-\epsilon')\cdot\textsf{Max-CSP}^f_n(\Phi) \le v' \le (1+\epsilon')\cdot\textsf{Max-CSP}^f_n(\Phi)$ with probability at least $2/3$. Then we could create a p-pass streaming algorithm $\mathcal{A}$ for $\textsf{Max-AND}^f_{n,\epsilon}$ which turns an instance Ψ of $\textsf{Max-AND}^d_n$ into an instance Φ of $\textsf{Max-CSP}^f_n$ as above, runs $\mathcal{A}'$ on the resulting stream Φ to obtain some v' as above, and then outputs $\frac{1}{w'}(v'-(1-\epsilon')\cdot w\cdot m)$.

Upper bounding the space complexity of $\mathcal{A}$: $\mathcal{A}$ requires only $O(\log m)$ additional bits over $\mathcal{A}'$ in order to compute m. However, without loss of generality, we may assume that every clause of Ψ is satisfied by some assignment of variables, by simply ignoring all clauses which are not satisfied by any assignment of variables. By a probabilistic argument, this then ensures that $\textsf{Max-AND}^d_n(\Psi) \ge m/2^k$, so $\log m = O(\log v')$. As $\mathcal{A}'$ has to output v', it requires at least $\log v'$ bits of memory, implying that the space required by $\mathcal{A}$ and $\mathcal{A}'$ are within a constant factor.

Proving the correctness of $\mathcal{A}$: Consider $v = \frac{1}{w'}(v'-(1-\epsilon')\cdot w\cdot m)$, the value returned by the algorithm. With probability $2/3$, it holds that $(1-\epsilon')\cdot\textsf{Max-CSP}^f_n(\Phi) \le v' \le (1+\epsilon')\cdot\textsf{Max-CSP}^f_n(\Phi)$. As such, suppose that this happens.

We claim that $v \ge (1-\epsilon)\cdot\textsf{Max-AND}^d_n(\Psi)$. This follows directly as $\epsilon \ge \epsilon'$, $v' \ge (1-\epsilon')\cdot\textsf{Max-CSP}^f_n(\Phi)$, and $w'\cdot\textsf{Max-AND}^d_n(\Psi) = \textsf{Max-CSP}^f_n(\Phi) - w\cdot m$.

Next, we claim that $v \le (1+\epsilon)\cdot\textsf{Max-AND}^d_n(\Psi)$ using the fact that $v' \le (1+\epsilon')\cdot\textsf{Max-CSP}^f_n(\Phi)$. Recalling our assumption that each clause in Ψ is satisfied by some assignment of variables, we get that $\textsf{Max-AND}^d_n(\Psi) \ge m/2^k$. Thus,

$$\begin{aligned}
v &= \frac{1}{w'}(v'-(1-\epsilon')\cdot w\cdot m)\\
&\le \frac{1}{w'}\big((1+\epsilon')\cdot\textsf{Max-CSP}^f_n(\Phi)-(1-\epsilon')\cdot w\cdot m\big)\\
&= (1+\epsilon')\cdot\textsf{Max-AND}^d_n(\Psi)+2\epsilon'\cdot\frac{wm}{w'}\\
&\le (1+\epsilon')\cdot\textsf{Max-AND}^d_n(\Psi)+2\epsilon'\cdot\alpha 2^k\cdot\textsf{Max-AND}^d_n(\Psi)\\
&= \big(1+\big(1+\alpha 2^{k+1}\big)\epsilon'\big)\cdot\textsf{Max-AND}^d_n(\Psi)\\
&\le (1+\epsilon)\cdot\textsf{Max-AND}^d_n(\Psi).
\end{aligned}$$

Thus, $(1-\epsilon)\cdot\textsf{Max-AND}^d_n(\Psi) \le v \le (1+\epsilon)\cdot\textsf{Max-AND}^d_n(\Psi)$ with probability at least $2/3$. □

4 Communication Lower Bound for Max-CSP

In this section we prove Theorem 1.1. By standard argument, Theorem 1.1 is implied by the following communication lower bound:

Theorem 4.1. *Let $k \in \mathbb{N}$ and $f : \{0,1\}^k \to \{0,1\}$. For all $n \in \mathbb{N}$, we have*

$$\mathsf{CC}(\mathsf{Max\text{-}CSP}_n^f) \geq \Omega\big(n^{\mathsf{deg}(f)}\big).$$

