Characterizing Streaming Decidability of CSPs via Non-Redundancy

Amatya Sharma
University of Michigan, Ann Arbor, MI, USA

Santhoshini Velusamy
University of Waterloo, Canada

Abstract

We study the single-pass streaming complexity of deciding satisfiability of Constraint Satisfaction Problems (CSPs). A CSP is specified by a constraint language Γ, that is, a finite set of k-ary relations over the domain $[q] = \{0, \ldots, q-1\}$. An instance of $\mathsf{CSP}(\Gamma)$ consists of m constraints over n variables $x_1, \ldots, x_n$ taking values in $[q]$. Each constraint C_i is of the form $\{R_i, (x_{i_1} + \lambda_{i_1}, \ldots, x_{i_k} + \lambda_{i_k})\}$, where $R_i \in \Gamma$ and $\lambda_{i_1}, \ldots, \lambda_{i_k} \in [q]$ are constants; it is satisfied if and only if $(x_{i_1} + \lambda_{i_1}, \ldots, x_{i_k} + \lambda_{i_k}) \in R_i$, where addition is modulo q. In the streaming model, constraints arrive one by one, and the goal is to determine, using minimum memory, whether there exists an assignment satisfying all constraints.

For k-SAT, Vu (TCS 2024) proves an optimal $\Omega_k(n^k)$ space lower bound, while for general CSPs, Chou, Golovnev, Sudan, and Velusamy (JACM 2024) establish an $\Omega(n)$ lower bound; a complete characterization has remained open. We close this gap by showing that the single-pass streaming space complexity of $\mathsf{CSP}(\Gamma)$ is precisely governed by its *non-redundancy*, a structural parameter introduced by Bessiere, Carbonnel, and Katsirelos (AAAI 2020). The non-redundancy $\mathsf{NRD}_n(\Gamma)$ is the maximum number of constraints over n variables such that every constraint is *non-redundant*, i.e., omitting it strictly expands the set of satisfying assignments. We prove that the single-pass streaming complexity of $\mathsf{CSP}(\Gamma)$ is characterized, up to a logarithmic factor, by $\mathsf{NRD}_n(\Gamma)$. We also extend this characterization to *positive* Boolean CSPs, i.e., instances in which no additive shifts are applied, a class that includes graph 2-colorability (equivalently, bipartiteness) as a canonical example.

A key ingredient in our lower bound proof is a binary relation $\mathcal{E}$ on the set of assignments $[q]^n$, where $(a, b) \in \mathcal{E}$ if every instance satisfied by a is also satisfied by b. While it is immediate that $\mathcal{E}$ is reflexive and transitive, its symmetry, which would make it an equivalence relation, is non-trivial. We show that $\mathcal{E}$ is an equivalence relation for general CSPs, and for positive Boolean CSPs when excluding the two *constant assignments* (0^n and 1^n). We believe this equivalence structure could be of independent interest.

2012 ACM Subject Classification Theory of computation → Streaming models; Theory of computation → Lower bounds and information complexity; Theory of computation → Streaming, sublinear and near linear time algorithms

Keywords and phrases Constraint Satisfaction Problems, Streaming Algorithms, Satisfiability, Non-Redundancy

Digital Object Identifier 10.4230/LIPIcs.ESA.2026.131

Related Version *Full Version*: `http://arxiv.org/abs/2604.21922`

Funding *Amatya Sharma*: Work done in part when the author was visiting the Toyota Technological Institute at Chicago as an intern in Fall 2025.
Santhoshini Velusamy: Supported by NSERC Discovery grant RGPIN-2026-04689 and ECR supplement DGECR-2026-00421. Supported in part by NSF award CCF 2348475 when the author was affiliated with Toyota Technological Institute at Chicago.

Acknowledgements We thank Joshua Brakensiek, Aaron Putterman, and Venkatesan Guruswami for helping us better understand the context of NRD in the literature. We also thank the anonymous reviewers of ESA 2026 for their helpful feedback.

34th Annual European Symposium on Algorithms (ESA 2026).
Editors: Philip Bille, Seth Pettie, and Sabine Storandt; Article No. 131; pp. 131:1–131:20
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Constraint satisfaction problems (CSPs) provide a unifying framework that captures a wide range of computational problems, and serve as a central object of study in theoretical computer science. They have been studied extensively from the viewpoints of complexity theory [59, 38, 28, 36, 48, 16, 20, 65], approximation [47, 46, 58, 7, 50, 19, 33, 4], (approximate) sparsification [51, 31, 17, 44, 45], kernelization and exact sparsification [26, 37, 21, 55, 11, 18, 12, 13, 15], and, more recently, streaming and sketching [41, 35, 42, 24, 22, 61, 52, 64, 23, 62, 29, 30, 8, 63]. Among the most basic objectives for a CSP instance are deciding *satisfiability* [59, 27, 38, 28, 36, 48, 16, 20, 65, 23, 64], maximizing the number of satisfied constraints (Max-CSP) [47, 46, 36, 56, 48, 49, 58, 7, 57, 32, 39, 9, 40, 52, 23], and minimizing the number of unsatisfied constraints[1] (Min-CSP) [46, 10, 2, 25, 43, 34, 64, 6, 5]. In the streaming literature, while Max-CSPs have been extensively studied, with several sharp dichotomy characterizations [24, 52, 23, 29, 63, 8, 30], the satisfiability and Min-CSP variants remain relatively under-explored. In this work, we study the streaming satisfiability problem for finite-domain CSPs.

A constraint satisfaction problem is defined over a finite domain $[q] := \{0, 1, \ldots, q-1\}$ and specified by a finite constraint language Γ consisting of k-ary relations over $[q]$[2]. An instance of $\mathsf{CSP}(\Gamma)$ consists of n variables $x_1, \ldots, x_n$ taking values in $[q]$ and m constraints $C_1, \ldots, C_m$, where each C_i has the form $\{R_i, (x_{i_1} + \lambda_{i_1}, \ldots, x_{i_k} + \lambda_{i_k})\}$ for some relation $R_i \in \Gamma$, k-tuple of variables $(x_{i_1}, \ldots, x_{i_k})$, and constants $\lambda_{i_1}, \ldots, \lambda_{i_k} \in [q]$. An assignment $a \in [q]^n$ satisfies C_i if $(a_{i_1} + \lambda_{i_1}, \ldots, a_{i_k} + \lambda_{i_k}) \in R_i$ (with all arithmetic modulo q), and satisfies the instance if it satisfies every constraint. We say that an instance is satisfiable if there exists an assignment that satisfies it. In the streaming model, constraints arrive one by one, and the goal is to decide if the instance is satisfiable, using as little memory as possible. A deterministic streaming algorithm must succeed on every input, while a randomized algorithm is required to succeed with probability at least 2/3 on every input. Since any instance has at most $O(n^k)$ constraints, storing them all gives a trivial $\widetilde{O}(n^k)$ upper bound. For k-SAT, Vu [64] proved a matching $\Omega_k(n^k)$ lower bound, showing this is essentially tight. For more general CSPs, however, the best known lower bound is $\Omega(n)$, due to Chou, Golovnev, Sudan, and Velusamy [23]. This gap motivates our central question:

Can we completely characterize the single-pass streaming complexity of deciding satisfiability of CSPs*?*

We answer this in the affirmative, obtaining a complete characterization in terms of a structural parameter called *non-redundancy*. For $\mathsf{CSP}(\Gamma)$, the non-redundancy $\mathsf{NRD}_n(\Gamma)$ is the size of the largest instance over n variables in which every constraint is *non-redundant*, i.e., removing any single constraint strictly increases the number of satisfying assignments. Equivalently, every constraint admits a *witness* assignment that violates it but satisfies all other constraints.

As a simple example, $\mathsf{NRD}_n(\text{2-SAT}) = \Omega(n^2)$. Consider a complete graph on n vertices, with each edge (i, j) defining the constraint $x_i \vee x_j$.[3] The assignment $x_i = x_j = 0$ and $x_k = 1$ for all $k \neq i, j$ satisfies every other constraint but violates this one. The same construction also generalizes naturally to k-SAT. Thus, Vu's result is a special case of our main theorem.

[1] Although Max-CSP and Min-CSP coincide in the exact optimization setting, their approximation behaviours are markedly different, as we discuss later.
[2] We think of k and q as constants.
[3] In our notation, 2-SAT is specified by the single binary relation $R = \{(0, 1), (1, 0), (1, 1)\}$, and an OR constraint such as $x_i \vee \neg x_j$ is expressed as $\{R, (x_i, x_j + 1)\}$, where the shift of 1 on x_j encodes the negation $\neg x_j$.

▶ **Theorem 1** (Main result). *For every constraint language* Γ *containing* k*-ary relations over* $[q]$*, to solve the satisfiability of* $\mathsf{CSP}(\Gamma)$*, there exists:*
(a) *a deterministic streaming algorithm in* $O_{q,k}(\mathsf{NRD}_n(\Gamma)\log n)$ *space, and*
(b) *no randomized streaming algorithm in* $o(\mathsf{NRD}_n(\Gamma))$ *space.*

Our characterization immediately converts known non-redundancy bounds into streaming decidability bounds. For example, k-SAT has $\mathsf{NRD}_n = \Theta(n^k)$, recovering Vu's $\widetilde{\Theta}(n^k)$ bound. Affine languages such as k-LIN have $\mathsf{NRD}_n = \Theta(n)$, and hence satisfiability is decidable in near-linear space. This is in contrast with the exact optimization version: deciding whether a Boolean Max-k-LIN instance has an assignment satisfying at least a given threshold, equivalently computing the optimum value exactly, has $\widetilde{\Theta}(n^k)$ streaming space complexity [52]. For graph q-coloring, equivalently the CSP over the binary relation $\mathsf{NEQ}_q = \{(a,b) \in [q]^2 : a \neq b\}$, one has $\mathsf{NRD}_n = \Theta(n^2)$ for $q \geq 3$, giving a quadratic streaming lower bound. More generally, all the known non-redundancy bounds and reduction frameworks for predicates such as OR/k-SAT, affine predicates/k-LIN, disequality/coloring, cut/NAE-type predicates, and low-arity symmetric Boolean predicates immediately translate into corresponding streaming decidability bounds through our theorem [11, 18, 12, 13, 60, 14].

