

Conditional Dichotomy of Boolean Ordered Promise CSPs

Joshua Brakensiek ✉

Computer Science Department, Stanford University, CA, USA

Venkatesan Guruswami ✉

Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, USA

Sai Sandeep ✉

Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, USA

Abstract

Promise Constraint Satisfaction Problems (PCSPs) are a generalization of Constraint Satisfaction Problems (CSPs) where each predicate has a strong and a weak form and given a CSP instance, the objective is to distinguish if the strong form can be satisfied vs. even the weak form cannot be satisfied. Since their formal introduction by Austrin, Guruswami, and Håstad [1], there has been a flurry of works on PCSPs, including recent breakthroughs in approximate graph coloring [4, 24, 35]. The key tool in studying PCSPs is the algebraic framework developed in the context of CSPs where the closure properties of the satisfying solutions known as *polymorphisms* are analyzed.

The polymorphisms of PCSPs are significantly richer than CSPs – even in the Boolean case, we still do not know if there exists a dichotomy result for PCSPs analogous to Schaefer’s dichotomy result [32] for CSPs. In this paper, we study a special case of Boolean PCSPs, namely Boolean *Ordered* PCSPs where the Boolean PCSPs have the predicate $x \leq y$. In the algebraic framework, this is the special case of Boolean PCSPs when the polymorphisms are *monotone functions*. We prove that Boolean Ordered PCSPs exhibit a computational dichotomy assuming the Rich 2-to-1 Conjecture [9] which is a perfect completeness surrogate of the Unique Games Conjecture.

In particular, assuming the Rich 2-to-1 Conjecture, we prove that a Boolean Ordered PCSP can be solved in polynomial time if for every $\epsilon > 0$, it has polymorphisms where each coordinate has *Shapley value* at most ϵ , else it is NP-hard. The algorithmic part of our dichotomy result is based on a structural lemma showing that Boolean monotone functions with each coordinate having low Shapley value have arbitrarily large threshold functions as minors. The hardness part proceeds by showing that the Shapley value is consistent under a uniformly random 2-to-1 minor. As a structural result of independent interest, we construct an example to show that the Shapley value can be inconsistent under an adversarial 2-to-1 minor.

2012 ACM Subject Classification Theory of computation → Constraint and logic programming; Theory of computation → Approximation algorithms analysis; Theory of computation → Problems, reductions and completeness; Applied computing → Economics

Keywords and phrases promise constraint satisfaction, Boolean ordered PCSP, Shapley value, rich 2-to-1 conjecture, random minor

Digital Object Identifier 10.4230/LIPIcs.ICALP.2021.37

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: <https://arxiv.org/abs/2102.11854> [7]

Funding *Joshua Brakensiek*: Research supported in part by an NSF Graduate Research Fellowship. *Venkatesan Guruswami*: Research supported in part by NSF grant CCF-1908125 and a Simons Investigator award.

Sai Sandeep: Research supported in part by NSF grants CCF-1563742 and CCF-1908125.

Acknowledgements We thank Libor Barto, whose talk [2] and insightful discussions inspired our work.



© Joshua Brakensiek, Venkatesan Guruswami, and Sai Sandeep; licensed under Creative Commons License CC-BY 4.0

48th International Colloquium on Automata, Languages, and Programming (ICALP 2021).

Editors: Nikhil Bansal, Emanuela Merelli, and James Worrell; Article No. 37; pp. 37:1–37:15

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



1 Introduction

Constraint satisfaction problems (CSP) have played a very influential role in the theory of computation, providing an excellent testbed for the development of both algorithmic and hardness techniques, which then extend to more general settings. A CSP over domain D is specified by a finite collection \mathbb{A} of predicates over D , and is denoted as $\text{CSP}(\mathbb{A})$. Given an input containing n variables with constraints on the variables using these predicates, the objective is to identify if we can assign values from D to the variables that satisfies all the constraints. Examples of CSPs include classical problems such as 3-SAT and 3-Coloring of graphs.

When the domain is Boolean, Schaefer [32] proved that every CSP is either in P or is NP-Complete. Feder and Vardi [15] conjectured that the same should hold over arbitrary domains as well. They also showed that the then known algorithmic results all follow by the algebraic closure properties of the CSPs. This notion was formalized by Jeavons, Cohen, and Gyssens [17, 18] and other works [11] that crystallized the (*universal*) *algebraic approach* to CSPs. In the algebraic approach, the higher-order closure properties obeyed by the predicates, namely their *polymorphisms*, are studied. A polymorphism is a function that when applied coordinate-wise to arbitrary satisfying assignments to the predicate, is guaranteed to produce an output that satisfies the predicate. For example, consider an arbitrary instance I of the 2-SAT problem over n variables, and suppose that $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \{0, 1\}^n$ are three assignments that satisfy all the constraints in I . Now, if we compute $\mathbf{u} \in \{0, 1\}^n$ that is obtained by setting $u_i = \text{MAJ}(x_i, y_i, z_i)$ for all $i \in [n]$, the assignment \mathbf{u} also satisfies all the constraints of I . Thus, the majority function on 3 bits is a polymorphism of the 2-SAT CSP. On the other hand, for the 3-SAT problem, it is not hard to prove that the only polymorphisms are the dictator functions. The algebraic approach has been immensely successful and culminated in the recent resolution of Feder-Vardi conjecture by Bulatov [10] and Zhuk [36]. Further, these proofs yield a precise understanding of the mathematical structure underlying efficient algorithms: if the CSP has a “non-trivial” polymorphisms, the CSP is polytime solvable, and otherwise, it is NP-complete.

In this paper, we study Promise Constraint Satisfaction Problems (PCSPs) that vastly generalize the CSPs. In the PCSPs, each predicate has a weak and a strong form—given an instance of PCSP containing n variables with the constraints, the goal is to distinguish between the case that the stronger form can be satisfied vs. even the weaker one cannot be satisfied. A classical example of PCSP is the approximate graph coloring, where given a graph G , the goal is to distinguish between the cases that G can be colored with c colors vs. it cannot be colored with s colors for some $c \leq s$. Another example is the (1-in-3 SAT, NAE-3-SAT), wherein given a 1-in-3-SAT instance that is promised to be satisfiable, the objective is to assign 0, 1 values to the variables such that each constraint is satisfied as a NAE-3-SAT instance, i.e., both 0 and 1 occur in every constraint. While the individual CSPs, namely 1-in-3-SAT and NAE-3-SAT are both NP-hard, the above PCSP is in P. The study of PCSPs was formally initiated by Austrin, Guruswami, and Håstad [1]. and since then, there has been a lot of recent interest in PCSPs, including the development of a systematic theory in [4, 6] and leading to breakthroughs in approximate graph coloring [4, 24, 35].