In turn, Theorem 4.1 follows directly from Lemma 4.2 below and Corollary 3.2:

Lemma 4.2. *Let $k \in \mathbb{N}$. For all $n \in \mathbb{N}$, we have*

$$\mathsf{CC}\big(\mathsf{Max\text{-}AND}_n^k\big) \geq \Omega\big(n^k\big).$$

Observe that Lemma 4.2 follows from an $\Omega\big(n^k\big)$ lower bound on $\mathsf{CC}(\mathsf{Max\text{-}CSP}_n^{g_k})$ for *any* function $g_k : \{0,1\}^k \to \{0,1\}$. The reason is that *any* g_k can be written in DNF form, by looking at its truth table and writing it as an OR of a set of AND clauses, such that any satisfying assignment satisfies exactly one of the AND clauses (and a non-satisfying assignment satisfies none). Now, given an instance of $\mathsf{Max\text{-}CSP}_n^{g_k}$, we convert it to an instance of $\mathsf{Max\text{-}AND}_n^k$ by replacing each constraint with the corresponding set of AND clauses. Observe that the values of the two instances are the same and therefore, a lower bound for $\mathsf{Max\text{-}CSP}_n^{g_k}$ implies a lower bound for $\mathsf{Max\text{-}AND}_n^k$. Thus, the following lemma implies Lemma 4.2:

Lemma 4.3. *Let $k \in \mathbb{N}$ and let $g_k(x_1, \ldots, x_k) = x_k \oplus \left(\bigvee_{i \in [k-1]} x_i\right)$. For all $n \in \mathbb{N}$, we have:*

$$\mathsf{CC}(\mathsf{Max\text{-}Pos\text{-}CSP}_n^{g_k}) \geq \Omega\big(n^k\big).$$

As mentioned above, a weaker version of Lemma 4.3, that shows a lower bound on the communication complexity of $\mathsf{Max\text{-}CSP}_n^{g_k}$ (instead of that of $\mathsf{Max\text{-}Pos\text{-}CSP}_n^{g_k}$) suffices to prove Lemma 4.2. Nevertheless, we chose to prove the stronger version as it can be shown to also imply Theorem 1.2 for *weighted* graphs, as $g_2(x_1, x_2) = \mathsf{XOR}(x_1, x_2)$, and that this is also part of the reason for selecting these specific g_k functions. In Section 5, we give an alternative proof that also works for unweighted graphs. The rest of this section is devoted to proving Lemma 4.3.

4.1 Proof of Lemma 4.3

In this section we prove Lemma 4.3. Fix $n, k \in \mathbb{N}$. Let U be the set of all subsets of $[n]$ of size exactly k and let $u = |U| = \binom{n}{k}$. When we take a set $S \in U$, we denote its elements by $s_1 < s_2 < \ldots < s_k$ and use the notation S_{-k} to denote the set $S \setminus \{s_k\}$ (the set of all elements but the largest).

We prove the assertion by reducing $\mathsf{DISJ}_{u,m}$ to $\mathsf{Max\text{-}Pos\text{-}CSP}_n^{g_k}$, for $m = \left\lfloor \frac{u}{4} \right\rfloor - 1$. Note that by Lemma 2.1, since $4m+1 \leq u$ it holds that $\mathsf{CC}(\mathsf{DISJ}_{u,m}) \geq \mathsf{CC}(\mathsf{DISJ}_{4m+1,m}) \geq \Omega(m) = \Omega\big(n^k\big)$. Therefore, such a reduction indeed gives the claimed $\mathsf{CC}(\mathsf{Max\text{-}Pos\text{-}CSP}_n^{g_k}) \geq \Omega\big(n^k\big)$.