Non-redundancy also plays a central role in the sparsification theory of CSPs. The concept was first introduced by Bessiere, Carbonnel, and Katsirelos [11] in the context of learning CSP instances from membership queries, and has since emerged as a fundamental structural parameter underlying exact sparsification and kernelization [26, 37, 21, 55, 18]. Very recently, it was also shown to govern approximate sparsification up to polylogarithmic factors [12, 13, 15]. Our work adds to this growing line of research by showing that non-redundancy also characterizes the streaming complexity of deciding CSP satisfiability.

We also extend our characterization to *positive* Boolean CSPs, where constraints are applied only to variables and not their negations. This well-studied special class captures problems such as 2-colorability that fall outside the previous definition. While the upper bound follows immediately from Theorem 1, the lower bound requires a different argument, as constraints can no longer be applied to negated variables. An additional subtlety arises in this setting: without literals, satisfiability can become trivial for some constraint languages. For example, consider positive 2-LIN, which contains the single binary relation $R = \{(1,1),(0,0)\}$. Although $\mathsf{NRD}_n(\text{2-LIN}) = \Theta(n)$, every positive instance of 2-LIN is trivially satisfied by the constant assignments 0^n and 1^n. Excluding *constant-satisfiable* languages, where every instance is satisfied by some constant assignment, we still obtain a tight characterization based on non-redundancy. We formally state our result below.

▶ **Theorem 2.** *For every non-constant-satisfiable Boolean constraint language* Γ *containing* k*-ary Boolean relations, no randomized streaming algorithm can decide satisfiability of positive* $\mathsf{CSP}(\Gamma)$ *in* $o(\mathsf{NRD}_n(\Gamma))$ *space.*

Here and throughout, $\mathsf{NRD}_n(\Gamma)$ denotes non-redundancy in the general model allowing literals. For positive Boolean CSPs, this causes no loss: by a model-robustness result of [60], non-redundancy in the literal, positive, and simple positive models agree up to constant-factor changes in the number of variables. Thus a lower bound in terms of $\mathsf{NRD}_n(\Gamma)$ may be proved starting from a positive non-redundant instance of asymptotically the same size.

We now discuss how the different CSP objectives relate to one another and how our results compare to those obtained under other objectives.

We first consider the Max-CSP objective, where the goal is to find an assignment maximizing the number of satisfied constraints. In the polynomial-time setting, satisfiability can be significantly easier than Max-CSP. For example, Max-2-SAT is NP-hard while satisfiability of a 2-SAT instance is decidable in polynomial time. Our results show for the first time that an analogous phenomenon occurs in the streaming setting, though interestingly not for 2-SAT.[4] Specifically, Kol, Paramonov, Saxena, and Yu [52] give a complete characterization of the streaming complexity of Max-CSPs, showing that for Max-k-LIN, the CSP where constraints correspond to linear equations over $\mathbb{F}_2$, the space complexity is $\Theta(n^k)$, up to logarithmic factors. In contrast, we show that satisfiability of k-LIN can be decided deterministically in $\widetilde{O}(n)$ space.

Next, we consider the Min-CSP objective, where the goal is to find an assignment minimizing the number of unsatisfied constraints. Although Min-CSP and Max-CSP are equivalent in the *exact* setting, since minimizing unsatisfied constraints is the same as maximizing satisfied ones, they differ significantly in the *approximation* setting. For $\alpha \in (0, 1)$, a deterministic α-approximation algorithm for Max-CSP outputs, on every instance, an assignment satisfying at least an α-fraction of the maximum number of satisfiable constraints. A randomized approximation algorithm is defined analogously, except it is only required to succeed with probability at least 2/3 on every input. For Min-CSP, an α-approximation algorithm outputs an assignment that unsatisfies at most a $1/\alpha$-fraction of the minimum number of unsatisfied constraints. Observe that if α is non-zero and the input instance is satisfiable, then the assignment output by the α-approximation algorithm for Min-CSP must itself be satisfying. Thus, Min-CSP is at least as hard as deciding satisfiability, and all our lower bounds immediately carry over to any non-trivial approximation of Min-CSP.

While a standard edge-sampling algorithm achieves an arbitrary constant-factor approximation for Max-CSP in linear space [3], we show for several Min-CSP problems that any non-trivial approximation requires *super-linear* space. For instance, graph q-coloring (equivalently, the relation $x \neq y$ over $[q]$ for $q \geq 3$), as well as other graph and digraph homomorphism problems, satisfy $\mathsf{NRD}_n(\Gamma) = \Theta(n^2)$ [11, 18], and thus admit quadratic space lower bounds.

Finally, we observe that even for approximating Min-k-LIN, our lower bound is essentially tight. Specifically, a streaming algorithm for code sparsification due to Kuchlous [54] can be adapted to obtain $(1 - \epsilon)$-approximation for Min-k-LIN in near-linear space for every constant $\epsilon > 0$. Whether non-redundancy also governs the streaming approximability of general Min-CSPs remains an interesting open problem.

1.1 Technical overview

We now give a brief technical overview of the proofs of our upper and lower bounds.

Upper bound

We give a simple deterministic greedy algorithm that maintains a non-redundant sub-instance with the same set of satisfying assignments as the original instance. Upon the arrival of each constraint, the algorithm checks whether it is redundant with respect to the stored sub-instance, and retains it if not. Since the number of stored constraints cannot exceed the non-redundancy of the CSP, the upper bound follows immediately.

[4] Surprisingly, the streaming complexities of decidability of 2-SAT and Max-2-SAT coincide.

Lower bound

The lower bound proof is technically more involved. The standard approach for proving streaming lower bounds is via a reduction from a hard one-way communication problem. A space-efficient streaming algorithm can be converted into an efficient one-way communication protocol with communication complexity matching the space complexity of the streaming algorithm. We obtain our reduction from one of the most well-studied problems in communication complexity, the INDEX problem. Here, Alice receives an N-bit string S and Bob receives an index $i \in [N]$; Alice sends a single message to Bob, who must then output the bit S_i. It is well-known that any randomized one-way protocol for INDEX that succeeds with probability at least $2/3$ requires $\Omega(N)$ bits of communication [1, 53].

As a starting point, we describe our reduction for 2-SAT. We consider the INDEX problem with $N = \binom{n}{2}$ and interpret Alice's input as a graph G on n vertices and Bob's input as an edge query (i, j). Together, they must determine whether (i, j) is an edge in G. We reduce this to the streaming decidability of 2-SAT as follows. Alice constructs an instance I_G of 2-SAT on n variables by adding the constraint $x_u \vee x_v$ for every edge (u, v) in G. Bob constructs a two-constraint instance $I_{(i,j)}$ consisting of $\neg x_i \vee \neg x_i$ and $\neg x_j \vee \neg x_j$, which is satisfied if and only if $x_i = x_j = 0$. Thus, $I_G \cup I_{(i,j)}$[5] is satisfiable if and only if (i, j) is not an edge in G, and therefore any randomized streaming algorithm deciding satisfiability of 2-SAT must use $\Omega(n^2)$ space. We note that Vu [64] also proves an optimal lower bound for k-SAT via a reduction from the INDEX problem, though his reduction is different and tailored[6] more specifically to k-SAT. The reduction described above, by contrast, generalizes naturally to arbitrary CSPs, as we now describe.

Recall that a complete graph on n vertices, with each edge (i, j) defining the constraint $x_i \vee x_j$, is a non-redundant instance of 2-SAT; call this instance I_{K_n}. Alice's input can thus be viewed as a sub-instance I_G of I_{K_n}, and Bob's input as a constraint C in I_{K_n}. In the reduction, Bob constructs an instance I_C such that $I_G \cup I_C$ is satisfiable if and only if $C \notin I_G$. We show that an analogous reduction is feasible for general CSPs. Specifically, we start with the largest non-redundant instance I_n of $\mathsf{CSP}(\Gamma)$ on n variables, where Alice receives a sub-instance I of I_n and Bob receives a constraint C in I_n. We then demonstrate that Bob can always construct an instance I_C such that $I \cup I_C$ is satisfiable if and only if $C \notin I$. Since I_n is non-redundant, every constraint C in I_n admits a witness assignment $a_C \in [q]^n$ that unsatisfies C but satisfies every other constraint in I_n. If Bob can construct an instance I_C whose *unique* satisfying assignment is a_C, then we are done. While this is possible for 2-SAT, it need not hold for other CSPs. For example, for 2-LIN, if a is a satisfying assignment, then its complement $\overline{a}$ is also satisfying. This is not an obstacle, however, if a is a witness assignment then so is $\overline{a}$, and the same reduction goes through.

The right goal is therefore not to force a_C as the unique satisfying assignment, but to construct an instance I_C whose satisfying assignments are precisely the assignments that behave like a_C on all instances. This is enough for the reduction: if a_C violates C, then every assignment that behaves like a_C also violates C. For general CSPs, this motivates us to define a binary relation $\mathcal{E}$ on the set of assignments $[q]^n$, where $(a, b) \in \mathcal{E}$ if every instance satisfied by a is also satisfied by b. While reflexivity and transitivity are immediate, symmetry, which would make $\mathcal{E}$ an equivalence relation, is not. We establish the following.

---

[5] Here, we are taking the union of all the constraints.

[6] While the reduction in [23] is more general than that of [64], it relies on a different communication problem, namely disjointness, and is incomparable to ours. Moreover, their approach yields only an $\Omega(n)$ lower bound.