The central question in the study of PCSPs is whether there exists a complexity dichotomy for PCSPs i.e. if every PCSP is either in P or is NP-complete. As is the case with CSPs, the key tool towards establishing such a dichotomy result is the algebraic approach. The Galois correspondence from the CSP world extends to PCSPs, i.e., the polymorphisms fully capture the computational complexity of the underlying PCSP [6, 30]. This has been extended to show

that just the identities satisfied by the polymorphisms suffice to capture the computational complexity of the underlying PCSP [4]. However, the polymorphisms of PCSPs are much richer, and characterizing which polymorphisms lead to algorithms and which ones lead to hardness has been a challenging problem. Conceptually, the principal difficulty is that the polymorphisms for CSPs are closed under composition (hence referred to as *clones*), whereas for PCSPs, this is no longer the case.

As a result, even in the Boolean case, we do not have a dichotomy theorem for PCSPs. Towards establishing a potential Boolean PCSP dichotomy, progress has been made by Ficak, Kozik, Olsák and Stankiewicz [16], who obtained a dichotomy result when each predicate is symmetric. In this paper, we study Boolean PCSPs that contain the *simplest non-symmetric predicate*, $x \rightarrow y$. We call such Boolean PCSPs *Ordered* as we can also view the implication constraint as an ordering requirement $x \leq y$.

Ordered Boolean PCSPs have come under recent study. The work of Petr [29] (inspired by work of Barto [2, 3]) considered a special class of Ordered Boolean PCSPs which have an additional predicate $x \neq y$ (this corresponds to allowing negations in the constraints) as well as the requirement that the majority on three bits is *not* a polymorphism. In this setting Petr was able to show that such Ordered Boolean PCSPs are NP-hard. However, the approach considered does not seem immediately extendable to analyzing general Ordered Boolean PCSPs [3].

The main motivation for studying these PCSPs comes from the fact that adding the additional $x \leq y$ predicate is equivalent to restricting the polymorphisms of the PCSPs to be *monotone functions*. Monotonicity is an influential theme in the study of Boolean functions and complexity theory, and understanding the structure of polymorphisms in the monotone case is an important (and certainly necessary) subcase towards a general characterization of polymorphisms vs. tractability for arbitrary Boolean PCSPs. For the special case of Boolean Ordered PCSPs which include negation constraints, it was conjectured in [3] that polynomial time tractability is characterized by the existence of majority polymorphisms of arbitrarily large arity.

Our main result is that Boolean Ordered PCSPs exhibit a dichotomy, under the recently introduced *Rich 2-to-1 Conjecture* of Braverman, Khot, and Minzer [9].

► **Theorem 1.** *Assuming the Rich 2-to-1 Conjecture, every Ordered Boolean PCSP is either in P or is NP-Complete. Furthermore, an Ordered PCSP Γ is in P if and only if for every $\epsilon > 0$, there are polymorphisms of Γ with every coordinate having Shapley value at most ϵ . Equivalently, Γ is in P if and only if it has threshold polymorphisms of arbitrarily large arity.*

As a concrete example, recall the earlier mentioned example of (1-in-3-SAT, NAE-3-SAT). As it has threshold polymorphisms of arbitrarily large arity, it remains polynomial time solvable even after adding the predicate $x \rightarrow y$. However, if we also add another two-variable predicate $x \neq y$, the PCSP no longer has threshold polymorphisms, and by our above result, it becomes NP-Complete.

We obtain the conditional dichotomy result by analyzing the polymorphisms of the Ordered PCSPs. The key idea in the algebraic approach to PCSPs is that the PCSP is tractable if the polymorphisms are close to symmetric, and the PCSP is hard if all the polymorphisms have a small number of “important” coordinates. More concretely, on the algorithmic front, it has been proved that symmetric polymorphisms of arbitrarily large arities lead to polynomial time algorithms for PCSPs [8]. On the hardness side, if all the polymorphisms depend on a bounded number of coordinates, then the underlying PCSP is NP-hard [1]. This has been extended to various other notions, including combinatorial ones

such as C -fixing [5], and topological ones such as having a bounded number of coordinates with non-zero winding number [24]. In this paper, we study the monotone polymorphisms using analytical techniques.

In particular, we use Shapley value to analyze the monotone polymorphisms. For a monotone function $f : \{0, 1\}^n \rightarrow \{0, 1\}$, Shapley value of a coordinate i is the probability that on a random path from $\{0, 0, \dots, 0\}$ to $\{1, 1, \dots, 1\}$, the function value turns from 0 to 1 when we switch the i th coordinate to 1. Initially studied to understand the power of an individual in voting systems [33], Shapley value has now found applications in various settings, especially in game theory [26, 27]. In our setting, there are two advantages of using Shapley value to study the polymorphisms. First, it is a relative measure of the importance of a coordinate, as opposed to other notions of Influence which are absolute. This helps in bounding the number of coordinates with Shapley value above a certain threshold. Second, it is a versatile measure with combinatorial and analytical interpretations [12] which helps in proving that Shapley value stays consistent under function minors¹, a key property necessary in both the algorithm and the hardness.

Algorithm Overview. We obtain our algorithmic result by using the Basic Linear Programming with Affine relaxation (BLP+Affine relaxation), combined with a structural result regarding the monotone functions with bounded Shapley value. As mentioned earlier, PCSPs with symmetric polymorphisms of arbitrarily large arities can be solved in polynomial time using the BLP+Affine relaxation algorithm [8]. Our main structural result is that Boolean functions with bounded Shapley value have arbitrarily large threshold functions as minors. Since the set of polymorphisms of a PCSP are closed under taking minors, this proves that the underlying PCSP Γ has arbitrarily large threshold functions as polymorphisms, which then implies that Γ is in P. The key tool underlying our structural result is a result of Kalai [19] that states that under certain conditions, monotone Boolean functions with arbitrarily small Shapley value have a sharp threshold.