4.1.1 The Reduction

Let $(\mathbf{y}, \mathbf{z})$ be an instance of $\mathsf{DISJ}_{u,m}$. Recall that $\|\mathbf{y}\| = \|\mathbf{z}\| = m$. We view $\mathbf{y}$ and $\mathbf{z}$ as elements in $\{0,1\}^U$, vectors indexed by elements of U (for $S \in U$, we write, *e.g.*, y_S, to mean coordinate S of $\mathbf{y}$). We construct an instance $(\Psi, C) = ((\Psi^{\mathbf{y}}, \Psi^{\mathbf{z}}), C)$ for $\mathsf{Max\text{-}Pos\text{-}CSP}_n^{g_k}$ over the variables $\mathbf{x} = (x_1, \ldots, x_n)$ as follows. Let $C = 4u - 4m + k$. For every $S \in U$, if $y_S = 0$, Alice adds to $\Psi^{\mathbf{y}}$ the following three clauses: $\mathbf{x}_S$, $\left(\mathbf{x}_{S_{-k}}, 0\right)$, and $\left(\mathbf{0}^{k-1}, x_{s_k}\right)$. Intuitively, these clauses allow us to embed an OR clause, as can be seen in the following equality: Let $\mathbf{w} \in \{0,1\}^k$ and let $b = \bigvee_{i \in [k-1]} w_i$. Then,

$$g_k(\mathbf{w}) + g_k(w_1, \ldots, w_{k-1}, 0) + g_k\left(\mathbf{0}^{k-1}, w_k\right) = \\ (b \oplus w_k) + b + w_k = 2(b \vee w_k) = 2 \cdot \left(\bigvee_{i \in [k]} w_i\right). \tag{3}$$

Likewise, Bob constructs an analogous set of clauses $\Psi^{\mathbf{z}}$, using $\mathbf{z}$ in place of $\mathbf{y}$.

Additionally, Alice adds the following clauses to $\Psi^{\mathbf{y}}$: For $i \in \{1, \ldots, n/2\}$, the clause $\left(\mathbf{1}^{k-1}, x_i\right)$. Bob adds the following clauses to $\Psi^{\mathbf{z}}$: For $i \in \{n/2+1, \ldots, n\}$, the clause $\left(\mathbf{1}^{k-1}, x_i\right)$ (we assume that n is even here). Observe that since $\|\mathbf{y}\| = \|\mathbf{z}\|$, we get that $\Psi^{\mathbf{y}}$ and $\Psi^{\mathbf{z}}$ have the same number of clauses.

4.1.2 Analysis

We next prove that the reduction works. Let $(\mathbf{y}, \mathbf{z})$ be an instance of $\mathsf{DISJ}_{u,m}$ and let $(\Psi, C) = ((\Psi^{\mathbf{y}}, \Psi^{\mathbf{z}}), C)$ be the instance of $\mathsf{Max\text{-}Pos\text{-}CSP}_n^{g_k}$ resulting from the reduction. We next show that $\mathsf{Max\text{-}Pos\text{-}CSP}_n^{g_k}(\Psi) < C$ if and only if $\mathsf{DISJ}_{u,m}(\mathbf{y}, \mathbf{z}) = 1$.

Let $\mathbf{x} \in \{0,1\}^n$ be an assignment. We denote $U_0(\mathbf{x}) = \left\{S \in U : \bigvee_{i \in S} x_i = 0\right\}$. Now, let us calculate $\Psi(\mathbf{x})$ using Eq. (3) (observe that $y_S = 0$ means $1 - y_S = 1$):

$$\begin{aligned}
\Psi(\mathbf{x}) &= \sum_{S \in U} (2 - y_S - z_S)\left(g_k(\mathbf{x}_S) + g_k\left(\mathbf{x}_{S_{-k}}, 0\right) + g_k\left(\mathbf{0}^{k-1}, x_{s_k}\right)\right) + \sum_{i \in [n]} g_k\left(\mathbf{1}^{k-1}, x_i\right) \\
&= 2 \sum_{S \in U} (2 - y_S - z_S)\left(\bigvee_{i \in S} x_i\right) + \sum_{i \in [n]} (1 - x_i) \\
&= 4u - 2\|\mathbf{y}\| - 2\|\mathbf{z}\| - 2 \sum_{S \in U_0(\mathbf{x})} (2 - y_S - z_S) + n - \|\mathbf{x}\| \\
&= 4u - 4m - 2 \sum_{S \in U_0(\mathbf{x})} (2 - y_S - z_S) + n - \|\mathbf{x}\|. \qquad (4)
\end{aligned}$$