▶ **Theorem 3** (informal, restatement of Theorems 9 and 10)**.**
- *For general* CSPs*, $\mathcal{E}$ is an equivalence relation on all assignments;*
- *For positive Boolean* CSPs*, $\mathcal{E}$ is an equivalence relation when restricted to non-constant assignments.*

▶ **Example 4** (Equivalence classes for 2-LIN)**.** For 2-LIN in the general model, an assignment a and its complement $\overline{a}$ satisfy exactly the same constraints. Hence the equivalence class of a under the relation $\mathcal{E}$ defined above is $[a]_{\mathcal{E}} = \{a, \overline{a}\}$. Thus no 2-LIN instance can distinguish between a and $\overline{a}$, and the relevant object for Bob's instance I_C is not the single witness assignment a_C, but the whole assignment class $[a_C]_{\mathcal{E}}$. This is still sufficient for the reduction: if a_C violates the constraint C, then every assignment in $[a_C]_{\mathcal{E}}$ also violates C.

With the above theorem, the reduction for general CSPs follows immediately. Bob constructs an instance I_C, which is the union of all instances that have a_C as a satisfying assignment. By the equivalence structure of $\mathcal{E}$, it is not hard to show that the satisfying assignments of I_C are exactly the equivalence class of a_C, and the reduction follows. The argument is slightly more subtle for positive Boolean CSPs, since the equivalence relation does not hold for all assignments. Nevertheless, this case is manageable. Observe that no two constraints in a non-redundant instance can share the same witness assignment, by definition. Therefore, we can remove the constraints whose witness assignment is 0^n or 1^n, yielding a non-redundant instance of asymptotically the same size, for which the same reduction goes through.

Finally, we show in Section 4.3 that our reductions do not extend to positive CSPs over larger domains, i.e., $q > 2$. The failure is not merely a limitation of our proof technique via the relation $\mathcal{E}$; rather, the lower bound itself is false in general. We provide two counterexamples over non-Boolean domains: one using a non-singleton language (Section 4.3.1) and another using a singleton language (Section 4.3.2), for which single-pass streaming satisfiability is strictly easier than $\mathsf{NRD}_n(\Gamma)$ and the relation $\mathcal{E}$ fails to be symmetric.

1.2 Organization

We begin in Section 2 with notation and basic definitions. In Section 3, we state our main results and the general streaming upper bounds. The lower bound for general CSPs is proved in Section 4.1, where we establish the equivalence-class structure of the relation $\mathcal{E}$ and derive the streaming lower bound from it. We then turn in Section 4.2 to positive Boolean CSPs, proving that $\mathcal{E}$ remains an equivalence relation on non-constant assignments and using this to obtain the corresponding lower bound. In Section 4.3, we show that this phenomenon does not extend to positive CSPs on larger alphabets.

2 Preliminaries and notation

2.1 Constraint languages and instances

Fix a domain $[q] := \{0, 1, \ldots, q-1\}$ for some integer $q \geq 2$. A constraint language Γ over $[q]$ is a set of relations over $[q]$. Let k denote the maximum arity of a relation in Γ. By padding each relation $R \subseteq [q]^\ell$ with $\ell < k$ to $\widehat{R} := R \times [q]^{k-\ell} \subseteq [q]^k$, and using padded variables in constraint scopes, we may assume without loss of generality that every relation in Γ has arity exactly k. Throughout the paper, we only consider finite constraint languages of bounded arity, i.e., languages for which the maximum arity k is bounded by a constant.

An instance I of $\mathsf{CSP}(\Gamma)$ on variables $x_1, \ldots, x_n$ consists of a set of m constraints. For each $i \in [m]$, the constraint C_i is of the form $\{R_i, (x_{i_1} + \lambda_{i_1}, \ldots, x_{i_k} + \lambda_{i_k})\}$, where $R_i \in \Gamma$, $(i_1, \ldots, i_k) \in [n]^k$, and $\lambda_{i_1}, \ldots, \lambda_{i_k} \in [q]$, with all arithmetic modulo q. An assignment $a \in [q]^n$ satisfies such a constraint if $(a_{i_1} + \lambda_{i_1}, \ldots, a_{i_k} + \lambda_{i_k}) \in R_i$. We write $\mathrm{Sat}(I) \subseteq [q]^n$ for the set of satisfying assignments of I. Since duplicate constraints do not affect satisfiability, we identify an instance with the set of its constraints. Unless explicitly stated otherwise, all CSP instances in this paper are stated in this general model.

We call a constraint language Γ *non-trivial* if every relation in Γ is non-empty (i.e., not trivially unsatisfiable) and at least one relation in Γ is proper (i.e., not trivially satisfiable[7]). For streaming satisfiability, this assumption is without loss of generality because constraints using trivially satisfiable relations can simply be ignored, while the appearance of a constraint using an empty relation makes the instance immediately unsatisfiable. Thus, we restrict our attention to non-trivial languages throughout the paper.

Positive CSPs

We call a CSP a *positive* CSP if every constraint C_i is applied with all the shifts λ_{i_j} set to zero, so every constraint has the form $\{R, (x_{i_1}, \ldots, x_{i_k})\}$.

We say that for a positive $\mathsf{CSP}(\Gamma)$, the underlying constraint language Γ is *constant-satisfiable* if there exists some $\alpha \in [q]$ such that $\alpha^k \in R$ for every relation $R \in \Gamma$. Equivalently, the all-α assignment trivially satisfies every instance of positive $\mathsf{CSP}(\Gamma)$. If no such α exists, then Γ is *non-constant-satisfiable*. In our lower bound proof for positive CSPs, we will assume that Γ is non-constant-satisfiable (note this is not a default assumption for general CSPs).

2.2 Equivalent instances, redundancy, and witnesses

Two instances I and J of $\mathsf{CSP}(\Gamma)$ are *equivalent* if $\mathrm{Sat}(I) = \mathrm{Sat}(J)$. A constraint $C \in I$ is *redundant* if deleting it does not change the set of satisfying assignments, that is, $\mathrm{Sat}(I \setminus \{C\}) = \mathrm{Sat}(I)$. Equivalently, C is redundant if every assignment satisfying all constraints in $I \setminus \{C\}$ also satisfies C. We say that I is non-redundant if none of its constraints is redundant.

The main combinatorial parameter in this paper is the maximum size of a non-redundant instance on n variables. For a constraint language Γ, we define

$$\mathsf{NRD}_n(\Gamma) := \max\{|I| : I \text{ is a } \textit{non-redundant instance of } \mathsf{CSP}(\Gamma) \text{ on } n \text{ variables}\}.$$

If $I = \{C_1, \ldots, C_m\}$ is non-redundant, then for each $i \in [m]$ there exists an assignment that satisfies every constraint in $I \setminus \{C_i\}$ but unsatisfies C_i. We call any such assignment a *witness assignment* for C_i. Note that any sub-instance of a non-redundant instance is also non-redundant.

Any non-trivial Γ contains a relation R with $\emptyset \neq R \subsetneq [q]^k$. We can partition the variables into $\lfloor n/k \rfloor$ pairwise disjoint k-tuples, and apply the relation R on each tuple. It is easy to see that each constraint in this set is non-redundant. Thus, we have $\mathsf{NRD}_n(\Gamma) \geq \lfloor n/k \rfloor \geq \Omega_k(n)$.

Model robustness for non-redundancy

Unless stated otherwise, $\mathsf{NRD}_n(\Gamma)$ denotes non-redundancy in the general model defined above, where constraints may use additive shifts. For Boolean languages, this is equivalent, up to constant-factor changes in the number of variables, to the usual positive models. We recall the precise robustness statement from [60, Theorem 6.1].

[7] A relation is trivially satisfiable if it is satisfied by every assignment.

For a Boolean constraint language Γ, let $\mathsf{NRD}_n^{\mathrm{lit}}(\Gamma)$ denote non-redundancy in the literal model, where constraints may be applied to variables, negated variables, and constants. Let $\mathsf{NRD}_n^{+}(\Gamma)$ denote non-redundancy in the positive tuple model, where constraints have the form $\{R, (x_{i_1}, \ldots, x_{i_k})\}$, with repetitions allowed. Let $\mathsf{NRD}_n^{\mathrm{simp}}(\Gamma)$ denote non-redundancy in the simple positive model, where constraints are positive and the variables appearing in each constraint are distinct.

▶ **Theorem 5** (Model robustness of Boolean non-redundancy, [60, Theorem 6.1]). *For every fixed finite Boolean constraint language Γ of bounded arity,*

$$\mathsf{NRD}_n^{\mathrm{simp}}(\Gamma) \le \mathsf{NRD}_n^{+}(\Gamma) \le \mathsf{NRD}_n^{\mathrm{lit}}(\Gamma) \le \mathsf{NRD}_{O_\Gamma(n)}^{\mathrm{simp}}(\Gamma).$$

Consequently, for fixed Boolean languages, the literal, positive tuple, and simple positive models have the same asymptotic non-redundancy up to constant-factor changes in the number of variables.

In particular, throughout our positive Boolean lower bound, we may continue to write $\mathsf{NRD}_n(\Gamma)$ for the non-redundancy parameter and suppress this constant-factor change in the number of variables in asymptotic statements while working with positive non-redundant instances.

2.3 Streaming problems

Streaming Satisfiability

Fix a constraint language Γ. In the single-pass *streaming satisfiability* problem, the input is a stream of constraints of an instance of $\mathsf{CSP}(\Gamma)$ on n variables, and at the end of the stream, the algorithm must decide whether the instance is satisfiable. A deterministic streaming algorithm must succeed on every input, while a randomized algorithm is required to succeed with probability at least $2/3$ on every input.

Streaming Enumeration and Exact Sparsification

We also consider the single-pass *streaming enumeration* problem, in which the algorithm must output the full satisfying set $\mathrm{Sat}(I)$, or output $\emptyset$ if the instance is unsatisfiable. We also consider the single-pass *streaming exact sparsification* problem, in which the algorithm must output, at the end of the stream, an equivalent sub-instance $J \subseteq I$ of the streamed instance I, that is, a sub-instance satisfying $\mathrm{Sat}(J) = \mathrm{Sat}(I)$. The output tape is write-only and does not count toward the workspace.

