

Nine lower bound conjectures on streaming approximation algorithms for CSPs

Noah G. Singer*

September 29, 2025

Open Problems Column Edited by William Gasarch

Request for Columns! I invite any reader who has knowledge of some area to contact me and arrange to write a column about open problems in that area. That area can be (1) broad or narrow or anywhere inbetween, and (2) really important or really unimportant or anywhere inbetween.

1 Introduction

Inspired by an open question at the 2011 Bertinoro workshop [22], the last decade has seen an explosion of interest in using *streaming algorithms* for *approximating constraint satisfaction problems (CSPs)*. Some results we know in this area include:

- Single-pass lower bounds for MAX-CUT [32, 28, 29, 27],
- Multi-pass lower bounds for MAX-CUT [2, 1, 6, 14] and other CSPs [15],
- Algorithms and lower bounds for approximating MAX-DICUT [19, 7, 37, 36, 35],
- Quantum algorithms and lower bounds for MAX-CUT and MAX-DICUT [24, 25, 26],
- Results on other specific CSPs, including unique games ([18]), monarchy-like predicates ([9]), and MAX- k AND ([41]),
- Dichotomy theorems and results for general CSPs [8, 10, 11],

and various other results, including lower bounds for *ordering CSPs* (including MAX-ACYCLIC-SUBGRAPH and MAX-BETWEENNESS) [40], for solving CSPs *exactly* [47, 44, 33], and for solving CSPs approximately on *dense* instances [3]. See the surveys [39, 43, 46] for some (perhaps already out of date!) exposition.

In this column, I highlight nine “frontier” conjectures that have emerged in recent works in this area (and give some brief overviews of the notions needed to understand the questions). I will do my best to cite conjectures if they already appear in published work; some appear may here for the first time.

Acknowledgements. These conjectures have emerged out of discussions and papers with several wonderful collaborators of mine including Raghuvansh Saxena, Madhu Sudan, and Santhoshini Velusamy. I would also like to thank Matthew Ding, William He, and Michael Kapralov for useful discussions; Bill Gasarch for his helpful proofreading and feedback on this column; and my advisor Ryan O’Donnell for his support. During the time of writing, I was supported by an NSF Graduate Research Fellowship (Award DGE2140739).

*Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA. Email: ngsinger@cs.cmu.edu.

2 Constraint satisfaction problems

Constraint satisfaction problems (CSPs) capture a broad class of computational problems. In this column, we will only consider *maximization* CSPs; these include numerous well-studied problems such as MAX-CUT, MAX-DICUT, MAX- k SAT, MAX- k XOR, MAX- q COLORING, and MAX- q UNIQUEGAMES.¹ These problems, and their hardness of approximation, have been studied extensively throughout complexity theory; see e.g. [23, 12, 17, 45, 20, 5] (a small, chronological sampling of many, many papers). Maximization CSPs are also intimately connected with the unique games conjecture [30, 31, 34] and with probabilistically checkable proofs [13].

In this column, we restrict further to the case of *Boolean* CSPs, which keeps things interesting while simplifying notation. Here is the general setup we consider. Let $k \in \mathbb{N}$ be a (typically small) number, the *arity*, and let $\Pi \subseteq (\{0, 1\}^k)^{\{0, 1\}}$ denote a set of *predicate* functions $\{0, 1\}^k \rightarrow \{0, 1\}$. For $n \in \mathbb{N}$, a *constraint* is a tuple $C = (j_1, \dots, j_k; \pi)$ for distinct $j_1, \dots, j_k \in [n]$ and $\pi \in \Pi$. An *assignment* is a vector $x = (x_1, \dots, x_n) \in \{0, 1\}^n$, and x *satisfies* the constraint $C = (j_1, \dots, j_k; \pi)$ iff $\pi(x_{j_1}, \dots, x_{j_k}) = 1$. An *instance* Φ consists of a list of constraints, and the *value* of an assignment $x \in \{0, 1\}^n$ on Φ is

$$\text{val}_\Phi(x) := \Pr_{C \sim \Phi} [x \text{ satisfies } C],$$

(here the distribution on C is uniform over all constraints, or sometimes Φ might also specify a weight distribution). The goal of the problem MAX-CSP(Π) is to approximate the quantity

$$\text{max-val}(\Phi) := \max_{x \in \{0, 1\}^n} \text{val}_\Phi(x),$$

the maximum value of any assignment. Specifically, we say $v \in [0, 1]$ is an α -*approximation* for MAX-CSP(Π) if $\alpha \cdot \text{max-val}(\Phi) \leq v \leq \text{max-val}(\Phi)$. We let

$$\alpha_{\text{triv}}(\Pi) := \lim_{n \rightarrow \infty} \left[\inf_{\Phi, \text{ MAX-CSP}(\Pi) \text{ inst. on } n \text{ vars.}} \text{max-val}(\Phi) \right]$$

denote the so-called “trivial approximation ratio” for Π ; this is, informally, the best possible lower bound on $\text{max-val}(\Phi)$ which does not actually depend on Φ . Note that for every $\epsilon > 0$ and large enough n , the value $\alpha_{\text{triv}}(\Pi) - \epsilon$ is always a $(\alpha_{\text{triv}}(\Pi) - \epsilon)$ -approximation for MAX-CSP(Π). The complexity-theoretic question we are interested in is: Are $(\alpha_{\text{triv}}(\Pi) + \epsilon)$ -approximations possible, and if so, how large can ϵ be?

Examples of CSPs. The definition in the previous paragraph captures a wide array of CSPs, but it turns out that even very simple special cases are quite interesting from a complexity-theoretic perspective. The simplest interesting CSP is MAX-CUT, wherein $k = 2$ and $\Pi = \{\text{CUT}\}$ where $\text{CUT}(x_1, x_2) := x_1 \oplus x_2$ (where $\oplus$ is the binary XOR operation). (Equivalently, $\text{CUT}(x_1, x_2) = 1$ iff $x_1 \neq x_2$.) The second simplest CSP is MAX-DICUT, where again $k = 2$ but $\Pi = \{\text{DICUT}\}$ where $\text{DICUT}(x_1, x_2) := 1[x_1 = 1 \wedge x_2 = 0]$ (equiv., $\text{DICUT}(x_1, x_2) = x_1 \wedge \overline{x_2}$).