Hardness Overview. We obtain our hardness result assuming the Rich 2-to-1 Conjecture. Braverman, Khot, and Minzer [9] introduced the conjecture as a perfect completeness surrogate of the well known Unique Games Conjecture [21]. They also proved that the conjecture is equivalent to Unique Games Conjecture when we relax the perfect completeness requirement. The reduction from the Rich 2-to-1 Conjecture to PCSPs follows using the standard Label Cover-Long Code paradigm. The key ingredient in this reduction is a decoding of the Long Codes to a bounded number of coordinates that is consistent under function minors. We decode each Long Code function to the coordinates with $\Omega(1)$ Shapley value – as the sum of Shapley values of all the coordinates of any monotone function is equal to 1, there is a bounded number of such coordinates. We argue about the consistency of this decoding using a structural result that states that under a uniformly random minor, Shapley value is roughly preserved.

On the necessity of “richness” in 2-to-1 Conjecture. A natural question is whether our hardness result can be obtained using a weaker assumption such as the 2-to-1 conjecture (whose imperfect completeness version was recently established [13, 14, 22, 23]). We shed some light on this question by showing that there are monotone Boolean functions $f : \{0, 1\}^{2n} \rightarrow \{0, 1\}$ and $g : \{0, 1\}^n \rightarrow \{0, 1\}$ such that g is a minor of f with respect to the

¹ A minor (formally defined in Section 2) of a function $f : \{0, 1\}^m \rightarrow \{0, 1\}$ is a function $g : \{0, 1\}^n \rightarrow \{0, 1\}$ of smaller arity $n \leq m$ obtained from f by identifying sets of variables together.

2-to-1 function π , both the functions f and g have exactly one coordinate i_1, i_2 respectively, with $\Omega(1)$ Shapley value, and yet $\pi(i_1) \neq i_2$. Such an adversarial example is interesting from two angles: first, it shows that even using the 2-to-1 conjecture, the Shapley value based decoding is not consistent. Second, it gives an example of agents pairing up maliciously to completely alter the Shapley value. The underlying phenomenon is that the rich 2-to-1 games have “subcode-covering” property, which is absent in the standard 2-to-1 games, helping in preserving the consistency of any biased influence measure such as the Shapley value.

Organization. In Section 2, we formally define PCSPs, polymorphisms, and Shapley value. We present the algorithmic and hardness parts of our dichotomy result in Section 3 and Section 4 respectively. We present the adversarial example of a 2-to-1 minor that alters the Shapley value in Section 5.

2 Preliminaries

Notations. We use $[n]$ to denote the set $\{1, 2, \dots, n\}$. For a k -ary relation $A \subseteq [q]^k$, we abuse the notation and use A both as a subset of $[q]^k$, and also as a predicate $A : [q]^k \rightarrow \{0, 1\}$. For a vector $\mathbf{x} = (x_1, x_2, \dots, x_n) \in \{0, 1\}^n$, we use $\text{hw}(\mathbf{x})$ to denote $\sum_{i=1}^n x_i$. For two vectors $\mathbf{x}, \mathbf{y} \in \{0, 1\}^n$, we say that $\mathbf{x} \leq \mathbf{y}$ if $x_i \leq y_i$ for all $i \in [n]$. A Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ is called monotone if $f(\mathbf{x}) \leq f(\mathbf{y})$ for all $\mathbf{x} \leq \mathbf{y}$.

PCSPs and Polymorphisms. We first define Constraint Satisfaction Problems (CSP).

► **Definition 2 (CSP).** *Given a k -ary relation $A : D^k \rightarrow \{0, 1\}$ over a domain D , the Constraint Satisfaction Problem (CSP) associated with the predicate A takes a set of variables $V = \{v_1, v_2, \dots, v_n\}$ as input which are to be assigned values from D . There are m constraints (e_1, e_2, \dots, e_m) each consisting of $e_i = ((e_i)_1, (e_i)_2, \dots, (e_i)_k) \subseteq V^k$ that indicate that the corresponding assignment should belong to A . The objective is to identify if there is an assignment $V \rightarrow D$ that satisfies all the constraints.*

In general, we can have multiple relations A_1, A_2, \dots, A_l , and different constraints can use different relations. We denote such a CSP by $\text{CSP}(A_1, A_2, \dots, A_l)$.

We formally define Promise Constraint Satisfaction Problems (PCSP).

► **Definition 3 (PCSP).** *In a Promise Constraint Satisfaction Problem $\text{PCSP}(\Gamma)$ over a pair of domains D_1, D_2 , we have a set of pairs of relations $\Gamma = \{(A_1, B_1), (A_2, B_2), \dots, (A_l, B_l)\}$ such that for every $i \in [l]$, A_i is a subset of $D_1^{k_i}$ and B_i is a subset of $D_2^{k_i}$. Furthermore, there is a homomorphism $h : D_1 \rightarrow D_2$ such that for all $i \in [l]$ and $x \in D_1^{k_i}$, $x \in A_i$ implies $h(x) \in B_i$. Given a $\text{CSP}(A_1, A_2, \dots, A_l)$ instance, the objective is to distinguish between the two cases:*

1. *There is an assignment to the variables from D_1 that satisfies every constraint when viewed as $\text{CSP}(A_1, A_2, \dots, A_l)$.*
2. *There is no assignment to the variables from D_2 that satisfies every constraint when viewed as $\text{CSP}(B_1, B_2, \dots, B_l)$.*

We now define Boolean Ordered PCSPs.

► **Definition 4 (Boolean Ordered PCSP).** *A PCSP $\text{PCSP}(\Gamma)$ over a pair of domains D_1, D_2 with the set of pairs of relations $\Gamma = \{(A_1, B_1), (A_2, B_2), \dots, (A_l, B_l)\}$ is said to be Boolean Ordered if the following hold.*

1. *The domains are both Boolean i.e., $D_1 = D_2 = \{0, 1\}$.*
2. *There exists $i \in [l]$ such that $A_i = B_i = \{(0, 0), (0, 1), (1, 1)\}$.*

37:6 Conditional Dichotomy of Boolean Ordered PCSPs

Associated with every PCSP, there are polymorphisms that capture the closure properties of the satisfying solutions to the PCSP. More formally, we can define polymorphisms of a PCSP as follows.