General notation

In the Boolean setting, the alphabet is $\{0, 1\}$. For $n \ge 1$, we write 0^n and 1^n for the two constant assignments in $\{0, 1\}^n$. An assignment $a \in \{0, 1\}^n$ is *non-constant* if $a \notin \{0^n, 1^n\}$. All asymptotic notations are with respect to the number of variables n. In particular, $\widetilde{O}(\cdot)$, $\widetilde{\Omega}(\cdot)$, and $\widetilde{\Theta}(\cdot)$ suppress factors polylogarithmic in n, and the hidden constants may depend on the fixed alphabet size and constraint language.

3 Main Results and Upper Bounds

We now state our main results. The first theorem gives the complete characterization for general CSPs, up to a logarithmic factor, the single-pass streaming space complexity of deciding satisfiability is exactly governed by the non-redundancy parameter.

▶ **Theorem 1** (Main result). *For every constraint language Γ containing k-ary relations over $[q]$, to solve the satisfiability of* $\mathsf{CSP}(\Gamma)$*, there exists:*
(a) *a deterministic streaming algorithm in $O_{q,k}(\mathsf{NRD}_n(\Gamma)\log n)$ space, and*
(b) *no randomized streaming algorithm in $o(\mathsf{NRD}_n(\Gamma))$ space.*

We next turn to positive Boolean CSPs. In this setting, for non-constant-satisfiable languages, the same bound, tight up to a logarithmic factor, continues to hold.

▶ **Theorem 6** (Main positive Boolean CSP result). *For every non-constant-satisfiable Boolean constraint language Γ containing k-ary Boolean relations, to solve the satisfiability of positive* $\mathsf{CSP}(\Gamma)$ *there exists:*
(a) *a deterministic streaming algorithm in $O_{q,k}(\mathsf{NRD}_n(\Gamma)\log n)$ space; and*
(b) *no randomized streaming algorithm in $o(\mathsf{NRD}_n(\Gamma))$ space.*

We first prove the upper bounds. Throughout the stream, our algorithm is to maintain an equivalent non-redundant sub-instance of the constraints seen so far. The same algorithm works in both settings.

Proof of Theorem 1(a) and Theorem 6(a). Let F_t denote the set of first t constraints seen in the stream. After processing the first t constraints, the algorithm stores a non-redundant instance $K_t \subseteq F_t$ such that $\mathrm{Sat}(K_t) = \mathrm{Sat}(F_t)$. Initially $K_0 = \emptyset$.

When a new constraint C arrives, consider the instance $J := K_t \cup \{C\}$. Repeatedly delete from J any constraint that is redundant with respect to the current instance, and let the resulting non-redundant instance be K_{t+1}. Since we only delete redundant constraints, we have

$$\mathrm{Sat}(K_{t+1}) = \mathrm{Sat}(J) = \mathrm{Sat}(K_t) \cap \mathrm{Sat}(C) = \mathrm{Sat}(F_t) \cap \mathrm{Sat}(C) = \mathrm{Sat}(F_{t+1}).$$

By construction, each K_t is non-redundant, so $|K_t| \le \mathsf{NRD}_n(\Gamma)$. Since each relation in Γ has arity k, each stored constraint uses $O_{q,k}(\log n)$ bits, and therefore storing K_t takes $O_{q,k}(\mathsf{NRD}_n(\Gamma)\log n)$ bits.

It remains to show that each update can be performed within the same space bound. To test whether a constraint $C' \in J$ is redundant, it suffices to check whether there exists an assignment satisfying all constraints in $J \setminus \{C'\}$ while falsifying C'. Since we impose no time bound, we may brute-force over all assignments to the n variables using only a counter and the current assignment, and this requires $O_q(n)$ bits of space. Since $\mathsf{NRD}_n(\Gamma) \ge \Omega_k(n)$, every update can be carried out in $O_{q,k}(\mathsf{NRD}_n(\Gamma))$ space. At the end of the stream, the algorithm stores an equivalent non-redundant sub-instance K_T with $\mathrm{Sat}(K_T) = \mathrm{Sat}(I)$. Therefore I is satisfiable if and only if K_T is satisfiable. We can decide whether $\mathrm{Sat}(K_T) \neq \emptyset$ by brute force within $O_{q,k}(\mathsf{NRD}_n(\Gamma)\log n)$ space. This proves the theorem. ◀

This proves the deterministic upper bound for any general CSP, and thus extends trivially to positive Boolean CSPs as well. The lower bounds are proved in Section 4.1 for general CSPs and Section 4.2 for positive Boolean CSPs.

▶ Remark 7. The algorithm in Theorem 1 solves a more general problem than satisfiability: it maintains an equivalent non-redundant sub-instance throughout the stream. Consequently, it yields a deterministic single-pass streaming algorithm for exact sparsification and streaming enumeration in $O_{q,k}(\mathsf{NRD}_n(\Gamma)\log n)$ space. Conversely, both exact sparsification and enumeration are at least as hard as satisfiability, since an equivalent sub-instance preserves satisfiability exactly, and the full set of satisfying assignments reveals whether the instance is satisfiable. Thus the same $\Omega(\mathsf{NRD}_n(\Gamma))$ lower bound applies to these tasks as well.

4 Lower Bounds

In this section we prove the streaming lower bounds stated in Theorem 1(b) and Theorem 6(b). Both arguments proceed by reduction from the standard one-way communication problem INDEX_m in which Alice receives a string $x \in \{0,1\}^m$, Bob receives an index $i \in [m]$, and Bob must output x_i after receiving a single message from Alice. We use the following standard fact.

▶ **Lemma 8** ([1, 53]). *Every public-coin one-way communication protocol for* INDEX_m *with error probability at most* $1/3$ *requires* $\Omega(m)$ *bits of communication.*

4.1 General CSPs

Our lower bound for $\mathsf{CSP}(\Gamma)$ (Theorem 1(b)) is based on a structural relation on assignments. For $a, b \in [q]^n$, we write $(a, b) \in \mathcal{E}$ if every instance of $\mathsf{CSP}(\Gamma)$ that is satisfied by a is also satisfied by b. This relation is immediately reflexive and transitive, but symmetry is far from obvious. The key point of the next theorem is that, for every constraint language Γ, this relation is in fact an equivalence relation. Moreover, for every equivalence class, we show that there exists an instance whose set of satisfying assignments is exactly that class.

▶ **Theorem 9** (Equivalence Class Theorem for $\mathsf{CSP}(\Gamma)$). *Let $n \in \mathbb{N}$ and Γ be a constraint language containing k-ary relations over the domain $[q]$. For $a, b \in [q]^n$, define $(a, b) \in \mathcal{E}$ if every instance of* $\mathsf{CSP}(\Gamma)$ *on n variables that is satisfied by a is also satisfied by b. Then $\mathcal{E}$ is an equivalence relation on $[q]^n$.*

As a consequence, for every $a \in [q]^n$, there exists an instance P_a of $\mathsf{CSP}(\Gamma)$ *such that* $\mathrm{Sat}(P_a) = [a]_{\mathcal{E}}$*, where $[a]_{\mathcal{E}}$ denotes the equivalence class of a under $\mathcal{E}$.*

Proof. Reflexivity and transitivity are immediate, so it remains to prove symmetry. Fix $a, b \in [q]^n$ with $(a, b) \in \mathcal{E}$. For any $\Delta \in [q]^n$, define the shift operator τ_Δ on constraints by

$$\tau_\Delta\big(\{R, (x_{i_1} + \lambda_{i_1}, \ldots, x_{i_k} + \lambda_{i_k})\}\big) := \{R, (x_{i_1} + \lambda_{i_1} + \Delta_{i_1}, \ldots, x_{i_k} + \lambda_{i_k} + \Delta_{i_k})\},$$

with all additions modulo q, and extend τ_Δ to instances by applying it to each individual constraint in the instance. Then for every assignment z and every instance J, we have that z satisfies $\tau_\Delta(J)$ if and only if $z + \Delta$ satisfies J.

Let $\mathcal{S}_a$ denote the set of all instances of $\mathsf{CSP}(\Gamma)$ on n variables that are satisfied by a, and define $\mathcal{S}_b$ analogously. Now let $\Delta := a - b$. Then for every instance J, we have that a satisfies J if and only if b satisfies $\tau_\Delta(J)$, since $b + \Delta = b + (a - b) = a$. Thus τ_Δ maps $\mathcal{S}_a$ bijectively onto $\mathcal{S}_b$, with inverse $\tau_{-\Delta}$, and therefore $|\mathcal{S}_a| = |\mathcal{S}_b|$.

On the other hand, since $(a, b) \in \mathcal{E}$, every instance satisfied by a is also satisfied by b, so $\mathcal{S}_a \subseteq \mathcal{S}_b$. Because Γ contains k-ary relations, there are only finitely many distinct constraints on n variables, and hence only finitely many instances. Therefore $\mathcal{S}_a$ and $\mathcal{S}_b$ are finite sets. Since they are finite, have the same cardinality, and satisfy $\mathcal{S}_a \subseteq \mathcal{S}_b$, it follows that $\mathcal{S}_a = \mathcal{S}_b$. Hence every instance satisfied by b is also satisfied by a, i.e., $(b, a) \in \mathcal{E}$. Thus $\mathcal{E}$ is symmetric, and therefore an equivalence relation.