Here is another interesting example: For $k \in \mathbb{N}$ and $b \in \{0, 1\}^k$, let $\text{NOT}_b : \{0, 1\}^k \rightarrow \{0, 1\}^k$ be the function $\text{NOT}_b(x_1, \dots, x_k) := (x_1 \oplus b_1, \dots, x_k \oplus b_k)$. (It is useful to think of NOT_b as placing negations on some variables. For instance, $\text{NOT}_{011}(x_1, x_2, x_3) = (x_1, \overline{x_2}, \overline{x_3})$.) Let $k\text{AND} : \{0, 1\}^k \rightarrow \{0, 1\}$ be the function $k\text{AND}(x_1, \dots, x_k) = \bigwedge_{i=1}^k x_i$. In the MAX- k AND problem, $\Pi = \{k\text{AND} \circ \text{NOT}_b : b \in \{0, 1\}^k\}$. (For instance, $2\text{AND} \circ \text{NOT}_{01} = \text{DICUT}$.)

Note that MAX-CUT and MAX-DICUT both involve only one predicate. Further, the predicate CUT is symmetric to reordering its inputs. Thus, it simplifies notation to imagine MAX-CUT constraints as unordered pairs $\{j_1, j_2\}$ and MAX-DICUT constraints as ordered pairs (j_1, j_2) . Correspondingly, we can view the input to a MAX-CUT problem as an undirected graph $\mathcal{G}$ on vertex-set $[n]$ and the input to a MAX-DICUT problem as a directed graph $\mathcal{G}$ on $[n]$, and refer to the constraints in these problems as edges.

¹By “maximization”, we mean that the goal is to determine (or approximate) the maximum satisfiable fraction of constraints. A related problem is deciding whether there exists an assignment satisfying all constraints; a dichotomy theorem for such problems was shown in the seminal work of Schaefer [38], who showed that every (Boolean) such problem is either in P or is NP-complete. Creignou [12] established a similar theorem for Boolean maximization CSPs.

It is not hard to check that $\alpha_{\text{triv}}(\text{MAX-CUT}) = \frac{1}{2}$, $\alpha_{\text{triv}}(\text{MAX-DICUT}) = \frac{1}{4}$, and $\alpha_{\text{triv}}(\text{MAX-}k\text{AND}) = \frac{1}{2k}$. (For instance, for MAX-CUT, a random graph $\mathcal{G}$ with $\Omega_\epsilon(n)$ edges typically has $\text{max-val}(\mathcal{G}) \leq \frac{1}{2} + \frac{\epsilon}{2}$, while for every graph $\mathcal{G}$, a uniformly random assignment $\mathbf{x} \in \{0,1\}^n$ has $\mathbb{E} \text{val}_{\mathcal{G}}(\mathbf{x}) = \frac{1}{2}$.) Goemans and Williamson [17] famously showed that very nontrivial ($\frac{2}{\pi} \max_{0 \leq \theta \leq \pi} \frac{\theta}{1 - \cos \theta} \approx 0.878$ -) approximations to MAX-CUT are possible in polynomial time, and subsequent decades have seen extensive work on the polynomial-time approximability of these problems; beating this ratio is known to be NP-hard assuming the unique games conjecture [31].

3 Streaming algorithms

In this column, we are interested in the MAX-CSP(Π) problem in a specific algorithmic model, namely, the *streaming* model. In this model, the algorithm has the following kind of access to an input instance Φ : First, it receives the number of variables n in Φ , and then it receives the constraints $C_1, \dots, C_m$ in Φ one by one (in a possibly adversarial order). Between receiving constraints C_i and C_{i+1} , the algorithm may only store s bits of internal memory state, where s is a (typically small) function of n .² At the end of the stream, the algorithm is asked to output an α -approximation to $\text{max-val}(\Phi)$; the complexity-theoretic question is how much space is required to achieve particular values of α .

Formalizing this is not difficult: For fixed n , a deterministic algorithm for MAX-CSP(Π) is a pair $(\text{Alg} : \mathcal{C} \times \{0,1\}^s \rightarrow \{0,1\}^s, \text{Output} : \{0,1\}^s \rightarrow [0,1])$ where $\mathcal{C}$ is the set of possible constraints for Π . The algorithm starts at some initial state S_0 ; as constraints arrive, the state $S_{j+1} \leftarrow \text{Alg}(C_j, S_j)$ updates iteratively, and the final output is $\text{Output}(S_j)$. A randomized algorithm for MAX-CSP(Π) is a distribution over deterministic algorithms. In this column, we are concerned with algorithms achieving, say, $\frac{2}{3}$ probability of outputting correct approximations.

Standard sparsification arguments show that for any MAX-CSP(Π) instance Φ , if Φ is a “subsamped” random instance with $m = \Theta(n/\epsilon^2)$ constraints, each of which is sampled i.i.d. uniformly from Φ , then w.h.p. $|\text{max-val}(\Phi) - \text{max-val}(\Phi)| \leq \epsilon$. This essentially gives the following algorithmic result:

Theorem 3.1 (Folklore, see e.g. [10]). *For every constraint family Π and $\epsilon > 0$, there is an $(1 - \epsilon)$ -approximation streaming algorithm for MAX-CSP(Π) in $O(n \log n / \epsilon^2)$ bits of space.*

Remark 3.2. Our definition of the streaming model makes no assumptions about the algorithm’s running time, meaning that an algorithm can calculate $\text{max-val}(\Phi)$ *exactly* (even though this problem is NP-hard). $\diamond$

On the other end of the spectrum, simply outputting the trivial approximation $\alpha_{\text{triv}}(\text{MAX-CSP}(\Pi))$ uses zero space and achieves an $(\alpha_{\text{triv}}(\text{MAX-CSP}(\Pi)) - \epsilon)$ -approximation for $\epsilon > 0$. The “nontrivial” regime, therefore, is using space $\omega(1)$ and $o(n \log n)$ to get approximation ratios $\alpha_{\text{triv}}(\text{MAX-CSP}(\Pi)) < \alpha \leq 1$. For some CSPs, like MAX-CUT, it appears that this is essentially impossible [28, 27, 14], and the interesting questions are about proving lower bounds with optimal parameters (see §§4 and 5). For other CSPs, like MAX-DICUT, nontrivial approximations are possible [19, 7, 11, 36], and there are many open questions on tradeoffs between streaming parameters and the approximation ratio (see §7 below).