$\mathbf{y}$ and $\mathbf{z}$ intersect. First, suppose that $\mathsf{DISJ}_{u,m}(\mathbf{y}, \mathbf{z}) = 0$ and let $S^* \in U$ be such that $y_{S^*} = z_{S^*} = 1$. Consider the assignment $\mathbf{x} \in \{0,1\}^n$ with $x_i = 0$ if and only if $i \in S^*$. We will show that $\Psi(\mathbf{x}) = C$. To this end, observe that $U_0(\mathbf{x}) = \{S^*\}$, that $2 - y_{S^*} - z_{S^*} = 0$, and that $\|\mathbf{x}\| = n - k$. By Eq. (4), $\Psi(\mathbf{x}) = 4u - 4m - 0 + k = C$.

y and z are disjoint. Now suppose that $\mathsf{DISJ}_{u,m}(\mathbf{y},\mathbf{z}) = 1$. Thus, for every $S \in U$, $y_S = 0$ or $z_S = 0$, implying $2 - y_S - z_S \geq 1$. We will show that $\mathsf{Max\text{-}Pos\text{-}CSP}_n^{g_k}(\Psi) < C$.

Let $\mathbf{x} \in \{0,1\}^n$ be an assignment. We consider two cases. The first is the case where $U_0(\mathbf{x}) \neq \emptyset$. Note that in this case, $\|\mathbf{x}\| \leq n - k$ and also $|U_0(\mathbf{x})| = \binom{n-\|\mathbf{x}\|}{k}$. Also note that since $k \geq 1$, for all $\ell \geq k$, it holds that $\binom{\ell}{k} \geq \ell - k$. Thus, by Eq. (4), we have

$$\begin{aligned}
\Psi(\mathbf{x}) &\leq 4u - 4m - 2|U_0(\mathbf{x})| + (n - \|\mathbf{x}\|) \\
&= 4u - 4m - 2\binom{n - \|\mathbf{x}\|}{k} + (n - \|\mathbf{x}\|) \\
&< 4u - 4m - \binom{n - \|\mathbf{x}\|}{k} + (n - \|\mathbf{x}\|) \\
&\leq 4u - 4m + k \\
&= C.
\end{aligned}$$

Now consider the case where $U_0(\mathbf{x}) = \emptyset$. Note that this implies that $\|\mathbf{x}\| \geq n - k + 1$. By Eq. (4), we get $\Psi(\mathbf{x}) \leq 4u - 4m + (k-1) < C$.

5 Communication Lower Bound for Max-CUT

In this section, we prove Theorem 1.2. By a standard argument, Theorem 1.2 is implied by the following communication lower bound:

Theorem 5.1. $\mathsf{CC}(\mathsf{Max\text{-}CUT}_n) \geq \Omega(n^2)$.

Theorem 5.1 is proved in two steps. We first show a lower bound on the related problem 3IND-SET, and then show how to convert this lower bound to a communication lower bound for Max-CUT.

5.1 Lower Bound for 3IND-SET

In this section, we prove a lower bound on the communication complexity necessary to solve the independent set problem $\mathsf{3IND\text{-}SET}_n$. In this problem, both Alice and Bob are given (disjoint) sets of edges over the same set of n vertices and their goal is to output whether or not the graph formed by the union of their sets has an independent set of size 3.

Theorem 5.2 (see [PS82][12]). $\mathsf{CC}(\mathsf{3IND\text{-}SET}_n) \geq \Omega(n^2)$.

Proof. We prove this result by a reduction. Let $m = \frac{n^2-1}{4}$. Recall by Lemma 2.1 that $\mathsf{CC}(\mathsf{DISJ}_{n^2,m}) \geq \Omega(n^2)$. Given an instance $\mathbf{x}, \mathbf{y}$ of $\mathsf{DISJ}_{n^2,m}$, where Alice's and Bob's inputs are viewed as vectors $\mathbf{x}, \mathbf{y} \in \{0,1\}^{[n]\times[n]}$ respectively, Alice and Bob create an instance of $\mathsf{3IND\text{-}SET}_{3n}$ as follows: They view the $3n$ vertices as 3 disjoint sets V_0, V_A, and V_B of n vertices each and construct the following edges:

[12] We thank the anonymous reviewer for telling us that this theorem follows from [PS82].