For the second claim, fix $a \in [q]^n$. For every $u \notin [a]_{\mathcal{E}}$, we have $(a, u) \notin \mathcal{E}$, so there exists an instance J_u such that a satisfies J_u but u does not. Let $P_a := \bigcup_{u \notin [a]_{\mathcal{E}}} J_u$. Since the domain is finite, there are only finitely many such assignments u. If $c \in [a]_{\mathcal{E}}$, then $(a, c) \in \mathcal{E}$, and since a satisfies every J_u, it follows that c satisfies every J_u, and hence c satisfies P_a. Conversely, if $u \notin [a]_{\mathcal{E}}$, then u does not satisfy J_u, and therefore does not satisfy P_a. Thus $\mathrm{Sat}(P_a) = [a]_{\mathcal{E}}$. ◀

We now specify our reduction from INDEX for the main streaming lower bound. Intuitively, to prove our lower bound, starting from a maximum non-redundant instance, we take for each constraint C_i a witness assignment a_i that satisfies all other constraints but falsifies C_i. We use the above theorem to get the suffix P_{a_i} whose set of satisfying assignments is exactly the assignments that are equivalent to a_i under $\mathcal{E}$. This is precisely what is needed for the reduction below.

Proof of Theorem 1(b). Let $m := \mathsf{NRD}_n(\Gamma)$, and let $I = \{C_1, \ldots, C_m\}$ be a non-redundant instance of $\mathsf{CSP}(\Gamma)$ on n variables with exactly m constraints. For each $i \in [m]$, fix a witness assignment $a_i \in [q]^n$ such that a_i satisfies every constraint in $I \setminus \{C_i\}$ but does not satisfy C_i. We now give a one-way communication protocol for INDEX_m. Recall that Alice is given $x \in \{0,1\}^m$, Bob is given $i \in [m]$, and the goal is to recover x_i. Let $S_x := \{j \in [m] : x_j = 1\}$. Alice streams the sub-instance $I_{S_x} := \{C_j : j \in S_x\}$ through the streaming algorithm and sends Bob the resulting memory state. Bob now appends the suffix P_{a_i} from Theorem 9. If $x_i = 0$, then $i \notin S_x$, so every constraint of I_{S_x} belongs to $I \setminus \{C_i\}$ and is therefore satisfied by a_i. Also, $a_i \in [a_i]_{\mathcal{E}} = \mathrm{Sat}(P_{a_i})$. Hence $I_{S_x} \cup P_{a_i}$ is satisfiable. If $x_i = 1$, then $C_i \in I_{S_x}$. Moreover, every assignment in $\mathrm{Sat}(P_{a_i}) = [a_i]_{\mathcal{E}}$ satisfies exactly the same instances (and hence, constraints) as a_i. Since a_i unsatisfies C_i, it follows that no assignment in $[a_i]_{\mathcal{E}}$ satisfies C_i. Hence no assignment can satisfy $I_{S_x} \cup P_{a_i}$, so this instance is unsatisfiable.

Thus Bob can recover x_i from the output of the streaming algorithm. Therefore any randomized single-pass streaming algorithm for satisfiability yields a public-coin one-way protocol for INDEX_m with the same message length, and thus by Lemma 8, requires $\Omega(m) = \Omega(\mathsf{NRD}_n(\Gamma))$ space. ◀

4.2 Positive Boolean CSPs

We now turn to the proof of Theorem 6(b) that states the lower bound for positive Boolean CSPs. Here the same relation $\mathcal{E}$ is still the right structural object, but the argument is more delicate: in the positive setting, symmetry need not hold automatically. The next theorem shows that, under the natural assumption that Γ is non-constant-satisfiable, the relation $\mathcal{E}$ remains an equivalence relation once we restrict to non-constant assignments.

▶ **Theorem 10** (Non-constant Equivalence Theorem for positive Boolean CSPs). *Let $n \in \mathbb{N}$ and Γ be a non-constant-satisfiable constraint language containing k-ary Boolean relations. For $a, b \in \{0,1\}^n$, define $(a,b) \in \mathcal{E}$ if every instance of the positive* $\mathsf{CSP}(\Gamma)$ *on n variables that is satisfied by a is also satisfied by b. Then the restriction of $\mathcal{E}$ to the non-constant assignments $\{0,1\}^n \setminus \{0^n, 1^n\}$ is an equivalence relation.*

Proof. Reflexivity and transitivity are immediate, so it suffices to prove symmetry. Fix distinct assignments $a, b \in \{0,1\}^n \setminus \{0^n, 1^n\}$ with $(a,b) \in \mathcal{E}$. Define

$$E_0 := \{j \in [n] : a_j = b_j = 0\}, \qquad E_1 := \{j \in [n] : a_j = b_j = 1\}$$

$$D_{01} := \{j \in [n] : a_j = 0,\ b_j = 1\}, \qquad D_{10} := \{j \in [n] : a_j = 1,\ b_j = 0\}.$$

Also define $P(a,b) := \{(a_j, b_j) : j \in [n]\} \subseteq \{0,1\}^2$. By definition, $(a,b) \in \mathcal{E}$ iff for every relation $R \in \Gamma$ of arity k the following holds:

$$\text{For every map } \psi : [k] \to P(a,b), \text{ if } (\psi_1(1), \ldots, \psi_1(k)) \in R, \text{ then } (\psi_2(1), \ldots, \psi_2(k)) \in R,$$

where $\psi_t(\ell)$ denotes the t-th coordinate of the pair $\psi(\ell)$. In words, whenever a tuple in R is obtained by reading the first coordinates of some k pairs from $P(a,b)$, the tuple obtained by reading the corresponding second coordinates must also lie in R.

Consider the following three cases:

1. Suppose first that $D_{01} \neq \emptyset$ and $D_{10} = \emptyset$. Then $E_1 \neq \emptyset$ because a is non-constant, and $E_0 \neq \emptyset$ because b is non-constant and $D_{10} = \emptyset$. Fix any relation $R \in \Gamma$ and any tuple $u \in R$. We claim that $1^k \in R$. If $u \neq 1^k$, choose any coordinate r with $u_r = 0$, and define $\psi : [k] \to P(a, b)$ by setting $\psi(r) := (0, 1)$, setting $\psi(\ell) := (1, 1)$ for $\ell \neq r$ with $u_\ell = 1$, and setting $\psi(\ell) := (0, 0)$ for $\ell \neq r$ with $u_\ell = 0$. Then $(\psi_1(1), \ldots, \psi_1(k)) = u \in R$, so $(\psi_2(1), \ldots, \psi_2(k)) = (u_1, \ldots, u_{r-1}, 1, u_{r+1}, \ldots, u_k) \in R$. Repeating this argument coordinate by coordinate gives $1^k \in R$. Since R was arbitrary, every relation in Γ contains the all-1 tuple, contradicting that Γ is non-constant-satisfiable.
2. Suppose next that $D_{01} = \emptyset$ and $D_{10} \neq \emptyset$. An analogous argument shows that every relation in Γ contains the all-0 tuple, again a contradiction.
3. Therefore the only remaining possibility is that $D_{01} \neq \emptyset$ and $D_{10} \neq \emptyset$. In this case $P(a, b) = P(b, a)$, since both off-diagonal pairs $(0, 1)$ and $(1, 0)$ are present and the diagonal pairs are fixed under swapping. Hence the defining implication for $(a, b) \in \mathcal{E}$ is exactly the defining implication for $(b, a) \in \mathcal{E}$, and therefore $(b, a) \in \mathcal{E}$.

Thus $\mathcal{E}$ is symmetric on $\{0, 1\}^n \setminus \{0^n, 1^n\}$, and its restriction to the non-constant assignments is an equivalence relation. ◀

Although we do not need it for the lower bound, the same argument also gives a simple description of the $\mathcal{E}$-equivalence classes on non-constant assignments.

▶ Remark 11. For every non-constant assignment a, the $\mathcal{E}$-equivalence class of a inside $\{0, 1\}^n \setminus \{0^n, 1^n\}$ is either $[a]_\mathcal{E} = \{a\}$ or $[a]_\mathcal{E} = \{a, \overline{a}\}$.

Proof. Fix distinct non-constant assignments a, b with $(a, b) \in \mathcal{E}$. By the previous argument, both D_{01} and D_{10} must be non-empty. We claim that $E_0 = E_1 = \emptyset$. Indeed, if $E_0 \neq \emptyset$, then using a coordinate from E_0 and one from D_{10} using the same mapping argument as before, we conclude that every non-empty relation in Γ must contain 0^k, contradicting that Γ is non-constant-satisfiable. Similarly, if $E_1 \neq \emptyset$, then using a coordinate from E_1 and one from D_{01} forces every non-empty relation to contain 1^k, again a contradiction. Hence every coordinate lies in either D_{01} or D_{10}, so $b_j = 1 - a_j$ for all $j \in [n]$. Therefore $b = \overline{a}$. ◀

Unlike the general case, this restricted equivalence theorem is not by itself enough for the lower bound, because both the constant assignments must still be excluded explicitly. The next lemma strengthens the theorem in the form needed for the reduction. Given a non-constant witness assignment a falsifying a constraint C, it constructs a suffix $P_{a,C}$ that is satisfied by a and forces every satisfying assignment to falsify C.

▶ **Lemma 12.** *Let C be a constraint in a positive* $\mathsf{CSP}(\Gamma)$ *instance, and let $a \in \{0, 1\}^n \setminus \{0^n, 1^n\}$ be such that a* unsatisfies *C. Then there exists an instance $P_{a,C}$ of positive* $\mathsf{CSP}(\Gamma)$ *such that a* satisfies *$P_{a,C}$ and every assignment satisfying $P_{a,C}$ falsifies C.*

Proof. Since Γ is non-constant-satisfiable, there exist (not necessarily distinct) relations $R^{(0)}, R^{(1)} \in \Gamma$ such that $0^k \notin R^{(0)}$ and $1^k \notin R^{(1)}$. Fix tuples $u \in R^{(0)}$ and $v \in R^{(1)}$. Since a is non-constant, choose coordinates $p, q \in [n]$ with $a_p = 0$ and $a_q = 1$. Let H_a be the instance of positive $\mathsf{CSP}(\Gamma)$ on the two variables x_p, x_q consisting of the following two constraints:

1. the constraint $\{R^{(0)}, (x'_1, \ldots, x'_k)\}$, where $x'_t = x_p$ if $u_t = 0$ and $x'_t = x_q$ if $u_t = 1$;
2. the constraint $\{R^{(1)}, (x''_1, \ldots, x''_k)\}$, where $x''_t = x_p$ if $v_t = 0$ and $x''_t = x_q$ if $v_t = 1$.