Variant models. There are a few interesting variations on the streaming model we described above. At times, we make the model more generous to algorithms:

- assuming the provided list of constraints in Π is uniformly randomly ordered (as opposed to adversarially ordered),
- assuming the instance Π is “bounded-degree”, meaning that every individual variable $i \in [n]$ appears in only $O(1)$ constraints,
- allowing the algorithm to make multiple passes over the list of constraints $C_1, \dots, C_m$.

Conversely, when proving lower bounds, we might need to make more stringent assumptions on possible algorithms. Specifically, a *sketching* algorithm is a special type of streaming algorithm describable

²In this column, space is always measured in bits.

by functions $\text{Compress} : \mathcal{C} \rightarrow \{0,1\}^s$ and $\text{Compose} : \{0,1\}^s \times \{0,1\}^s \rightarrow \{0,1\}^s$ such that $\text{Alg}(S, C) = \text{Compose}(S, \text{Compress}(C))$ and:

$$\begin{aligned} \text{Compose}(\text{Compress}(C_1), \text{Compose}(\text{Compress}(C_2), \text{Compress}(C_3))) \\ = \text{Compose}(\text{Compose}(\text{Compress}(C_1), \text{Compress}(C_2)), \text{Compress}(C_3)). \end{aligned}$$

Informally, this rules out streaming algorithms that treat constraints differently depending on where they appear in the stream.³

Why streaming CSPs? There are a few reasons for why it is so interesting to study the approximability of CSPs via streaming algorithms. By ignoring time complexity, we can prove (unconditional!) lower bounds against streaming algorithms; these can be viewed as information-theoretic limits on the extent to which a MAX-CSP(II) instance Φ can be *compressed* while maintaining enough information to recover $\text{max-val}(\Phi)$. Indeed, all existing streaming lower bounds we cite in this column are unconditional and proven via techniques from communication complexity. At the same time, streaming algorithms can achieve nontrivial approximations for many problems, including MAX-DICUT ([19]). Progress on streaming algorithms for CSPs has employed ideas from sketching, sampling, local, and distributed algorithms; in turn, this progress has led to simpler *polynomial-time* approximation algorithms for some problems [4]. See Remark 6.1 below for some (very rough) intuition on why some CSPs admit algorithms in the streaming setting and others do not.

4 Single-pass, linear(ish)-space streaming lower bounds

Recall that MAX-CUT is the “simplest interesting” example of a CSP, and that the trivial approximation threshold for MAX-CUT is $\alpha_{\text{triv}}(\text{CUT}) = \frac{1}{2}$. It turns out that in the (sublinear-space) streaming setting, doing any better than a trivial $\frac{1}{2}$ -approximation for MAX-CUT is very hard. After a significant line of work [32, 28, 29, 27], the strongest single-pass lower bounds for MAX-CUT which we currently know are the following:

Theorem 4.1 (Kapralov and Krachun [27]). *For every $\epsilon > 0$, every single-pass adversarial-order streaming algorithm which $(\frac{1}{2} + \epsilon)$ -approximates MAX-CUT uses $\Omega(n)$ space.*

Theorem 4.2 (Kapralov, Khanna, and Sudan [28]). *For every $\epsilon > 0$, every single-pass random-order streaming algorithm which $(\frac{1}{2} + \epsilon)$ -approximates MAX-CUT uses $\Omega(\sqrt{n})$ space.*

Note the comparative weaknesses of the two bounds: The first holds only for adversarial-order streams (but in $o(n)$ space), and the second holds only in $o(\sqrt{n})$ space (but in randomly-ordered streams). It is natural to ask whether the limitations in Theorems 4.1 and 4.2 are artificial, or whether we can generalize both bounds simultaneously into a single lower bound:

Conjecture 1

For every $\epsilon > 0$, every single-pass random-order streaming algorithm which $(\frac{1}{2} + \epsilon)$ -approximates MAX-CUT uses $\Omega(n)$ space.

Remark 4.3. There are interesting technical reasons for why assuming adversarial input ordering and/or $o(\sqrt{n})$ -space makes it easier to prove streaming lower bounds. We will not delve deeply into lower bound techniques in this column, but we remark that the reasons are “real” for other CSPs: we know that from [36, 37] that for the related MAX-DICUT problem, allowing either random input ordering or $\tilde{O}(\sqrt{n})$ space strictly increases the achievable approximation ratio (vs. what $o(\sqrt{n})$ -space, adversarial-ordering algorithms can achieve). $\diamond$

³The reasons we consider sketching algorithms are twofold. Firstly, many natural algorithms for streaming CSPs are sketching algorithms [19, 7, 11, 36]. Secondly, sketching algorithms can be simulated in the *simultaneous communication model*. In turn, this model can be simulated by the *sequential communication model* (which can also simulate general streaming algorithms). It is often easier to prove lower bounds in the simultaneous model.

All known lower bounds for approximating CSPs via streaming algorithms, including [Theorems 4.1](#) and [4.2](#), use the following framework: Define two distributions $\mathcal{D}_{\text{Yes}}$ and $\mathcal{D}_{\text{No}}$ over streams of constraints, show that w.h.p. there is a large gap between the $\text{MAX-CSP}(\Pi)$ values of the corresponding instances, and then show that these distributions are indistinguishable in the streaming model of interest via a reduction from a hard one-way communication problem. Naturally, technical details of the “source” communication problem have significant impacts on the exact type of hardness we get for the “target” streaming problem (CSP approximation).