1. The vertices in the set V_0 are all connected to each other to form a clique. The same for the sets V_A and V_B. Finally, for all $j \neq j' \in [n]$, vertex j in V_A is connected to vertex j' in V_B. Note that these edges are known to both Alice and Bob as they are independent of their input.

2. For all $(j, j') \in [n] \times [n]$, Alice (respectively, Bob) adds an edge between vertex j in V_0 and vertex j' in V_A (respectively, V_B) if and only if $x_{(j,j')} = 0$ (resp. $y_{(j,j')} = 0$). These edges are functions of the input and are only known to one of the parties. Moreover, Alice's and Bob's edges are disjoint.

We claim that the above graph has an independent set of size 3 if and only if Alice's and Bob's inputs for disjointness are intersecting. Indeed, as Item 1 implies that the sets V_0, V_A, V_B all form cliques, any independent set of size 3 must have exactly one vertex from each of these sets. Moreover, due to edges between V_A and V_B defined above, we get that an independent set of size 3 exists if and only if there exists $(j, j') \in [n] \times [n]$ such that vertex j in V_0, vertex j' in V_A, and vertex j' in V_B form an independent set. Due to the edges in Item 2, this happens if and only if there exists $(j, j') \in [n] \times [n]$ such that $x_{(j,j')} = y_{(j,j')} = 1$, as desired. □

5.2 Lower Bound for Max-CUT

We now reduce 3IND-SET to Max-CUT and prove Theorem 5.1.

Proof of Theorem 5.1. We prove this result by a reduction from 3IND-SET$_n$. Given an instance $G = (V, E = E_A \cup E_B)$ of 3IND-SET$_n$, where Alice has edges E_A and Bob has edges E_B, Alice and Bob create an instance G' of Max-CUT$_{21n}$ as follows: They view the $21n$ vertices as 3 disjoint sets V_G, V_0, and V_1 of n, $10n$, and $10n$ vertices respectively and construct the following edges:

1. The set V_0 and V_1 are made to form a complete bipartite graph by connecting every vertex in V_0 with every vertex in V_1. Also, for all $j \in [n]$, we connect vertex j in V_G to vertex j in V_1. Note that these edges are known to both Alice and Bob as they are independent of their input.

2. For each edge $(j, j') \in E_A$, Alice creates the corresponding edge in V_G and also connects vertex j in V_G to vertex j' in V_0 and connects vertex j' in V_G to vertex j in V_0. We call these three edges the "frame" of (j, j') and note that these edges are functions of Alice's input and are only known to her. We construct Bob's edges analogously. Observe that Alice's and Bob's edges are disjoint (as they were disjoint in the 3IND-SET instance).

We now claim that the constructed instance has a maximum cut size of at least $C = (10n)^2 + 2|E| + 3$ if and only if G has a 3-independent set[13]. To see the "if" direction, let

[13]Note that both the parties can compute C by computing $|E|$ which requires only $O(\log n)$ bits of communication. This communication can be ignored as we are proving an $\Omega(n^2)$ lower bound.

$\{i, j, k\}$ be an independent set of size 3 in G and consider the cut formed by putting V_0 and vertices i, j, k of V_G on one side and every other vertex on the other. This cut has $(10n)^2 + 3$ edges of Item 1 above ($(10n)^2$ between V_0 and V_1 and 3 edges between V_G and V_1) and also has $2|E|$ of Item 2 above (as $\{i, j, k\}$ is an independent set, 2 out of 3 edges in all the frames are in the cut). Thus, there exists a cut of size at least C, as desired.

It remains to show the "only if" direction. Suppose that G has no independent set of size 3 and suppose for the sake of contradiction that the largest (breaking ties arbitrarily) cut $(S, \overline{S})$ in the instance G' has size at least C. As there are only $3|E| + n \leq \frac{3}{2} \cdot n^2$ other edges in the graph, the cut $(S, \overline{S})$ must have at least $C - \frac{3}{2} \cdot n^2 > 90n^2$ of the edges between V_0 and V_1 in Item 1. Observe that this is possible only if at least $9n$ of the vertices in V_0 are on one side of the cut and at least $9n$ of the vertices in V_1 are on the other side of the cut. Without loss of generality, we assume that S has at least $9n$ of the vertices in V_0 (and at most n of the vertices in V_1).