Then a satisfies H_a, while neither 0^n nor 1^n satisfies H_a.

Let $[a]_{\mathcal{E}}$ denote the $\mathcal{E}$-equivalence class of a within the set $\{0,1\}^n \setminus \{0^n, 1^n\}$. For every non-constant assignment $b \notin [a]_{\mathcal{E}}$ such that b satisfies C, Theorem 10 implies $(a,b) \notin \mathcal{E}$, so there exists an instance J_b such that a satisfies J_b but b unsatisfies J_b. Define

$$P_{a,C} := H_a \cup \bigcup_{\substack{b \in \{0,1\}^n \setminus \{0^n,1^n\} \\ b \notin [a]_{\mathcal{E}},\ b \text{ satisfies } C}} J_b.$$

Then a satisfies $P_{a,C}$.

Now let c be any assignment satisfying $P_{a,C}$. Suppose for contradiction that c satisfies C. Since c satisfies H_a, the assignment c is non-constant. If $c \notin [a]_{\mathcal{E}}$, then J_c appears in the union defining $P_{a,C}$, and c unsatisfies J_c, which is a contradiction. Hence $c \in [a]_{\mathcal{E}}$, so $(c,a) \in \mathcal{E}$. Since c satisfies C, the one-constraint instance $\{C\}$ is satisfied by c, and therefore also by a, contradicting that a unsatisfies C. Thus every assignment satisfying $P_{a,C}$ unsatisfies C. ◀

We now complete the reduction. The point of Lemma 12 is that whenever a constraint C_i has a non-constant witness a_i, it yields a suffix P_i that is satisfied by a_i and forces every satisfying assignment to falsify C_i. Thus the same reduction as in the general case goes through, after separately handling the exceptional constraints that admit only constant witnesses.

Proof of Theorem 6(b). Let $m := \mathsf{NRD}_n(\Gamma)$, and let $I = \{C_1, \ldots, C_m\}$ be a non-redundant instance of $\mathsf{CSP}(\Gamma)$ on n variables with exactly m constraints. Call a constraint C_i bad if every witness for C_i is a constant assignment (i.e., 0^n or 1^n), and good otherwise. We claim that there are at most two bad constraints. Say three constraints were bad, then two of them would share the same constant witness assignment. This is impossible because a witness for one constraint satisfies every other constraint, and hence cannot falsify two distinct constraints. Let $G \subseteq [m]$ denote the set of good constraints. Then $|G| \geq m - 2$. Using the fact that $m = \mathsf{NRD}_n(\Gamma) \geq \Omega_k(n)$, we get $|G| = \Omega(m) = \Omega(\mathsf{NRD}_n(\Gamma))$.

Fix, for each $i \in G$, a non-constant witness $a_i \in \{0,1\}^n$ such that a_i satisfies every constraint in $I \setminus \{C_i\}$ and falsifies C_i. By Lemma 12, for each $i \in G$ there exists an instance $P_i := P_{a_i, C_i}$ such that a_i satisfies P_i and every assignment satisfying P_i falsifies C_i.

We now give a one-way communication protocol for $\mathsf{INDEX}_{|G|}$. Enumerate $G = \{g_1, \ldots, g_t\}$, where $t := |G|$. Alice is given $x \in \{0,1\}^t$, Bob is given $r \in [t]$, and the goal is to recover x_r. Let $S_x := \{g_j : x_j = 1\} \subseteq G$. Recall that Alice streams the sub-instance $I_{S_x} := \{C_j : j \in S_x\}$ through the streaming algorithm and sends Bob the resulting memory state. Bob sets $i := g_r$ and appends the suffix P_i.

If $x_r = 0$, then $i \notin S_x$, so every constraint of I_{S_x} belongs to $I \setminus \{C_i\}$ and is therefore satisfied by a_i. Since a_i satisfies P_i as well, the instance $I_{S_x} \cup P_i$ is satisfiable.

If $x_r = 1$, then $i \in S_x$, so $C_i \in I_{S_x}$. But every assignment satisfying P_i falsifies C_i, by Lemma 12. Hence $I_{S_x} \cup P_i$ is unsatisfiable.

Thus Bob can recover x_r from the output of the streaming algorithm. Therefore any randomized single-pass streaming algorithm for satisfiability yields a public-coin one-way communication protocol for INDEX_t with the same message length, and therefore by Lemma 8, requires at least $\Omega(t) = \Omega(|G|) = \Omega(\mathsf{NRD}_n(\Gamma))$ space. ◀

4.3 Counterexamples over larger domains

We now show that the lower bound for positive Boolean CSPs does not extend to positive CSPs over larger domains. This is not merely a failure of our proof technique via the equivalence relation $\mathcal{E}$; the lower bound itself is false in general. We give two counterexamples over non-Boolean domains. In each case, single-pass streaming satisfiability is strictly easier than what $\mathsf{NRD}_n(\Gamma)$ would suggest, and the relation $\mathcal{E}$ also fails to be symmetric.

4.3.1 Non-Singleton Constraint Language

Let the domain be $[3] = \{0, 1, 2\}$, and let the language be:

$$\Gamma := \{R, U_1, U_2\}, \quad \text{such that} \quad R := [3]^2 \setminus \{(0,0)\}, \quad U_1 := \{1\}, \quad U_2 := \{2\}$$

Clearly Γ is non-constant-satisfiable (the all-0 assignment violates both U_1 and U_2, the all-1 assignment violates U_2, and the all-2 assignment violates U_1).

Linear Streaming Satisfiability

We first observe that streaming satisfiability for positive $\mathsf{CSP}(\Gamma)$ is easy. Observe that once a variable is assigned a non-zero value, every binary constraint of type R involving that variable is automatically satisfied, since every pair in $\{1, 2\}^2$ belongs to R. Therefore an instance is satisfiable if and only if no variable receives both unary constraints $\{U_1, (x_i)\}$ and $\{U_2, (x_i)\}$. This gives a single-pass $O(n)$-space algorithm: for each variable, store whether it has received a U_1-constraint and whether it has received a U_2-constraint, and reject if both occur. If no conflict occurs, then assigning each variable a value in $\{1, 2\}$ consistent with its unary constraints satisfies every constraint in the instance.

Quadratic NRD

On the other hand, $\mathsf{NRD}_n(\Gamma)$ is quadratic. Consider the instance $I := \{\{R, (x_i, x_j)\} : 1 \leq i < j \leq n\}$. We claim that I is non-redundant. Fix any constraint $\{R, (x_i, x_j)\}$ in I. Set $x_i = x_j = 0$ and set every other variable to 1. This assignment falsifies the chosen constraint, since $(0, 0) \notin R$, but satisfies every other constraint in I, because every other pair has at least one endpoint equal to 1. Thus every constraint in I has a witness assignment, and so I is non-redundant. It follows that $\mathsf{NRD}_n(\Gamma) \geq \binom{n}{2} = \Omega(n^2)$. Since each relation in Γ has arity at most 2, there are only $O(n^2)$ distinct binary constraints and $O(n)$ unary constraints on n variables, so trivially $\mathsf{NRD}_n(\Gamma) = O(n^2)$. Hence $\mathsf{NRD}_n(\Gamma) = \Theta(n^2)$.

No Equivalence Relation

We next observe that the relation $\mathcal{E}$ is not symmetric for this language. Let $a = (0, 1) \in [3]^2$ and $b = (2, 1) \in [3]^2$. Every atomic constraint satisfied by a is also satisfied by b. In particular, $\{U_1, (x_2)\}$ is satisfied by both assignments, and every binary R-constraint satisfied by a remains satisfied after changing the first coordinate from 0 to 2, since the only forbidden tuple of R is $(0, 0)$. It follows that $(a, b) \in \mathcal{E}$. However, b satisfies the unary constraint $\{U_2, (x_1)\}$, while a does not. Therefore $(b, a) \notin \mathcal{E}$, and hence $\mathcal{E}$ is not symmetric.

We conclude that for the positive $\mathsf{CSP}(\Gamma)$ defined by the non-constant-satisfiable language Γ as above, single-pass streaming satisfiability has space complexity $O(n)$ while $\mathsf{NRD}_n(\Gamma) = \Theta(n^2)$. In particular, the lower bound by non-redundancy fails in general over larger domains, and the equivalence-theorem mechanism also breaks down.

The same construction extends immediately to every fixed arity $r \geq 2$. Indeed, if one replaces R by $R_r := [3]^r \setminus \{0^r\}$ and keeps the same unary relations U_1 and U_2, then the same proof gives a positive CSP with single-pass streaming satisfiability complexity $O(n)$ and non-redundancy $\Theta(n^r)$.

4.3.2 Singleton Constraint Language

We now show that the same phenomenon occurs even for a language consisting of a single binary relation. Let the domain be $[4] = \{0, 1, 2, 3\}$, and let $\Gamma := \{T\}$, where $T \subseteq [4]^2$ is defined by $T(x, y)$ iff ($x \in \{2, 3\}$ or $y \in \{2, 3\}$) and $x \not\equiv y \pmod 2$. Equivalently,

$$\Gamma := \{T\}, \quad \text{such that} \quad T = \{(0, 3), (3, 0), (1, 2), (2, 1), (2, 3), (3, 2)\}.$$

Clearly Γ is non-constant-satisfiable, since $(c, c) \notin T$ for every $c \in [4]$.