In the proof of [Theorem 4.1](#), $\mathcal{D}_{\text{Yes}}$ and $\mathcal{D}_{\text{No}}$ have order-sensitive definitions. More precisely, each stream in the support of $\mathcal{D}_{\text{Yes}}$ and $\mathcal{D}_{\text{No}}$ can be divided into $O(1)$ successive chunks such that within each chunk, the corresponding edges form a matching. The input distributions used in [Theorem 4.2](#) do not have this structure, which turns out to make proving lower bounds hairier. Morally, this is why the authors of [\[28\]](#) had to “settle” for a $o(\sqrt{n})$ -space lower bound. This gap between $\sqrt{n}$ space and n space is a common theme for several of the conjectures in this column.

Remark 4.4. [Conjecture 1](#) would imply lower bounds for $(\frac{1}{2} + \epsilon)$ -approximating MAX-DICUT with single-pass $o(n)$ -space random-order streaming algorithms (via the trivial reduction that randomly directs each edge); this would demonstrate the tightness of the random-ordering streaming algorithm for MAX-DICUT in [\[35\]](#). $\diamond$

Another conjecture about lower bounds for MAX-CUT with single-pass algorithms is the following:

Conjecture 2

For every $\epsilon > 0$, every single-pass adversarial-ordering streaming algorithm which $(\frac{1}{2} + \epsilon)$ -approximates MAX-CUT uses $\Omega(n \log n)$ space.

I.e., we hope to improve over [Theorem 4.1](#) by an additional logarithmic factor in the space usage. This would match the space usage of the generic sparsifier-based $(1 - \epsilon)$ -approximation for all CSPs ([Theorem 3.1](#)).

5 Multi-pass streaming lower bounds

For a long time, despite some works [\[2, 1\]](#) making partial progress, we seemed very far from any full understanding of the hardness of approximating MAX-CUT once algorithms are allowed more than one pass over the input distribution. This changed with the recent breakthrough work of Fei, Minzer, and Wang [\[14\]](#), who proved the following amazing result:

Theorem 5.1 (Fei, Minzer, and Wang [\[14\]](#)). *For every $\epsilon > 0$, every k -pass, s -space streaming algorithm which $(\frac{1}{2} + \epsilon)$ -approximates MAX-CUT has $ks = \Omega(\sqrt[3]{n})$.*

The proof of [\[14\]](#) introduces some very novel ideas to the study of streaming CSP approximations, including an argument which formalizes some folklore intuition about streaming algorithms for MAX-CUT : Optimal algorithms essentially just use their memory space to remember increasingly large connected components in the graph, and then search for odd-length cycles in these components as they keep seeing additional edges. Thus, the task of proving lower bounds against arbitrary algorithms morally reduces to proving lower bounds only against these algorithms.⁴

There are numerous interesting questions following up on [\[14\]](#). For instance, it is not clear at all what happens once we allow $\omega(\sqrt[3]{n})$ space and $O(1)$ passes. For starters, we conjecture the following, which would generalize the $o(n)$ -space lower bound for a single pass in [Theorem 4.1](#):

⁴[\[27\]](#) also includes a very nice analysis of these “component-growing” algorithms in the single-pass setting. It would be very interesting if the reduction to component-growing protocols in [\[14\]](#) could be reworked into the single-pass setting, giving a simpler proof of the [\[27\]](#) result for general algorithms.

Conjecture 3

For every $\epsilon > 0$, every two-pass, adversarial-order streaming algorithm which $(\frac{1}{2} + \epsilon)$ -approximates MAX-CUT uses $\Omega(n)$ space.

It is also interesting to consider how crucial the $\sqrt[3]{n}$ -space threshold in [Theorem 5.1](#) is. Perhaps one could prove the following:

Conjecture 4

For every $\epsilon > 0$, every k -pass, s -space streaming algorithm which $(\frac{1}{2} + \epsilon)$ -approximates MAX-CUT has $ks = \Omega(\sqrt{n})$.

However, to my current knowledge, there are multiple places where the [\[14\]](#) argument breaks beyond $\sqrt[3]{n}$ space, and so proving [Conjecture 4](#) may be very hard.

Remark 5.2. There is a folklore result which shows that the hard distributions $\mathcal{D}_{\text{Yes}}$ and $\mathcal{D}_{\text{No}}$ used in [\[14\]](#) to prove [Theorem 5.1](#) (which are roughly the same instances as those used in [\[28, 27\]](#) to prove [Theorems 4.1](#) and [4.2](#)) are actually *distinguishable* in $\tilde{O}(\sqrt{n})$ space and $\tilde{O}(1)$ passes. Very roughly, in this regime, one can take $O(\sqrt{n})$ random walks of length $O(\log n)$ in the input graph and find odd-length cycles in $\mathcal{D}_{\text{No}}$ via looking at collisions among the walks' endpoints. This is why our [Conjecture 4](#) goes only up to the $\sqrt{n}$ threshold. $\diamond$

Beyond $\sqrt{n}$ space, it is much less clear what should happen. One reasonably safe conjecture might be the following:

Conjecture 5

For every $C > 0$, there exists some $\epsilon > 0$ such that every streaming algorithm which $(1 - \epsilon)$ -approximates MAX-CUT uses $\Omega(n^C)$ passes or $\Omega(n)$ space.

See also [\[42, Rmk. 1.5\]](#) for discussion on semidefinite-programming-based multi-pass algorithms for MAX-CUT.

6 More $o(\sqrt{n})$ -space streaming lower bounds

It turns out that MAX-DICUT behaves very differently than MAX-CUT in the streaming setting: It admits nontrivial approximations, while MAX-CUT does not. Some intuition for this is the following:

Remark 6.1. A directed graph $\mathcal{G}$ is satisfiable for MAX-DICUT iff every vertex has either all outgoing or all incoming edges. Thus, it is easy to detect *locally* whether a MAX-DICUT instance is not perfectly satisfiable, i.e., by just looking at the neighborhood of every vertex independently. MAX-CUT does not have such a nice characterization: A graph $\mathcal{G}$ is bipartite (a.k.a., is perfectly satisfiable for MAX-CUT) iff it contains no *odd cycles*, and so certifying unsatisfiability for MAX-CUT requires finding an odd-length cycle, which, in a sparse graph, might have length $\Omega(\log n)$. Thus, very roughly, it is possible to “reason locally” about MAX-DICUT, while MAX-CUT requires an algorithm to “reason globally”. $\diamond$