► **Definition 5 (Polymorphisms).** For $PCSP(\Gamma)$ with $\Gamma = \{(A_1, B_1), (A_2, B_2), \dots, (A_l, B_l)\}$ where for every $i \in [l]$, $A_i : [q_1]^{k_i} \rightarrow \{0, 1\}$, $B_i : [q_2]^{k_i} \rightarrow \{0, 1\}$, a polymorphism of arity n is a function $f : [q_1]^n \rightarrow [q_2]$ that satisfies the below property for all $i \in [l]$. For all $(\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{k_i})$ such that for all $j \in [n]$, $((\mathbf{v}_1)_j, (\mathbf{v}_2)_j, \dots, (\mathbf{v}_{k_i})_j) \in A_i$, we have

$$(f(\mathbf{v}_1), f(\mathbf{v}_2), \dots, f(\mathbf{v}_{k_i})) \in B_i$$

We use $\text{Pol}(\Gamma)$ to denote the family of all the polymorphisms of $PCSP(\Gamma)$.

A crucial property satisfied by $\text{Pol}(\Gamma)$ is that the family of functions is closed under taking minors. We first define the minor of a function formally.

► **Definition 6 (Minor of a function).** For a Boolean function $f : [q]^n \rightarrow [q']$, the function $g : [q]^m \rightarrow [q']$ is said to be a minor of f with respect to the function $\pi : [n] \rightarrow [m]$ if

$$g(x_1, x_2, \dots, x_m) = f(x_{\pi(1)}, x_{\pi(2)}, \dots, x_{\pi(n)}) \forall x_1, x_2, \dots, x_m \in [q]$$

We say that a function g is a minor of f if there exists some π such that g is a minor of f with respect to π .

We are often interested in 2-to-1 minors. A function g is said to be a 2-to-1 minor of f if there exists a 2-to-1 function π such that g is a minor of f with respect to π , where 2-to-1 function is defined below.

► **Definition 7 (2-to-1 function).** A function $\pi : [2n] \rightarrow [n]$ is said to be a 2-to-1 function if

$$|\pi^{-1}(i)| = 2 \forall i \in [n]$$

We use $\mathcal{F}_{2 \rightarrow 1}(n)$ to denote the set of all the 2-to-1 functions from $[2n]$ to $[n]$.

By the definition of the polymorphisms, we can infer that if $f \in \text{Pol}(\Gamma)$ for a PCSP Γ , then for all functions g such that g is a minor of f , we have $g \in \text{Pol}(\Gamma)$. Such a family of functions that is closed under taking minors is called as a *minion*. We often refer to the family of polymorphisms of a PCSP as the polymorphism minion.

We refer the reader to [4] for an extensive introduction to PCSPs and polymorphisms.

Shapley value. Let $f : \{0, 1\}^n \rightarrow \{0, 1\}$ be a monotone Boolean function. We can view the monotone Boolean function f as a voting scheme between two parties, and n agents: the winner of the voting scheme when the i th agent votes for $\mathbf{x}_i \in \{0, 1\}$ is $f(\mathbf{x})$. The relative power of an agent in a voting scheme is typically measured using the Shapley-Shubix Index, also known as Shapley Value.

Informally speaking, the Shapley Value of a coordinate i is the probability that the i th agent is the altering vote when we start with all zeroes and flip the votes in a uniformly random order. More formally,

► **Definition 8 (Shapley value).** Let $f : \{0, 1\}^n \rightarrow \{0, 1\}$ be a monotone Boolean function. Let $\sigma \in S_n$ be a uniformly random permutation of $[n]$. For an integer $j \in [n]$, let P_j denote the set of first j elements of σ i.e., $P_j := \{\sigma(1), \sigma(2), \dots, \sigma(j)\}$. The Shapley value $\Phi_f(i)$ of the coordinate $i \in [n]$ is defined as

$$\Phi_f(i) := Pr_{\sigma} \{\exists j \in [n] : \sigma(j) = i, f(P_{j-1}) = 0, f(P_j) = 1\}$$

We also give an alternate definition of Shapley value using the notion of boundary of a coordinate. For a monotone Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ and coordinate $i \in [n]$, let $\mathcal{B}_f(i)$ denote the boundary of the coordinate i i.e.

$$\mathcal{B}_f(i) := \{S \subseteq [n] \setminus \{i\} : f(\{i\} \cup S) = 1, f(S) = 0\}$$

By the monotonicity of f , we can infer that $\mathcal{B}_f(i)$ satisfies the following sandwich property that will be useful later.

► **Proposition 9.** *Let $f : \{0, 1\}^n \rightarrow \{0, 1\}$ be a monotone Boolean function and let $i \in [n]$. Then, for every pair of sets $S_1, S_2 \in \mathcal{B}_f(i)$ with $S_1 \subseteq S_2$, we have $S \in \mathcal{B}_f(i)$ for all S such that $S_1 \subseteq S \subseteq S_2$.*

Proof. By the monotonicity of f , we have $f(S \cup \{i\}) \geq f(S_1 \cup \{i\}) = 1$, and thus, $f(S \cup \{i\}) = 1$. Similarly, we have $f(S) \leq f(S_2) = 0$, and thus, $f(S) = 0$. ◀

For an index $j \in \{0, 1, \dots, n-1\}$, let $\mu_f(j)^{(i)}$ denote the fraction of subsets of $[n]$ of size j that are in $\mathcal{B}_f(i)$ i.e.

$$\mu_f(j)^{(i)} := \left| \mathcal{B}_f(i) \cap \binom{[n]}{j} \right| / \binom{[n]}{j}.$$

We can rewrite the definition of Shapley value of the i th coordinate as the following [34]:

$$\Phi_f(i) = \frac{\sum_{j=0}^{n-1} \mu_f(j)^{(i)}}{n}. \tag{1}$$

3 Algorithm when Shapley values are small

In this section, we show that monotone Boolean functions where each coordinate has bounded Shapley value has arbitrarily large threshold functions as minors, thereby proving the algorithmic part of our dichotomy result.

Let L be a positive integer and $0 \leq \tau \leq L$ be a non-negative integer. We let $\text{THR}_{L,\tau} : \{0, 1\}^L \rightarrow \{0, 1\}$ be the threshold function on L variables with threshold τ . More formally,

$$\text{THR}_{L,\tau}(\mathbf{x}) := \begin{cases} 1 & \text{if } \text{hw}(\mathbf{x}) \geq \tau \\ 0 & \text{otherwise.} \end{cases}$$

For a monotone Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ and real number $p \in [0, 1]$, let $P_p(f)$ denote the expected value of $f(x)$ where each element $x_i, i \in [n]$ is independently set to be 1 with probability p and 0 with probability $1 - p$. For every monotone function f , the function $P_p(f)$ is a strictly monotone continuous function in p on the interval $[0, 1]$. The value $p_c = p_c(f)$ at which $P_{p_c}(f) = \frac{1}{2}$ is called the *critical probability* of f .