We claim that, in fact, S has all the vertices in V_0 and none of the vertices in V_1. Indeed, suppose that there is a vertex in $V_0 \setminus S$ and consider the cut obtained by moving this vertex to S. As $9n$ of the vertices in V_1 are in $\overline{S}$, we have by Items 1 and 2 that moving this vertex to S cuts at least $9n$ new edges and "uncuts" at most $6n$ edges, thereby increasing the size of the cut, and contradicting the fact that $(S, \overline{S})$ was the largest cut. A similar argument applies if there is a vertex in $V_1 \cap S$ and we are done.

Defining $T = S \setminus V_0$ and using the above claim, we get that $T \subseteq V_G$ and $(S, \overline{S}) = ((T \cup V_0), ((V_G \setminus T) \cup V_1))$. Letting E_T be the set of edges with both endpoints in T and using a calculation similar to that in the "if" direction above, we get that the size of the cut $(S, \overline{S})$ is at most $(10n)^2 + |T| + 2 \cdot (|E| - |E_T|)$. Now, we claim (proved later) that $|E_T| \geq \frac{|T|^2}{4} - \frac{|T|}{2}$, implying that the size of the cut $(S, \overline{S})$ is at most $(10n)^2 + 2 \cdot |T| + 2 \cdot |E| - \frac{|T|^2}{2}$. Setting $z = |T|$ in the identity $(z-2)^2 = z^2 - 4z + 4 \geq 0$, this is at most $(10n)^2 + 2 + 2 \cdot |E| < C$, a contradiction.

It remains to prove the claim. As $T \subseteq V_G$, we can identify T with a subset of the vertices in G. With this identification, E_T is just the subgraph of G induced by those vertices, and does not have an independent set of size 3. It follows that the complement of this subgraph does not have a triangle and therefore, has at most $\frac{|T|^2}{4}$ edges by Turán's Theorem [Man07, Tur41]. As the maximum number of edges is $\binom{|T|}{2}$, we get that:

$$E_T \geq \frac{|T| \cdot (|T| - 1)}{2} - \frac{|T|^2}{4} = \frac{|T|^2}{4} - \frac{|T|}{2}.$$

□

References

[ACHKP21] Amir Abboud, Keren Censor-Hillel, Seri Khoury, and Ami Paz. Smaller cuts, higher lower bounds. *ACM Transactions on Algorithms (TALG)*, 17(4):1–40,

2021. 3

[AKSY20] Sepehr Assadi, Gillat Kol, Raghuvansh R Saxena, and Huacheng Yu. Multi-pass graph streaming lower bounds for cycle counting, max-cut, matching size, and other problems. In *Symposium on Foundations of Computer Science (FOCS)*, 2020. 3

[AR20] Sepehr Assadi and Ran Raz. Near-quadratic lower bounds for two-pass graph streaming algorithms. In *Symposium on Foundations of Computer Science (FOCS)*, pages 342–353, 2020. 1

[Aus07] Per Austrin. Balanced max 2-sat might not be the hardest. In David S. Johnson and Uriel Feige, editors, *Symposium on Theory of Computing (STOC)*, pages 189–197, 2007. 1

[Aus10] Per Austrin. Towards sharp inapproximability for any 2-csp. *SIAM J. Comput.*, 39(6):2430–2463, 2010. 1

[AV21] Sepehr Assadi and N Vishvajeet. Graph streaming lower bounds for parameter estimation and property testing via a streaming xor lemma. In *Symposium on Theory of Computing (STOC)*, pages 612–625. Association for Computing Machinery, 2021. 3, 18

[BCHD+19] Nir Bacrach, Keren Censor-Hillel, Michal Dory, Yuval Efron, Dean Leitersdorf, and Ami Paz. Hardness of distributed optimization. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing*, pages 238–247, 2019. 3

[BDV18] Aditya Bhaskara, Samira Daruki, and Suresh Venkatasubramanian. Sublinear algorithms for maxcut and correlation clustering. In *International Colloquium on Automata, Languages, and Programming (ICALP)*, 2018. 3