Building on [\[19\]](#), Chou, Golovnev, and Velusamy [\[7\]](#) proved the following characterization for MAX-DICUT:

Theorem 6.2 (Chou, Golovnev, and Velusamy [\[7\]](#)). *For every $\epsilon > 0$, there is an $O(\log n)$ -space sketching algorithm which $(\frac{4}{9} - \epsilon)$ -approximates MAX-DICUT, but every streaming algorithm which $(\frac{4}{9} + \epsilon)$ -approximates MAX-DICUT uses $\Omega(\sqrt{n})$ space.*

Here the pesky $o(\sqrt{n})$ -space threshold pops up again.⁵ Chou, Golovnev, Sudan, and Velusamy [\[11\]](#)

⁵The [\[7\]](#) algorithm is based on a quantitative form of the observation in [Remark 6.1](#). It measures a quantity called the

generalized the [7] result into a *dichotomy theorem* between $\tilde{O}(1)$ and $o(\sqrt{n})$ -space for *sketching* algorithms for *all* CSPs (!):

Theorem 6.3 (Chou, Golovnev, Sudan, and Velusamy [11]). *For every $k \in \mathbb{N}$, predicate family $\Pi \subseteq (\{0, 1\}^k)^{\{0, 1\}}$, and $\alpha \in [0, 1]$, either:*

1. *For every $\epsilon > 0$, there is a sketching algorithm $(\alpha - \epsilon)$ -approximating $\text{MAX-CSP}(\Pi)$ in $O(\text{polylog } n)$ space.*
2. *For every $\epsilon > 0$, every sketching algorithm which $(\alpha + \epsilon)$ -approximates $\text{MAX-CSP}(\Pi)$ uses $\Omega(\sqrt{n})$ space.*

Remark 6.4. [11] also describes an algorithm for deciding whether Item 1 or Item 2 applies, which runs in polynomial space in the relevant parameters. $\diamond$

Note that the general lower bound (Item 2 in Theorem 6.3) only holds against *sketching* algorithms. The authors of [11] also provide technical conditions under which lower bounds hold more generally against streaming algorithms. These conditions recover all previously known $o(\sqrt{n})$ -space streaming lower bounds ([28, 18, 7]), but we appear far from knowing whether Item 2 holds against streaming algorithms for all Π and α . Hence, it makes sense to examine some CSPs for which the currently-known sketching lower bounds (à la [11]) are stronger than the currently-known streaming lower bounds.

For instance, using the [11] characterization, Boyland, Hwang, Prasad, Singer, and Velusamy [4] proved the following:

Theorem 6.5 (Boyland, Hwang, Prasad, Singer, and Velusamy [4]). *For every $\epsilon > 0$, there is an $O(\log n)$ -space sketching algorithm which $(\frac{2}{9} - \epsilon)$ -approximates MAX-3AND , but every sketching algorithm which $(\frac{2}{9} + \epsilon)$ -approximates MAX-3AND uses $\Omega(\sqrt{n})$ space.*

A natural follow-up conjecture (which did appear in [4]) is the following:

Conjecture 6

For every $\epsilon > 0$, every single-pass streaming algorithms which $(\frac{2}{9} + \epsilon)$ -approximates MAX-3AND uses $\Omega(\sqrt{n})$ space.

In fact, [4] contains a general theorem of this form with an explicit constant for $\text{MAX-}k\text{AND}$ for every k .

Remark 6.6. [4] does use the [11] technical condition to show a streaming lower bound for $(\frac{2}{9} + \epsilon)$ -approximating MAX-3AND when $\epsilon > 0.0141$, but they also show that the condition *cannot* give a full $\frac{2}{9}$ -approximation lower bound. $\diamond$

Remark 6.7. I am aware of unpublished work of Raghuvansh Saxena that shows that $\mathcal{D}_{\text{Yes}}$ and $\mathcal{D}_{\text{No}}$ distributions constructed from the procedure in [11] for sketching MAX-3AND are indeed distinguishable via streaming algorithms. [4] shows that this pair of the distribution is the unique “[11]-type” pair giving a $(\frac{2}{9} + \epsilon)$ -approximation sketching lower bound. $\diamond$

Conversely, refuting Conjecture 6 by demonstrating a streaming algorithm which strictly outperformed all sketching algorithms would of course also be very interesting.

A related example is the *monarchy* function

$$k\text{MONARCHY}(x_1, \dots, x_k) := \left(\bigwedge_{i=2}^k x_i \right) \vee \left(x_1 \wedge \left(\bigvee_{i=2}^k x_i \right) \right).^6$$

We define the CSP $\text{MAX-}k\text{MONARCHY} := \text{MAX-CSP}(\Pi)$ with $\Pi := \{k\text{MONARCHY} \circ \text{NOT}_b\}_{b \in \{0, 1\}^k}$. Note that α_{triv} for this CSP is $\frac{1}{2}$. Chou, Golovnev, Shahrashbi, Sudan, and Velusamy [9] studied this (and related) functions vis-a-vis the [11] dichotomy theorem, and showed the following approximation resistance result:

average bias of a directed graph, which detects whether typical vertices have either almost all outgoing or almost all incoming edges.

⁶Think of this as a voting scheme where there is 1 monarch and $k - 1$ subjects. x_1 is the monarch’s vote and each x_i for $i \in \{2, \dots, k\}$ is subject i ’s vote. The vote passes if all the subjects vote affirmatively, or if the monarch votes affirmatively and at least one subject does too.

Theorem 6.8 (Chou, Golovnev, Shahrashbi, Sudan, and Velusamy [9]). *For every $k \geq 5$ and $\epsilon > 0$, every single-pass sketching algorithm which $(\frac{1}{2} + \epsilon)$ -approximates MAX- k MONARCHY uses $\Omega(\sqrt{n})$ space.*

We naturally conjecture an analogue of [Conjecture 6](#) for this problem:

Conjecture 7

For every $k \geq 5$ and $\epsilon > 0$, every single-pass streaming algorithm which $(\frac{1}{2} + \epsilon)$ -approximates MAX- k MONARCHY uses $\Omega(\sqrt{n})$ space.