Using the Russo-Margulis Lemma [25, 31] and Poincaré Inequality, we can show the following lemma that we need later.

► **Lemma 10** (Exercise 8.29(e) in [28]). *Let f be a non-constant monotone Boolean function with critical probability $p_c \leq \frac{1}{2}$. Let $p_1 := \frac{1}{(2\nu)^2} p_c$ for $\nu > 0$. If $p_1 \leq \frac{1}{2}$, then $P_{p_1}(f) \geq 1 - \nu$.*

We now define the threshold interval of f .

► **Definition 11.** *For a monotone function f and $0 < \epsilon < \frac{1}{2}$, we define $T_\epsilon(f) := p_2 - p_1$, where p_2 and p_1 are such that $P_{p_1}(f) = \epsilon, P_{p_2}(f) = 1 - \epsilon$.*

Kalai [19] proved the following result regarding monotone Boolean functions.

► **Theorem 12.** *For every $a, \epsilon, \gamma > 0$, there exists $\delta := \delta(a, \epsilon, \gamma) > 0$ such that for every monotone Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ with $\Phi_f(i) \leq \delta$ for all $i \in [n]$ and $a \leq p_c(f) \leq 1 - a$, then $T_\epsilon(f) \leq \gamma$.*

We will use this result to show that for every monotone function where each coordinate has bounded Shapley value has arbitrarily large threshold functions as minor.

► **Lemma 13.** *For every $L \geq 2$, there exists a $\delta := \delta(L) > 0$ such that the following holds. For any monotone Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ with*

$$\Phi_f(i) \leq \delta \quad \forall i \in [n]$$

there exists a positive integer $L' \in \{L, L + 1\}$ and a non-negative integer τ such that $\text{THR}_{L', \tau}$ is a minor of f .

Proof. We obtain $\delta := \delta(L) > 0$ from Theorem 12 by setting $\epsilon = \frac{1}{2L+1}, \gamma = a = \frac{1}{L^3}$. Our goal is to show that for this parameter δ , for every monotone Boolean function f with each coordinate having Shapley value at most δ , there exists $L' \in \{L, L + 1\}$ and τ such that $\text{THR}_{L', \tau}$ is a minor of f .

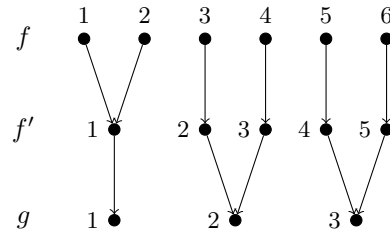
We assume that f is a non-constant function, else we have a trivial minor by setting $\tau = 0$ or $\tau = L'$. Let p_c be the critical probability of f .

Case 1: $p_c < a = \frac{1}{L^3}$. Let $p_1 = L^2 p_c < \frac{1}{L}$. Using Lemma 10, we can conclude that $P_{p_1}(f) \geq 1 - \frac{1}{2L}$. As $P_p(f)$ is monotone, we get that $P_{\frac{1}{L}}(f) > 1 - \frac{1}{2L}$. We let $g : \{0, 1\}^L \rightarrow \{0, 1\}$ be a uniformly random minor of f i.e. we choose the function $\pi : [n] \rightarrow [L]$ by choosing each value $\pi(i)$ uniformly and independently at random from $[L]$, and we let g to be the minor of f with respect to π .

Note that for every $i \in [L]$, the distribution of $g(\{i\})$ over the random minor g is the same as sampling a random input to f where we set each bit to 1 with probability $\frac{1}{L}$. As $P_{\frac{1}{L}}(f) \geq 1 - \frac{1}{2L}$, we get that for each $i \in [L]$, $g(\{i\}) = 1$ with probability at least $1 - \frac{1}{2L}$. By union bound, with probability at least $\frac{1}{2}$, $g(\{i\}) = 1$ for all $i \in [L]$. As $f(0, 0, \dots, 0) = 0$, $g(\phi) = 0$ as well. Thus, with probability at least $\frac{1}{2}$, $g = \text{THR}_{L, 1}$. Hence, $\text{THR}_{L, 1}$ is a minor of f .

Case 2: $p_c > 1 - a = 1 - \frac{1}{L^3}$. Let f^\dagger be the Boolean dual of f defined as $f^\dagger(x) = 1 - f(\bar{x})$. Note that $P_p(f^\dagger) = 1 - P_{1-p}(f)$ for all $p \in [0, 1]$. Thus, $p_c(f^\dagger) = 1 - p_c < a$. Using the previous case, we can infer that $\text{THR}_{L, 1}$ is a minor of f^\dagger with respect to a function $\pi : [n] \rightarrow [L]$. The same function π proves that $\text{THR}_{L, 1}^\dagger = \text{THR}_{L, L}$ is a minor of f .

Case 3: $a \leq p_c \leq 1 - a$. Using Theorem 12, we obtain p_1 such that $P_{p_1}(f) \leq \epsilon$, and $P_{p_1 + \gamma} \geq 1 - \epsilon$, where $\epsilon = \frac{1}{2L+1}, \gamma = \frac{1}{L^3}$. As $\gamma < \frac{1}{L(L+1)}$, there exists $L' \in \{L, L + 1\}$ and $\tau \in [L']$ such that $p_1 + \gamma < \frac{\tau}{L'}$ and $p_1 > \frac{\tau-1}{L'}$. Thus, we get that $P_{\frac{\tau}{L'}}(f) > 1 - \epsilon$ and $P_{\frac{\tau-1}{L'}} < \epsilon$. Let $g : \{0, 1\}^{L'} \rightarrow \{0, 1\}$ be a uniformly random minor of f i.e. we choose $\pi : [n] \rightarrow [L']$ by setting each value uniformly and independently at random from $[L']$ and set g to be the minor of f with respect to π . For a vector $\mathbf{x} \in \{0, 1\}^{L'}$ with $\text{hw}(\mathbf{x}) = \tau$, with probability greater than $1 - \frac{1}{2L+1}$, $g(\mathbf{x}) = 1$. Similarly, for $\mathbf{x} \in \{0, 1\}^{L'}$ with $\text{hw}(\mathbf{x}) = \tau - 1$, with probability greater than $1 - \frac{1}{2L+1}$, $g(\mathbf{x}) = 0$. Thus, with non-zero probability, $g(\mathbf{x}) = 1$ for all $x \in \{0, 1\}^{L'}$ with $\text{hw}(\mathbf{x}) = \tau$ and $g(\mathbf{x}) = 0$ for all $\mathbf{x} \in \{0, 1\}^{L'}$ with $\text{hw}(\mathbf{x}) = \tau - 1$. In other words, with non-zero probability, g is equal to $\text{THR}_{L', \tau}$. Thus, $\text{THR}_{L', \tau}$ is a minor of f . ◀