[BHP+21] Joanna Boyland, Michael Hwang, Tarun Prasad, Noah Singer, and Santhoshini Velusamy. Closed-form expressions for the sketching approximability of (some) symmetric boolean csps. *arXiv preprint arXiv:2112.06319*, 2021. 1

[Bul17] Andrei A. Bulatov. A dichotomy theorem for nonuniform csps. In *Symposium on Foundations of Computer Science (FOCS)*, pages 319–330, 2017. 1

[CGS+22] Chi-Ning Chou, Alexander Golovnev, Madhu Sudan, Ameya Velingker, and Santhoshini Velusamy. Linear space streaming lower bounds for approximating csps. In *Symposium on Theory of Computing (STOC)*, pages 275–288. ACM, 2022. 1, 3

[CGSV21] Chi-Ning Chou, Alexander Golovnev, Madhu Sudan, and Santhoshini Velusamy. Approximability of all finite csps with linear sketches. In *Symposium on Foundations of Computer Science (FOCS)*, pages 1197–1208. IEEE, 2021. 1, 3

[CGV20] Chi-Ning Chou, Alexander Golovnev, and Santhoshini Velusamy. Optimal streaming approximations for all boolean max-2csps and max-ksat. In *Symposium on Foundations of Computer Science (FOCS)*, pages 330–341. IEEE, 2020. 1

[CKP+21] Lijie Chen, Gillat Kol, Dmitry Paramonov, Raghuvansh R Saxena, Zhao Song, and Huacheng Yu. Almost optimal super-constant-pass streaming lower bounds for reachability. In *Symposium on Theory of Computing (STOC)*, pages 570–583, 2021. 1

[CKS01] Nadia Creignou, Sanjeev Khanna, and Madhu Sudan. *Complexity classifications of Boolean constraint satisfaction problems.* SIAM, 2001. 3

[FV98] Tomás Feder and Moshe Y. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through datalog and group theory. *SIAM J. Comput.*, 28(1):57–104, 1998. 1

[GO16] Venkatesan Guruswami and Krzysztof Onak. Superlinear lower bounds for multipass graph processing. *Algorithmica*, 76(3):654–683, 2016. 1

[GT19] Venkatesan Guruswami and Runzhou Tao. Streaming hardness of unique games. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, 2019. 1

[Ind06] Piotr Indyk. Stable distributions, pseudorandom generators, embeddings, and data stream computation. *Journal of the ACM (JACM)*, 53(3):307–323, 2006. 18

[Kho10] Subhash Khot. On the unique games conjecture (invited survey). In *Conference on Computational (CCC)*, pages 99–121. IEEE Computer Society, 2010. 1

[KK15] Dmitry Kogan and Robert Krauthgamer. Sketching cuts in graphs and hypergraphs. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*, pages 367–376, 2015. 3

[KK19] Michael Kapralov and Dmitry Krachun. An optimal space lower bound for approximating max-cut. In *Symposium on Theory of Computing (STOC)*, pages 277–288, 2019. 3

[KKS15] Michael Kapralov, Sanjeev Khanna, and Madhu Sudan. Streaming lower bounds for approximating MAX-CUT. In *Symposium on Discrete Algorithms (SODA)*, pages 1263–1282, 2015. 3

[KKSV17] Michael Kapralov, Sanjeev Khanna, Madhu Sudan, and Ameya Velingker. $(1+\omega(1))$-approximation to max-cut requires linear space. In *Symposium on Discrete Algorithms (SODA)*, pages 1703–1722, 2017. 3

[KNW10] Daniel M Kane, Jelani Nelson, and David P Woodruff. On the exact space complexity of sketching and streaming small norms. In *Symposium on Discrete Algorithms (SODA)*, pages 1161–1178. SIAM, 2010. 18

[Man07] W. Mantel. Vraagstuk XXVIII. *Wiskundige Opgaven*, 10:60–61, 1907. 14

[Mos10] Elchanan Mossel. Gaussian bounds for noise correlation of functions. In *Geometric and Functional Analysis*, volume 19(6), pages 1713–1756, 2010. 1

[PS82] Christos H Papadimitriou and Michael Sipser. Communication complexity. In *Proceedings of the fourteenth annual ACM symposium on Theory of computing*, pages 196–200, 1982. 12