(See also [42, Rmk. 1.7] for discussion of algorithms for MAX- k MONARCHY which use $o(n)$ space.) Proving (or refuting) these conjectures would go a long way towards understanding the extent to which the [11] dichotomy theorem ([Theorem 6.3](#)) characterizes all *streaming* algorithms, not just sketching algorithms.

7 More lower bounds beyond $o(\sqrt{n})$ space

There are many further interesting questions that pop up once we move into the regime of $\tilde{O}(\sqrt{n})$ and beyond space (but still $o(n)$). In this regime, we do have strong streaming lower bounds for MAX-CUT ([Theorem 4.1](#) due to [27]) and more generally for MAX-CSP(Π) when Π has the so-called “wideness” property [10]. At the same time, $\tilde{O}(\sqrt{n})$ space is enough to enable some improved approximations, in particular for MAX-DICUT [37, 36] and probably also for MAX- k AND [41]. Towards the question of strong $o(n)$ -space lower bounds, a very strong conjecture would be the following:

Conjecture 8

Every predicate family Π which cannot be nontrivially approximated by $o(\sqrt{n})$ -space sketching algorithms (à la [11]) also cannot be nontrivially approximated by $o(n)$ -space streaming algorithms.

We would not be surprised if this conjecture is false, but having simple counterexamples would also help orient future study on these types of problems.

Towards the question of MAX-DICUT approximability, Saxena, Singer, Sudan, and Velusamy [36] (building on their earlier work [37], and combined with some numerical work due to [16, 41, 21]) proved the following theorem:

Theorem 7.1 (Saxena, Singer, Sudan, and Velusamy [36]). *There is a 0.485-approximation single-pass $\tilde{O}(\sqrt{n})$ -space adversarial-ordering streaming algorithm for MAX-DICUT.*

Note that this is strictly better than the $O(\log n)$ -space $(\frac{4}{9} - \epsilon)$ -approximations from [Theorem 6.2](#), which were optimal in $o(\sqrt{n})$ space [7].

The [36] algorithm relies on the notion of “oblivious algorithms” from [16]. For these “oblivious” algorithms, lower bounds (at roughly 0.49) were constructed in [16] and improved in [21].⁷ It is natural to ask whether one can do better in $o(n)$ space; Saxena, Singer, Sudan, and Velusamy [35] showed recently that this is possible:

Theorem 7.2 (Saxena, Singer, Sudan, and Velusamy [35]). *For every $\epsilon > 0$ and $D \in \mathbb{N}$, there is some $\delta > 0$ and a $(\frac{1}{2} - \epsilon)$ -approximation single-pass $\tilde{O}(n^{1-\delta})$ -space sketching algorithm for MAX-DICUT on graphs with maximum degree D .*

The maximum degree assumption here is a technical condition which, I believe, can be removed, though it might be quite annoying to do so. But is the tending-towards-linear space dependence in [Theorem 7.2](#) necessary? That is, could one even achieve $(\frac{1}{2} + \epsilon)$ -approximations for all $\epsilon > 0$ in $\tilde{O}(\sqrt{n})$ space? We conjecture that this is not possible:

⁷The optimal approximation ratio achievable by oblivious algorithms is not currently known; the current best word is due to Hwang, Singer, and Velusamy [21], who show that the constant is in the interval $[0.4853, 0.4889]$. Finding (or characterizing) the optimal ratio is an interesting open question.

Conjecture 9

There exist constants $\epsilon, \delta > 0$ such that every single-pass sketching algorithm which $(\frac{1}{2} - \epsilon)$ -approximates MAX-DICUT uses $\Omega(n^{\frac{1}{2} + \delta})$.

If this is true, one could even imagine a whole hierarchy of upper and lower bounds as the approximation ratio tends to $\frac{1}{2}$ and the space usage tends to $\Theta(n)$: That is, perhaps, for every $\delta > 0$, there exists $\epsilon > 0$ such that $(\frac{1}{2} - \epsilon)$ -approximating MAX-DICUT is hard in $o(n^{1-\delta})$ space.

References

- [1] Sepehr Assadi and Vishvajeet N. Graph streaming lower bounds for parameter estimation and property testing via a streaming XOR lemma. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing (STOC 2021)*, pages 612–625, virtual, June 2021. Association for Computing Machinery. doi: 10.1145/3406325.3451110.
- [2] 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 *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS 2020)*, pages 354–364, virtual, November 2020. IEEE Computer Society. doi: 10.1109/FOCS46700.2020.00041.
- [3] Aditya Bhaskara, Samira Daruki, and Suresh Venkatasubramanian. Sublinear Algorithms for MAXCUT and Correlation Clustering. In *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*, volume 107 of *LIPIcs*, pages 16:1–16:14, Prague, Czech Republic, 2018. Schloss Dagstuhl — Leibniz-Zentrum für Informatik. doi: 10.4230/LIPICS.ICALP.2018.16.
- [4] Joanna Boyland, Michael Hwang, Tarun Prasad, Noah Singer, and Santhoshini Velusamy. On sketching approximations for symmetric Boolean CSPs. In Amit Chakrabarti and Chaitanya Swamy, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX 2022)*, volume 245 of *LIPIcs*, pages 38:1–38:23, virtual, July 2022. Schloss Dagstuhl — Leibniz-Zentrum für Informatik. doi: 10.4230/LIPICS.APPROX/RANDOM.2022.38.
- [5] Joshua Brakensiek, Neng Huang, Aaron Potechin, and Uri Zwick. Separating MAX 2-AND, MAX DI-CUT and MAX CUT. In *Proceedings of the 64th Annual Symposium on Foundations of Computer Science (FOCS 2023)*, Santa Cruz, CA, USA, April 2023.
- [6] Lijie Chen, Gillat Kol, Dmitry Paramonov, Raghuvansh Saxena, Zhao Song, and Huacheng Yu. Towards Multi-Pass Streaming Lower Bounds for Optimal Approximation of Max-Cut. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2023)*, Florence, Italy, 2023. doi: 10.1137/1.9781611977554.ch35.
- [7] Chi-Ning Chou, Alexander Golovnev, and Santhoshini Velusamy. Optimal Streaming Approximations for all Boolean Max-2CSPs and Max- k SAT. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS 2020)*, pages 330–341, virtual, November 2020. IEEE Computer Society. doi: 10.1109/FOCS46700.2020.00039.
- [8] Chi-Ning Chou, Alexander Golovnev, Madhu Sudan, and Santhoshini Velusamy. Approximability of all Boolean CSPs with linear sketches. February 2021. URL <https://arxiv.org/abs/2102.12351>.
- [9] Chi-Ning Chou, Alexander Golovnev, Amirbehshad Shahrashbi, Madhu Sudan, and Santhoshini Velusamy. Sketching Approximability of (Weak) Monarchy Predicates. In Amit Chakrabarti and Chaitanya Swamy, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX 2022)*, volume 245 of *LIPIcs*, pages 35:1–35:17, virtual, 2022. Schloss Dagstuhl — Leibniz-Zentrum für Informatik. doi: 10.4230/LIPICS.APPROX/RANDOM.2022.35.