■ **Figure 1** An illustration of the two step minor approach: Here $f : \{0, 1\}^6 \rightarrow \{0, 1\}$ is a Boolean function, $f' : \{0, 1\}^5 \rightarrow \{0, 1\}$ is a minor of f with respect to the function $\pi_1 : [6] \rightarrow [5]$ with $\pi_1(i) = \max(i - 1, 1)$, and g is a minor of f' with respect to the function $\pi_2 : [5] \rightarrow [3]$ with $\pi_2(i) = \lceil \frac{i+1}{2} \rceil$.

Using the existence of arbitrarily large arity threshold minors, the algorithmic part of our Dichotomy result follows immediately.

► **Theorem 14.** *Let Γ be a Promise CSP template. Suppose that for every $\epsilon > 0$, there exists a function $f \in \text{Pol}(\Gamma)$, $f : \{0, 1\}^n \rightarrow \{0, 1\}$ such that $\Phi_i(f) \leq \epsilon$ for all $i \in [n]$. Then, $\text{PCSP}(\Gamma) \in \text{P}$.*

Proof. Using Lemma 13, we can conclude that there are infinitely many positive integers L such that there exists $\tau \in \{0, 1, \dots, L\}$ with $\text{THR}_{L,\tau} \in \text{Pol}(\Gamma)$. As the threshold functions are symmetric², $\text{Pol}(\Gamma)$ has symmetric polymorphisms of infinitely many arities. Thus, using the BLP+Affine algorithm of [8], $\text{PCSP}(\Gamma)$ can be solved in polynomial time. ◀

We remark that the above result is inspired by a special case shown by Barto [2] that a Boolean Ordered PCSP is polytime tractable if it has cyclic polymorphisms of arbitrarily large arities.

4 Hardness Assuming Rich 2-to-1 Conjecture

In this section, we prove the hardness part of our dichotomy result. First, we prove that Shapley value is preserved under uniformly random 2-to-1 minors, and then we use this to show the hardness assuming the Rich 2-to-1 Conjecture.

4.1 Shapley value under random 2-to-1 minor

Let $f : \{0, 1\}^{2n} \rightarrow \{0, 1\}$ be a monotone Boolean function with $\Phi_f(1) \geq \lambda$ for some absolute constant $\lambda > 0$. Let $g : \{0, 1\}^n \rightarrow \{0, 1\}$ be a minor of f with respect to the uniformly random 2-to-1 function $\pi : [2n] \rightarrow [n]$. Our goal in this subsection is to show that $\mathbb{E}_\pi[\Phi_g(\pi(1))] \geq \gamma$ for some function $\gamma := \gamma(\lambda) > 0$. We prove this in two steps. (See Figure 1)

1. First, we consider the minor of f , $f' : \{0, 1\}^{2n-1} \rightarrow \{0, 1\}$ obtained with respect to $\pi_1 : [2n] \rightarrow [2n-1]$ where $\pi_1(1) = \pi_1(2) = 1, \pi_1(i) = i - 1 \forall i \in \{3, 4, \dots, 2n\}$. We show that $\Phi_{f'}(1) \geq \frac{\lambda}{2}$.
2. Next, we consider a minor g of f' obtained with respect to the function $\pi_2 : [2n-1] \rightarrow [n]$ which has $\pi_2(1) = 1$ while the rest $2n - 2$ values are chosen using a uniformly random partition of $[2n - 2]$ into $n - 1$ pairs. We show that $\mathbb{E}_{\pi_2}[\Phi_g(1)] \geq \gamma$ for some function $\gamma := \gamma(\lambda) > 0$.

² A function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ is said to be symmetric if it is unchanged by any permutation of the input variables.

37:10 Conditional Dichotomy of Boolean Ordered PCSPs

Note that the process of first taking the f' minor and then obtaining g by partitioning $[2n-2]$ into $n-1$ uniformly random pairs is equivalent to taking a uniformly random 2-to-1 minor of f . Thus, the two steps together prove the required Shapley value property of the uniformly random 2-to-1 minor.

The first step is captured by the following pair of lemmas, which we prove in the full version.

► **Lemma 15.** *Let $f : \{0, 1\}^{2n} \rightarrow \{0, 1\}$ and $f' : \{0, 1\}^{2n-1} \rightarrow \{0, 1\}$ be monotone Boolean functions such that f' is a minor of f with respect to the function $\pi_1 : [2n] \rightarrow [2n-1]$ defined as $\pi_1(i) = \max(i-1, 1)$. If $\Phi_f(1) \geq \lambda$, then $\Phi_{f'}(1) \geq \frac{\lambda}{2}$.*

► **Lemma 16.** *Let $f' : \{0, 1\}^{2n-1} \rightarrow \{0, 1\}$ be a monotone Boolean function such that $\Phi_{f'}(1) = \lambda$ with $\lambda \geq \frac{1}{n}$. For an integer $j \in \{0, 1, \dots, 2n-2\}$, let $\mu'(j) = \mu_{f'}(j)^{(1)}$. Then, there exists an absolute constant $\gamma := \gamma(\lambda) > 0$ such that*

$$\frac{\sum_{j=0}^{n-1} \mu'(2j)}{n} \geq \gamma$$

We now prove the second step in the proof.