[Rag08] Prasad Raghavendra. Optimal algorithms and inapproximability results for every csp? In *Symposium on Theory of Computing (STOC)*, pages 245–254, 2008. 1

[Raz90] Alexander A Razborov. On the distributional complexity of disjointness. In *International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 249–253. Springer, 1990. 6

[Sch78] Thomas J. Schaefer. The complexity of satisfiability problems. In *Symposium on Theory of Computing (STOC)*, pages 216–226, 1978. 1

[SSV21] Noah Singer, Madhu Sudan, and Santhoshini Velusamy. Streaming approximation resistance of every ordering CSP. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, volume 207 of *LIPIcs*, pages 17:1–17:19, 2021. 1

[Sud22] Madhu Sudan. Streaming and sketching complexity of csps: A survey (invited talk). In *International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 229 of *LIPIcs*, pages 5:1–5:20, 2022. 3

[Tur41] Paul Turán. On an external problem in graph theory. *Mat. Fiz. Lapok*, 48:436–452, 1941. 14

[Zel11] Mariano Zelke. Intractability of min- and max-cut in streaming graphs. *Inf. Process. Lett.*, 111(3):145–150, 2011. 3

[Zhu20] Dmitriy Zhuk. A proof of the CSP dichotomy conjecture. *Journal of the ACM (JACM)*, 67(5):30:1–30:78, 2020. 1

A Streaming Lower Bound for Approximate Max-CSP

In this section, we will show a multi-pass lower bound for arbitrarily good approximations of Max-CSP.

Theorem A.1. *Let $k, n \in \mathbb{N}$ and $f : \{0,1\}^k \to \{0,1\}$ with $\mathsf{deg}(f) > 1$. Then, for all $\epsilon > 0, p \in \mathbb{N}$, any p-pass streaming algorithm for $\mathsf{Max\text{-}CSP}^f_{n,\epsilon}$ has space complexity $\Omega\left(n^{1-O(\epsilon p^2)}\right)$.*

Theorem A.1 is tight in two respects: First, recall from Section 1 that the space lower bound cannot be improved beyond $O(n)$, as there is an $O(n)$-space upper bound for any function f. Additionally, for the case where $\mathsf{deg}(f) \leq 1$, there is in fact an $O_\epsilon(\log n)$-space, one-pass streaming algorithm for $\mathsf{Max\text{-}CSP}^f_{n,\epsilon}$. The reason is that the only way $\mathsf{deg}(f) \leq 1$ is if f is constant (in which case an algorithm is trivial), or there exists $i \in [n]$ such that $f(\mathbf{x}) = x_i$ or $f(\mathbf{x}) = \overline{x_i}$, in which case $\mathsf{Max\text{-}CSP}^f_{n,\epsilon}$ is the same as approximating an ℓ_1-norm, algorithms for which can be found in, *e.g.*, [Ind06, KNW10].

Proof of Theorem A.1. Proof by contradiction. Suppose that there exists a p-pass streaming algorithm $\mathcal{A}$ for $\mathsf{Max\text{-}CSP}^f_{n,\epsilon}$ with a better space complexity. As $\mathsf{deg}(f) > 1$, we have by Corollary 3.3 that there exists $\epsilon' > 0$ and a streaming algorithm $\mathcal{A}'$ for $\mathsf{Max\text{-}AND}^2_{n,\epsilon'}$ with the same space complexity, up to constant factors.

We now claim that there exists $\epsilon'' > 0$ and a streaming algorithm $\mathcal{A}''$ for $\mathsf{Max\text{-}CSP}^{\mathsf{XOR}_2}_{n,\epsilon''}$ with the same space complexity. Indeed, we can expand any XOR constraint $a \oplus b$ as the sequence of two constraints $a \wedge \bar{b}$ and $\bar{a} \wedge b$ and observe that at most one of these two constraints can be satisfied by any assignment and is satisfied if and only if the assignment satisfies the constraint $a \oplus b$. The algorithm $\mathcal{A}''$ is obtained by running $\mathcal{A}'$ on the expanded constraints. Finally, as the problem $\mathsf{Max\text{-}CSP}^{\mathsf{XOR}_2}_{n,\epsilon''}$ subsumes $\mathsf{Max\text{-}CUT}_{n,\epsilon''}$, this contradicts Result 2 in [AV21].

□

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il