- [10] Chi-Ning Chou, Alexander Golovnev, Madhu Sudan, Ameya Velingker, and Santhoshini Velusamy. Linear Space Streaming Lower Bounds for Approximating CSPs. In *Proceedings of the 54th Annual ACM Symposium on Theory of Computing (STOC 2022)*, Rome, Italy, 2022. Association for Computing Machinery. doi: 10.1145/3519935.3519983.
- [11] Chi-Ning Chou, Alexander Golovnev, Madhu Sudan, and Santhoshini Velusamy. Sketching Approximability of All Finite CSPs. *Journal of the ACM*, 71(2):15:1–15:74, April 2024. doi: 10.1145/3649435.
- [12] N. Creignou. A Dichotomy Theorem for Maximum Generalized Satisfiability Problems. *Journal of Computer and System Sciences*, 51(3):511–522, December 1995. ISSN 00220000. doi: 10.1006/jcss.1995.1087. URL <https://linkinghub.elsevier.com/retrieve/pii/S0022000085710872>.
- [13] Irit Dinur. The PCP theorem by gap amplification. *Journal of the ACM*, 54(3):12, June 2007. ISSN 0004-5411, 1557-735X. doi: 10.1145/1236457.1236459. URL <https://dl.acm.org/doi/10.1145/1236457.1236459>.
- [14] Yumou Fei, Dor Minzer, and Shuo Wang. Multi-Pass Streaming Lower Bounds for Approximating Max-Cut. In *Proceedings of the 66th IEEE Symposium on Foundations of Computer Science (FOCS 2025)*, Sydney, Australia, March 2025. IEEE Computer Society.
- [15] Yumou Fei, Dor Minzer, and Shuo Wang. A Dichotomy Theorem for Multi-Pass Streaming CSPs, September 2025.
- [16] Uriel Feige and Shlomo Jozeph. Oblivious Algorithms for the Maximum Directed Cut Problem. *Algorithmica*, 71(2):409–428, February 2015. doi: 10.1007/s00453-013-9806-z.
- [17] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, November 1995. doi: 10.1145/227683.227684.
- [18] Venkatesan Guruswami and Runzhou Tao. Streaming Hardness of Unique Games. In Dimitris Achlioptas and László A. Végh, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX 2019)*, volume 145 of *LIPIcs*, pages 5:1–5:12, Cambridge, MA, USA, September 2019. Schloss Dagstuhl — Leibniz-Zentrum für Informatik. doi: 10.4230/LIPIcs.APPROX-RANDOM.2019.5.
- [19] Venkatesan Guruswami, Ameya Velingker, and Santhoshini Velusamy. Streaming Complexity of Approximating Max 2CSP and Max Acyclic Subgraph. In Klaus Jansen, José D. P. Rolim, David Williamson, and Santosh S. Vempala, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX 2017)*, volume 81 of *LIPIcs*, pages 8:1–8:19, Berkeley, CA, USA, August 2017. Schloss Dagstuhl — Leibniz-Zentrum für Informatik. doi: 10.4230/LIPIcs.APPROX-RANDOM.2017.8.
- [20] Johan Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, 2001. doi: 10.1145/502090.502098.
- [21] Samuel Hwang, Noah G. Singer, and Santhoshini Velusamy. Oblivious algorithms for Maximum Directed Cut: New upper and lower bounds. Technical report, May 2024. URL <https://arxiv.org/abs/2411.12976>.
- [22] Piotr Indyk, Andrew McGregor, Ilan Newman, and Krzysztof Onak. Open problems in data streams, property testing, and related topics, June 2011. URL https://sublinear.info/files/bertinoro2011_kanpur2009.pdf.
- [23] Brigitte Jaumard and Bruno Simeone. On the complexity of the maximum satisfiability problem for horn formulas. *Information Processing Letters*, 26(1):1–4, September 1987. ISSN 00200190. doi: 10.1016/0020-0190(87)90028-7. URL <https://linkinghub.elsevier.com/retrieve/pii/0020019087900287>.