► **Lemma 17.** *Suppose that $f' : \{0, 1\}^{2n-1} \rightarrow \{0, 1\}$ is a monotone Boolean function such that $\Phi_{f'}(1) \geq \lambda$ with $\lambda \geq \frac{1}{n}$. Let g be a random minor of f' with respect to $\pi_2 : [2n-1] \rightarrow [n]$ which is obtained by setting $\pi_2(1) = 1$, and for every $i > 1$, we randomly choose $j_1, j_2 \in [2n-1] \setminus \{1\}$ (without replacements) and set $\pi_2(j_1) = \pi_2(j_2) = i$. In other words, we choose a uniformly random partition of $[2n-1] \setminus \{1\}$ into $n-1$ pairs P_2, P_3, \dots, P_n and set $\pi_2(j) = i \forall j \in P_i$. Then, there exists $\gamma := \gamma(\lambda) > 0$ such that*

$$\mathbb{E}_{\pi_2}[\Phi_g(1)] \geq \gamma.$$

Proof. For ease of notation, we let $\mu'(j) = \mu_{f'}(j)^{(1)}$ and $\mu_g(j) = \mu_g(j)^{(1)}$. For a set $S \subseteq [n] \setminus \{1\}$ and a function $\pi_2 : [2n-1] \rightarrow [n]$ with $\pi_2(1) = 1$, and $|\pi_2^{-1}(i)| = 2$ for all $i \in \{2, 3, \dots, n\}$, let $\pi_2^{-1}(S)$ be the $2|S|$ sized subset of $\{2, 3, \dots, 2n-1\}$ defined as follows:

$$\pi_2^{-1}(S) := \{\pi_2^{-1}(i) : i \in S\}$$

For every set $S \subseteq \{2, 3, \dots, n\}$, when $\pi_2 : [2n-1] \rightarrow [n]$ is a uniformly random 2-to-1 minor with $\pi_2(1) = 1$, and the rest $2n-2$ elements are partitioned into $n-1$ pairs uniformly at random, the set $\pi_2^{-1}(S)$ is distributed uniformly in $\binom{[2n-1] \setminus \{1\}}{2|S|}$. Also note that $S \in \mathcal{B}_g(1)$ if and only if $\pi_2^{-1}(S) \in \mathcal{B}_{f'}(1)$. Thus, for every set $S \subseteq \{2, 3, \dots, n\}$, the probability that $S \in \mathcal{B}_g(1)$ (over the choice of π_2) is equal to $\mu'(2|S|)$. Summing over all such sets of size j , we get that for every $j \in \{0, 1, \dots, n-1\}$, the expected value of $\mu_g(j)$ is equal to $\mu'(2j)$.

$$\mathbb{E}_{\pi_2}[\mu_g(j)] = \mu'(2j) \forall j \in \{0, 1, \dots, n-1\}$$

By using Lemma 16, we can infer that there exists $\gamma = \gamma(\lambda) > 0$ such that $\sum_{j=0}^{n-1} \mathbb{E}_{\pi_2}[\mu_g(j)] = \sum_{j=0}^{n-1} \mu'(2j) \geq \gamma n$. Using Equation (1), we get

$$\mathbb{E}_{\pi_2}[\Phi_g(1)] = \mathbb{E}_{\pi_2} \left[\frac{\sum_{j=0}^{n-1} \mu_g(j)}{n} \right] = \frac{\sum_{j=0}^{n-1} \mathbb{E}_{\pi_2}[\mu_g(j)]}{n} \geq \gamma. \quad \blacktriangleleft$$

Lemma 15 and Lemma 17 together prove that Shapley value behaves well under uniformly random 2-to-1 minors for monotone Boolean functions.

► **Lemma 18.** *Suppose that $f : \{0, 1\}^{2n} \rightarrow \{0, 1\}$ is a monotone Boolean function such that $\Phi_f(1) \geq \lambda$ for some absolute constant $\lambda > 0$ with $\lambda \geq \frac{1}{n}$. Then, there exists $\gamma := \gamma(\lambda) > 0$ such that*

$$\mathbb{E}_\pi[\Phi_g(\pi(1))] \geq \gamma$$

where g is a minor of f with respect to the uniformly random 2-to-1 function π .

Proof. Combining Lemma 15 and Lemma 17, we can conclude that for every $i \in [2n], i > 1$, when $\pi : [2n] \rightarrow [n]$ is a uniformly random 2-to-1 minor conditioned on the fact that $\pi(1) = \pi(i)$, we have $\mathbb{E}_\pi[\Phi_g(\pi(1))] \geq \gamma$. Taking average over all the $i \in [2n], i > 1$, we get a proof that the same inequality holds when π is a uniformly random 2-to-1 minor. ◀

4.2 Reduction

We first formally define the Label Cover problem and state the Rich 2-to-1 Conjecture.

► **Definition 19** (Label Cover). *In the Label Cover problem $\mathcal{G} = (G, \Sigma_L, \Sigma_R, \Pi)$, the input is a bipartite graph $G = (L \cup R, E)$ with projection constraint $\Pi_e : \Sigma_L \rightarrow \Sigma_R$ on every edge $e \in E$. A labeling σ which assigns values from Σ_L to L and from Σ_R to R satisfies the constraint Π_e on the edge $e = (u, v)$ if $\Pi_e(\sigma(u)) = \sigma(v)$. The objective is to identify if there is a labeling that satisfies all the constraints.*

For every constant $\epsilon > 0$, it is NP-hard to distinguish between the case that a given Label Cover instance has a labeling that satisfies all the constraints vs. no labeling can satisfy more than ϵ fraction of the constraints. This hardness result for Label Cover has been instrumental in showing numerous strong, and sometimes optimal, inapproximability results for various computational problems. However, standard Label Cover seems insufficient as a starting point towards proving hardness results for approximate graph coloring and other 2-variable PCSPs. To circumvent this, the hardness of Label Cover on structured instances such as Unique Games, smooth Label Cover, etc. has been studied.

In his celebrated work proposing the Unique Games Conjecture [20], Khot also proposed the “2-to-1 conjecture” that the strong hardness of Label Cover holds when all the constraints of the Label Cover are 2-to-1 functions. The imperfect completeness version of this conjecture was recently established in a striking sequence of works [13, 14, 22, 23]. Braverman, Khot, and Minzer [9] put forth a stronger conjecture that states that the hardness of Label Cover holds when the distribution of 2-to-1 functions on edges incident on every vertex $u \in L$ is uniform over $\mathcal{F}_{2 \rightarrow 1}$.