Linear Streaming Satisfiability

We claim that satisfiability of the positive $\mathsf{CSP}(\Gamma)$ is exactly graph bipartiteness. Given an instance, form its underlying graph on vertex set $[n]$, where each constraint $\{T, (x_i, x_j)\}$ is viewed as an edge between i and j. If the instance is satisfiable, then every edge $\{T, (x_i, x_j)\}$ forces the assigned values on x_i and x_j to have opposite parity, so the parity of the assignment gives a proper 2-coloring of the graph. Conversely, if the graph is bipartite, then assigning the value 2 to one side and the value 3 to the other satisfies every edge, since $(2, 3), (3, 2) \in T$. Therefore satisfiability of $\mathsf{CSP}(\Gamma)$ is equivalent to graph bipartiteness. Since single-pass streaming bipartiteness can be decided in $O(n \log n)$ space, the same upper bound holds for streaming satisfiability of $\mathsf{CSP}(\Gamma)$.

Quadratic NRD

We now show that $\mathsf{NRD}_n(\Gamma) = \Theta(n^2)$. Let $L \cup R = [n]$ be a bipartition with $|L| = \lfloor n/2 \rfloor$ and $|R| = \lceil n/2 \rceil$, and consider the instance $I := \{\{T, (x_i, x_j)\} : i \in L,\ j \in R\}$, corresponding to the complete bipartite graph between L and R. We claim that I is non-redundant. Fix any constraint $\{T, (x_i, x_j)\}$ with $i \in L$ and $j \in R$. Assign the value 0 to x_i, the value 1 to x_j, the value 2 to every other variable in L, and the value 3 to every other variable in R. Then the chosen constraint is falsified, since $(0, 1) \notin T$, while every other constraint in I is satisfied: every remaining edge has endpoints of opposite parity, and at least one endpoint lies in $\{2, 3\}$. Thus every constraint in I has a witness assignment, so I is non-redundant. Hence $\mathsf{NRD}_n(\Gamma) \geq |L||R| = \Omega(n^2)$. Since each relation in Γ has arity at most 2, trivially $\mathsf{NRD}_n(\Gamma) = O(n^2)$, and therefore $\mathsf{NRD}_n(\Gamma) = \Theta(n^2)$.

No Equivalence Relation

Finally, we observe that $\mathcal{E}$ is not symmetric here either. Let $a = (0, 1, 3) \in [4]^3$ and $b = (2, 1, 3) \in [4]^3$. Under a, the only ordered pairs of coordinates that satisfy T are $(1, 3)$ and $(3, 1)$, since $(a_1, a_3) = (0, 3) \in T$ and $(a_3, a_1) = (3, 0) \in T$. Under b, these two ordered pairs are still satisfied, and in addition $(b_1, b_2) = (2, 1) \in T$ and $(b_2, b_1) = (1, 2) \in T$. Thus every atomic constraint satisfied by a is also satisfied by b, so $(a, b) \in \mathcal{E}$. However, b satisfies the constraint $\{T, (x_1, x_2)\}$, while a does not. Therefore $(b, a) \notin \mathcal{E}$, and hence $\mathcal{E}$ is not symmetric.

We conclude that even for the positive $\mathsf{CSP}(\Gamma)$ defined by the above singleton, non-constant-satisfiable binary relation T over a domain of size 4, the lower bound by non-redundancy fails, and the relation $\mathcal{E}$ need not be an equivalence relation.

5 Conclusion

We study the single-pass streaming satisfiability of CSPs through the lens of non-redundancy. Our main result shows that the streaming satisfiability of finite CSPs is tightly characterized by non-redundancy up to a logarithmic factor, with an analogous characterization holding for positive Boolean CSPs that are non-constant-satisfiable. On the algorithmic side, we give a deterministic upper bound by maintaining an equivalent non-redundant sub-instance throughout the stream. On the lower bound side, we introduce a relation on assignments, show that it is an equivalence relation in the settings above, and use this structure to obtain tight reductions from a communication problem.

There are several natural directions for future work. First, it would be interesting to understand *streaming approximate sparsification* through the same lens. In the classical offline setting, recent work of Brakensiek and Guruswami [12] shows that non-redundancy governs approximate sparsification up to polylogarithmic factors for unweighted CSPs, though their result is existential and non-constructive. A natural open question is therefore whether streaming algorithms can maintain small approximate sparsifiers in $\widetilde{O}(\mathsf{NRD}_n(\Gamma))$ space, or whether fundamental barriers emerge in the streaming setting.

Second, our counterexamples over larger domains show that positive CSPs cannot be characterized by non-redundancy in full generality. Even after excluding the trivial cases of empty relations and constant-satisfiable languages, there are positive constraint languages for which single-pass streaming satisfiability is strictly easier than what $\mathsf{NRD}_n(\Gamma)$ predicts. This suggests that the characterization for positive CSPs on non-Boolean domains could be quite different from our characterization, and is an interesting open problem.

Third, our results further highlight the importance of obtaining tight bounds on $\mathsf{NRD}_n(\Gamma)$ itself. Beyond the streaming setting, non-redundancy has emerged as a central structural parameter governing sparsification and kernelization. A better understanding of the possible asymptotic behaviours of $\mathsf{NRD}_n(\Gamma)$, together with more explicit classifications for natural predicate families, would directly sharpen the consequences of our streaming characterization.

References

1 Farid M. Ablayev. Lower bounds for one-way probabilistic communication complexity and their application to space complexity. *Theor. Comput. Sci.*, 157(2):139–159, 1996. doi:10.1016/0304-3975(95)00157-3.

2 Amit Agarwal, Moses Charikar, Konstantin Makarychev, and Yury Makarychev. $O(\sqrt{\log n})$ approximation algorithms for min uncut, min 2cnf deletion, and directed cut problems. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 573–581, 2005. doi:10.1145/1060590.1060675.

3 Kook Jin Ahn and Sudipto Guha. Graph sparsification in the semi-streaming model. In Susanne Albers, Alberto Marchetti-Spaccamela, Yossi Matias, Sotiris E. Nikoletseas, and Wolfgang Thomas, editors, *Automata, Languages and Programming, 36th Internatilonal Colloquium, ICALP 2009, Rhodes, Greece, July 5-12, 2009, Proceedings, Part II*, volume 5556 of *Lecture Notes in Computer Science*, pages 328–338, Berlin, Heidelberg, 2009. Springer. doi:10.1007/978-3-642-02930-1_27.

4 Vedat Levi Alev, Fernando Granha Jeronimo, and Madhur Tulsiani. Approximating constraint satisfaction problems on high-dimensional expanders. In David Zuckerman, editor, *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 180–201. IEEE Computer Society, 2019. doi:10.1109/FOCS.2019.00021.

5 Aditya Anand, Euiwoong Lee, Davide Mazzali, and Amatya Sharma. Min-csps on complete instances II: polylogarithmic approximation for min-nae-3-sat. In Alina Ene and Eshan Chattopadhyay, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2025, Berkeley, CA, USA, August 11-13, 2025*, volume 353 of *LIPIcs*, pages 5:1–5:23. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPIcs.APPROX/RANDOM.2025.5.

6 Aditya Anand, Euiwoong Lee, and Amatya Sharma. Min-csps on complete instances. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1178–1201. SIAM, 2025. `doi:10.1137/1.9781611978322.35`.

7 Per Austrin and Elchanan Mossel. Approximation resistant predicates from pairwise independence. *Computational Complexity*, 18(2):249–271, 2009. `doi:10.1007/s00037-009-0272-6`.

8 Amir Azarmehr, Soheil Behnezhad, and Shane Ferrante. Single-pass streaming csps via two-tier sampling. *CoRR*, abs/2604.01575, 2026. `doi:10.48550/arXiv.2604.01575`.

9 Mitali Bafna, Boaz Barak, Pravesh K Kothari, Tselil Schramm, and David Steurer. Playing unique games on certified small-set expanders. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1629–1642, 2021. `doi:10.1145/3406325.3451099`.

10 Cristina Bazgan, W Fernandez de La Vega, and Marek Karpinski. Polynomial time approximation schemes for dense instances of minimum constraint satisfaction. *Random Structures & Algorithms*, 23(1):73–91, 2003. `doi:10.1002/rsa.10072`.

11 Christian Bessiere, Clément Carbonnel, and George Katsirelos. Chain length and csps learnable with few queries. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, volume 34-02, pages 1420–1427. AAAI Press, 2020. `doi:10.1609/aaai.v34i02.5499`.

12 Joshua Brakensiek and Venkatesan Guruswami. Redundancy is all you need. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, pages 1614–1625, 2025. `doi:10.1145/3717823.3718212`.

13 Joshua Brakensiek, Venkatesan Guruswami, Bart M. P. Jansen, Victor Lagerkvist, and Magnus Wahlström. The richness of CSP non-redundancy. *CoRR*, abs/2507.07942, 2025. `doi:10.48550/arXiv.2507.07942`.

14 Joshua Brakensiek, Venkatesan Guruswami, Bart M. P. Jansen, Victor Lagerkvist, and Magnus Wahlström. Super-linear lower bounds for csp non-redundancy via shrinking instances, 2026. `doi:10.48550/arXiv.2605.19055`.

15 Joshua Brakensiek, Venkatesan Guruswami, and Aaron Putterman. Tight bounds for sparsifying random csps. *CoRR*, abs/2508.13345, 2025. `doi:10.48550/arXiv.2508.13345`.

16 Andrei Bulatov, Peter Jeavons, and Andrei Krokhin. Classifying the complexity of constraints using finite algebras. *SIAM journal on computing*, 34(3):720–742, 2005. `doi:10.1137/S0097539700376676`.

17 Silvia Butti and Stanislav Zivny. Sparsification of binary csps. *SIAM Journal on Discrete Mathematics*, 34(1):825–842, 2020. `doi:10.1137/19M1242446`.

18 Clément Carbonnel. On redundancy in constraint satisfaction problems. In Christine Solnon, editor, *28th International Conference on Principles and Practice of Constraint Programming, CP 2022, Haifa, Israel, July 31 – August 8, 2022*, volume 235 of *LIPIcs*, pages 11:1–11:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. `doi:10.4230/LIPIcs.CP.2022.11`.