- [24] John Kallaugher and Ojas Parekh. The Quantum and Classical Streaming Complexity of Quantum and Classical Max-Cut. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS 2022)*, pages 498–506, Denver, CO, USA, October 2022. IEEE Computer Society. doi: 10.1109/FOCS54457.2022.00054.
- [25] John Kallaugher, Ojas Parekh, and Nadezhda Voronova. Exponential Quantum Space Advantage for Approximating Maximum Directed Cut in the Streaming Model. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing (STOC 2024)*, pages 1805–1815, Vancouver, BC, Canada, June 2024. Association for Computing Machinery. doi: 10.1145/3618260.3649709.
- [26] John Kallaugher, Ojas Parekh, and Nadezhda Voronova. How to Design a Quantum Streaming Algorithm Without Knowing Anything About Quantum Computing. In *2025 Symposium on Simplicity in Algorithms (SOSA 2025)*, pages 9–45, New Orleans, LA, USA, January 2025. Society for Industrial and Applied Mathematics. doi: 10.1137/1.9781611978315.2.
- [27] Michael Kapralov and Dmitry Krachun. An optimal space lower bound for approximating MAX-CUT. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing (STOC 2019)*, pages 277–288, Phoenix, AZ, USA, June 2019. Association for Computing Machinery. doi: 10.1145/3313276.3316364.
- [28] Michael Kapralov, Sanjeev Khanna, and Madhu Sudan. Streaming lower bounds for approximating MAX-CUT. In *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2015)*, pages 1263–1282, San Diego, CA, USA, January 2015. Society for Industrial and Applied Mathematics. doi: 10.1137/1.9781611973730.84.
- [29] Michael Kapralov, Sanjeev Khanna, Madhu Sudan, and Ameya Velingker. $(1 + \omega(1))$ -approximation to MAX-CUT requires linear space. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2017)*, pages 1703–1722, Barcelona, Spain, January 2017. Society for Industrial and Applied Mathematics. doi: 10.5555/3039686.3039798.
- [30] Subhash Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC 2002)*, pages 767–775, Québec, Canada, May 2002. Association for Computing Machinery. doi: 10.1145/509907.510017.
- [31] Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O’Donnell. Optimal Inapproximability Results for MAX-CUT and Other 2-Variable CSPs? *SIAM Journal on Computing*, 37(1), May 2007. doi: 10.1137/S0097539705447372.
- [32] Dmitry Kogan and Robert Krauthgamer. Sketching cuts in graphs and hypergraphs. In *Proceedings of the 6th Annual Conference on Innovations in Theoretical Computer Science (ITCS 2015)*, pages 367–376, Rehovot, Israel, 2015. Association for Computing Machinery. doi: 10.1145/2688073.2688093.
- [33] Gillat Kol, Dmitry Paramonov, Raghuvansh R. Saxena, and Huacheng Yu. Characterizing the Multi-Pass Streaming Complexity for Solving Boolean CSPs Exactly. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*, volume 251 of *LIPIcs*, pages 80:1–80:15, Cambridge, MA, USA, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi: 10.4230/LIPIcs.ITCS.2023.80.
- [34] Prasad Raghavendra. Optimal algorithms and inapproximability results for every CSP? In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC 2008)*, pages 245–254, Victoria, BC, Canada, 2008. doi: 10.1145/1374376.1374414.
- [35] Raghuvansh Saxena, Noah G. Singer, Madhu Sudan, and Santhoshini Velusamy. Streaming Algorithms via Local Algorithms for Maximum Directed Cut. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2025)*, pages 3392–3408, New Orleans, LA, USA, 2025. Society for Industrial and Applied Mathematics. doi: 10.1137/1.9781611978322.111.

- [36] Raghuvarsh R. Saxena, Noah Singer, Madhu Sudan, and Santhoshini Velusamy. Improved streaming algorithms for Maximum Directed Cut via smoothed snapshots. In *63rd Annual Symposium on Foundations of Computer Science (FOCS 2023)*, pages 855–870, Santa Cruz, CA, USA, 2023. IEEE Computer Society. doi: 10.1109/FOCS57990.2023.00055.
- [37] Raghuvarsh R. Saxena, Noah G. Singer, Madhu Sudan, and Santhoshini Velusamy. Streaming complexity of CSPs with randomly ordered constraints. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2023)*, pages 4083–4103, Florence, Italy, January 2023. Society for Industrial and Applied Mathematics. doi: 10.1137/1.9781611977554.ch156.
- [38] Thomas J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the 10th Annual ACM Symposium on Theory of Computing (STOC 1978)*, pages 216–226, San Diego, CA, USA, May 1978. Association for Computing Machinery. doi: 10.1145/800133.804350.
- [39] Noah Singer. *On Streaming Approximation Algorithms for Constraint Satisfaction Problems*. Undergraduate thesis, Harvard University, Cambridge, MA, March 2022. URL <https://nrs.harvard.edu/URN-3:HUL.INSTREPOS:37371750>.
- [40] Noah Singer, Madhu Sudan, and Santhoshini Velusamy. Streaming approximation resistance of every ordering CSP. *Computational Complexity*, 33:6, 2024. doi: 10.1007/s00037-024-00252-5.
- [41] Noah G. Singer. Oblivious algorithms for the Max- k AND problem. In Nicole Megow and Adam D. Smith, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX 2023)*, volume 275 of *LIPIcs*, Atlanta, GA, USA, May 2023. doi: 10.4230/LIPIcs.APPROX/RANDOM.2023.15.
- [42] Noah G. Singer, Madhur Tulsiani, and Santhoshini Velusamy. Sketching approximations and LP approximations for finite CSPs are related, September 2025.
- [43] Madhu Sudan. Streaming and Sketching Complexity of CSPs: A survey (Invited Talk). In Mikołaj Bojańczyk, Emanuela Merelli, and David P. Woodruff, editors, *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022)*, volume 229 of *LIPIcs*, pages 5:1–5:20, Paris, France, 2022. Schloss Dagstuhl — Leibniz-Zentrum für Informatik. doi: 10.4230/LIPIcs.ICALP.2022.5.
- [44] Xiaoming Sun and David P. Woodruff. Tight Bounds for Graph Problems in Insertion Streams. In Naveen Garg, Klaus Jansen, Anup Rao, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2015)*, volume 40 of *LIPIcs*, pages 435–448, Princeton, NJ, USA, 2015. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi: 10.4230/LIPIcs.APPROX-RANDOM.2015.435.
- [45] Luca Trevisan, Gregory B. Sorkin, Madhu Sudan, and David P. Williamson. Gadgets, Approximation, and Linear Programming. *SIAM Journal on Computing*, 29(6):2074–2097, January 2000. doi: 10.1137/S0097539797328847.
- [46] Santhoshini Velusamy. *Approximability of Constraint Satisfaction Problems in the Streaming Setting*. PhD thesis, Harvard University, Cambridge, MA, May 2023. URL <https://nrs.harvard.edu/URN-3:HUL.INSTREPOS:37377430>.
- [47] Mariano Zelke. Intractability of min- and max-cut in streaming graphs. *Information Processing Letters*, 111(3):145–150, January 2011. doi: 10.1016/j.ipl.2010.10.017.