19 Siu On Chan, James R. Lee, Prasad Raghavendra, and David Steurer. Approximate Constraint Satisfaction Requires Large LP Relaxations. *Journal of the ACM*, 63(4):1–22, 2016. Conference version in FOCS 2013. `doi:10.1145/2811255`.

20 Hubie Chen. A rendezvous of logic, complexity, and algebra. *SIGACT News*, 37(4):85–114, 2006. `doi:10.1145/1189056.1189076`.

21 Hubie Chen, Bart MP Jansen, and Astrid Pieterse. Best-case and worst-case sparsifiability of boolean csps. *Algorithmica*, 82(8):2200–2242, 2020. `doi:10.1007/s00453-019-00660-y`.

22 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*. Association for Computing Machinery, 2022. `doi:10.1145/3519935.3519983`.

23 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, 2024. Conference version in FOCS 2021. doi:10.1145/3649435.

24 Chi-Ning Chou, Alexander Golovnev, and Santhoshini Velusamy. Optimal Streaming Approximations for all Boolean Max-2CSPs and Max-kSAT. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science*, pages 330–341. IEEE Computer Society, 2020. doi:10.1109/FOCS46700.2020.00039.

25 David A Cohen, Martin C Cooper, Peter G Jeavons, and Andrei A Krokhin. The complexity of soft constraint satisfaction. *Artificial Intelligence*, 170(11):983–1016, 2006. doi:10.1016/j.artint.2006.04.002.

26 Holger Dell and Dieter Van Melkebeek. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. *Journal of the ACM (JACM)*, 61(4):1–27, 2014. doi:10.1145/2629620.

27 Tomás Feder. Network flow and 2-satisfiability. *Algorithmica*, 11:291–319, 1994. doi:10.1007/BF01240738.

28 Tomás Feder and Moshe Y Vardi. The computational structure of monotone monadic snp and constraint satisfaction: A study through datalog and group theory. *SIAM Journal on Computing*, 28(1):57–104, 1998. doi:10.1137/S0097539794266766.

29 Yumou Fei, Dor Minzer, and Shuo Wang. A dichotomy theorem for multi-pass streaming csps. In Aditya Bhaskara and Artur Czumaj, editors, *Proceedings of the 58th Annual ACM Symposium on Theory of Computing, STOC 2026, Salt Lake City, UT, USA, June 22-26, 2026*, pages 246–255. ACM, 2026. To appear. doi:10.1145/3798129.3800744.

30 Yumou Fei, Dor Minzer, and Shuo Wang. Near-optimal space lower bounds for streaming csps. *CoRR*, abs/2604.01400, 2026. doi:10.48550/arXiv.2604.01400.

31 Arnold Filtser and Robert Krauthgamer. Sparsification of two-variable valued constraint satisfaction problems. *SIAM Journal on Discrete Mathematics*, 31(2):1263–1276, 2017. doi:10.1137/15M1046186.

32 Dimitris Fotakis, Michael Lampis, and Vangelis Th. Paschos. Sub-exponential approximation schemes for csps: From dense to almost sparse. In Nicolas Ollinger and Heribert Vollmer, editors, *33rd Symposium on Theoretical Aspects of Computer Science, STACS 2016, Orléans, France, February 17-20, 2016*, volume 47 of *LIPIcs*, pages 37:1–37:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.STACS.2016.37.

33 Mrinalkanti Ghosh and Madhur Tulsiani. From Weak to Strong LP Gaps for All CSPs. In Ryan O'Donnell, editor, *Schloss Dagstuhl – Leibniz-Zentrum für Informatik*, volume 79 of *LIPIcs*, pages 11:1–11:27, 2017. doi:10.4230/LIPIcs.CCC.2017.11.

34 Venkatesan Guruswami and Ali Kemal Sinop. Lasserre hierarchy, higher eigenvalues, and approximation schemes for graph partitioning and quadratic integer programming with psd objectives. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 482–491. IEEE, 2011. doi:10.1109/FOCS.2011.36.

35 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*, volume 81 of *LIPIcs*, pages 8:1–8:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.APPROX-RANDOM.2017.8.

36 Johan Håstad. Some optimal inapproximability results. *Journal of the ACM (JACM)*, 48(4):798–859, 2001. doi:10.1145/502090.502098.

37 Bart MP Jansen and Astrid Pieterse. Optimal sparsification for some binary csps using low-degree polynomials. *ACM Transactions on Computation Theory (TOCT)*, 11(4):1–26, 2019. doi:10.1145/3349618.

38 Peter Jeavons, David Cohen, and Marc Gyssens. Closure properties of constraints. *Journal of the ACM (JACM)*, 44(4):527–548, 1997. doi:10.1145/263867.263489.

39 Fernando Granha Jeronimo, Dylan Quintana, Shashank Srivastava, and Madhur Tulsiani. Unique decoding of explicit ε-balanced codes near the gilbert-varshamov bound. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 434–445. IEEE, IEEE, 2020. `doi:10.1109/FOCS46700.2020.00048`.

40 Fernando Granha Jeronimo, Shashank Srivastava, and Madhur Tulsiani. Near-linear time decoding of ta-shma's codes via splittable regularity. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1527–1536, 2021. `doi:10.1145/3406325.3451126`.

41 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*, pages 1263–1282. Society for Industrial and Applied Mathematics, 2015. `doi:10.1137/1.9781611973730.84`.

42 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*, pages 277–288. Association for Computing Machinery, 2019. `doi:10.1145/3313276.3316364`.

43 Marek Karpinski and Warren Schudy. Linear time approximation schemes for the Gale-Berlekamp game and related minimization problems. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 313–322, 2009. `doi:10.1145/1536414.1536458`.

44 Sanjeev Khanna, Aaron Putterman, and Madhu Sudan. Efficient algorithms and new characterizations for csp sparsification. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, pages 407–416, 2025. `doi:10.1145/3717823.3718205`.

45 Sanjeev Khanna, Aaron Putterman, and Madhu Sudan. A theory of spectral CSP sparsification. *CoRR*, abs/2504.16206, 2025. `doi:10.48550/arXiv.2504.16206`.

46 Sanjeev Khanna, Madhu Sudan, Luca Trevisan, and David P Williamson. The approximability of constraint satisfaction problems. *SIAM Journal on Computing*, 30(6):1863–1920, 2001. `doi:10.1137/S0097539799349948`.

47 Sanjeev Khanna, Madhu Sudan, and David P Williamson. A complete classification of the approximability of maximization problems derived from boolean constraint satisfaction. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 11–20, 1997. `doi:10.1145/258533.258538`.

48 Subhash Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the thirty-fourth annual ACM symposium on Theory of computing*, pages 767–775, 2002. `doi:10.1145/509907.510017`.

49 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):319–357, 2007. `doi:10.1137/S0097539705447372`.

50 Subhash Khot, Madhur Tulsiani, and Pratik Worah. A characterization of strong approximation resistance. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 – June 03, 2014*, pages 634–643. ACM, 2014. `doi:10.1145/2591796.2591817`.

51 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. `doi:10.1145/2688073.2688093`.

52 Gillat Kol, Dmitry Paramonov, Raghuvansh R. Saxena, and Huacheng Yu. Characterizing the multi-pass streaming complexity for solving boolean csps exactly. In *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*, volume 251 of *LIPIcs*, pages 80:1–80:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. `doi:10.4230/LIPIcs.ITCS.2023.80`.

53 Ilan Kremer, Noam Nisan, and Dana Ron. On randomized one-round communication complexity. *Comput. Complex.*, 8(1):21–49, 1999. `doi:10.1007/s000370050018`.

54 Sahil Kuchlous. Streaming algorithms for code sparsification, 2025.

55 Victor Lagerkvist and Magnus Wahlström. Sparsification of sat and csp problems via tractable extensions. *ACM Transactions on Computation Theory (TOCT)*, 12(2):1–29, 2020. doi:10.1145/3389411.

56 Michael Lewin, Dror Livnat, and Uri Zwick. Improved rounding techniques for the max 2-sat and max di-cut problems. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 67–82. Springer, 2002. doi:10.1007/3-540-47867-1_6.

57 Pasin Manurangsi and Dana Moshkovitz. Approximating dense max 2-csps. *CoRR*, abs/1507.08348:396, 2015. doi:10.48550/arXiv.1507.08348.

58 Prasad Raghavendra. Optimal algorithms and inapproximability results for every csp? In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 245–254, 2008. doi:10.1145/1374376.1374414.

59 Thomas J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the 10th ACM Symposium on Theory of Computing (STOC)*, pages 216–226. ACM, 1978. doi:10.1145/800133.804350.

60 Amatya Sharma and Santhoshini Velusamy. Non-redundancy of low-arity symmetric boolean csps, 2026. doi:10.48550/arXiv.2605.14007.

61 Noah G. Singer. Oblivious algorithms for the Max-kAND problem. In Nicole Megow and Adam D. Smith, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, volume 275 of *LIPIcs*, 2023. doi:10.4230/LIPIcs.APPROX/RANDOM.2023.15.

62 Noah G. Singer, Madhur Tulsiani, and Santhoshini Velusamy. Sketching approximations and LP approximations for finite csps are related. *CoRR*, abs/2509.17926, 2025. doi:10.48550/arXiv.2509.17926.

63 Noah G. Singer, Madhur Tulsiani, and Santhoshini Velusamy. Optimal single-pass streaming lower bounds for approximating csps. *CoRR*, abs/2604.08731, 2026. doi:10.48550/arXiv.2604.08731.

64 Hoa T Vu. Revisiting maximum satisfiability and related problems in data streams. *Theoretical Computer Science*, 982:114271, 2024. doi:10.1016/j.tcs.2023.114271.

65 Dmitriy Zhuk. A proof of the csp dichotomy conjecture. *Journal of the ACM (JACM)*, 67(5):1–78, 2020. doi:10.1145/3402029.