► **Definition 20** (Rich 2-to-1 Label Cover instances). *We call a Label Cover instance $\mathcal{G} = (G, \Sigma_L, \Sigma_R, \Pi)$ with $G = (L \cup R, E)$ a rich 2-to-1 instance if the following hold.*

1. *There exists an integer Σ such that $\Sigma_L = [2\Sigma]$, $\Sigma_R = [\Sigma]$, and every projection constraint $\Pi_e, e \in E$ is a 2-to-1 function.*
2. *For every vertex $u \in L$, the distribution of 2-to-1 functions \mathcal{P}_u obtained by first sampling a uniformly random neighbor v of u , and then picking $\Pi_e, e = (u, v)$, is uniform over $\mathcal{F}_{2 \rightarrow 1}(\Sigma)$.*

► **Conjecture 21** (Rich 2-to-1 Conjecture with Perfect Completeness, [9]). *For every $\epsilon > 0$, there exists an integer $\Sigma = \Sigma(\epsilon)$ such that given a rich 2-to-1 Label Cover instance \mathcal{G} , it is NP-Hard to distinguish between the following.*

1. *There is a labeling that satisfies all the constraints of \mathcal{G} .*
2. *No labeling can satisfy more than ϵ fraction of the constraints of \mathcal{G} .*

37:12 Conditional Dichotomy of Boolean Ordered PCSPs

We are now ready to state the hardness part of our dichotomy. It is proved using the Label Cover-Long Code framework. This reduction is standard in the PCSP literature, see e.g., [4], and we include the proof in the full version of the paper.

► **Theorem 22.** *Assume the Rich 2-to-1 Conjecture. Let $PCSP(\Gamma)$ be a Boolean Ordered PCSP such that there exists an absolute constant $\lambda > 0$ with $\max_{i \in [n]} \Phi_f(i) \geq \lambda$ for all functions $f : \{0, 1\}^n \rightarrow \{0, 1\}$, $f \in \text{Pol}(\Gamma)$. Then $PCSP(\Gamma)$ is NP-Hard.*

5 Adversarial 2-to-1 minor

We construct an example of a 2-to-1 minor where the Shapley value alters completely after taking the minor.

► **Theorem 23.** *Let $n \geq 2$ be a positive integer. There exist two monotone Boolean functions $f : \{0, 1\}^{2n} \rightarrow \{0, 1\}$ and $g : \{0, 1\}^n \rightarrow \{0, 1\}$ such that g is a 2-to-1 minor of f with respect to the 2-to-1 function $\pi : [2n] \rightarrow [n]$ defined as $\pi(i) = \lceil \frac{i}{2} \rceil$. Furthermore,*

1. $\Phi_g(1) = \Omega(1)$, and $\Phi_g(j) = o(1)$ for all $j > 1$.
2. $\Phi_f(3) = \Omega(1)$, and $\Phi_f(i) = o(1)$ for all $i \in [2n], i \neq 3$.

Proof. Similar to the proof of Theorem 22, we construct the minor function pair in two steps.

1. First, we construct Boolean monotone functions $f : \{0, 1\}^{2n-1} \rightarrow \{0, 1\}$ and $g : \{0, 1\}^n \rightarrow \{0, 1\}$ such that g is a minor of f with respect to the function $\pi : [2n-1] \rightarrow [n]$ defined as $\pi(1) = 1, \pi(i) = \lceil \frac{i+1}{2} \rceil$ for all $i > 1$. Furthermore, $\Phi_g(1) = \Omega(1)$, and $\Phi_g(j) = o(1)$ for all $j > 1$. We also have $\Phi_f(2) = \Omega(1)$, and $\Phi_f(i) = o(1)$ for all $i \in [2n-1], i \neq 2$.
2. We define the function $f' : \{0, 1\}^{2n} \rightarrow \{0, 1\}$ as

$$f'(y_1, y_2, \dots, y_{2n}) = f(y_1, y_3, \dots, y_{2n})$$

Note that g is a minor of f' with respect to the 2-to-1 function $\pi : [2n] \rightarrow [n]$ defined as $\pi(i) = \lceil \frac{i}{2} \rceil$. Furthermore, by definition, we have $\Phi_{f'}(3) = \Omega(1)$, and $\Phi_{f'}(i) = o(1)$ for all $i \in [2n], i \neq 3$.

Henceforth, our goal is to construct a pair of functions as in the first step above.

We define a partial Boolean function to be a function from $\{0, 1\}^n \rightarrow \{0, 1, ?\}$. A partial Boolean function on n variables is monotone if for all $\mathbf{p} \in \{0, 1\}^n$ and $\mathbf{q} \in \{0, 1\}^n$ such that $\mathbf{p} \leq \mathbf{q}$, if $f(\mathbf{p}) = 1$, then $f(\mathbf{q}) = 1$, and if $f(\mathbf{q}) = 0$, then $f(\mathbf{p}) = 0$.

Now, consider $g : \{0, 1\}^n \rightarrow \{0, 1\}$ to be

$$g(\mathbf{x}) = \begin{cases} 1 & \text{if } \sum_{j=2}^n x_j \geq \frac{51n}{100} \\ 0 & \text{if } \sum_{j=2}^n x_j \leq \frac{49n}{100} \\ x_1 & \text{if } \frac{49n}{100} < \sum_{j=2}^n x_j < \frac{51n}{100} \end{cases}$$

By definition, g is a monotone function, and using Equation (1), we can infer that $\Phi_g(1) = \frac{1}{50}$, and $\Phi_g(j) < \frac{1}{n}$ for all $j > 1$.

We now construct f in three steps. Start with $f = 'g'$.

1. (Preserving the minor) First, set the value of entries of f that are of the form $(x_1, x_2, x_2, \dots, x_n, x_n)$ as

$$f(x_1, x_2, x_2, \dots, x_n, x_n) = g(x_1, x_2, \dots, x_n) \quad \forall \mathbf{x} \in \{0, 1\}^n$$

We then extend it both upwards and downwards i.e. if $f(\mathbf{p})$ is set to 1 and $\mathbf{p} \leq \mathbf{q}$, then set $f(\mathbf{q}) = 1$ as well, and similarly, if $f(\mathbf{q})$ is set to 0, and $\mathbf{p} \leq \mathbf{q}$, then we set $f(\mathbf{p}) = 0$. This ensures that g is a minor of f and that the partial function f is monotone.

2. (Destroying the influence of 1) Next, we ensure that the Shapley value of the coordinate 1 is low by the following operation: consider all \mathbf{y} such that $f(\mathbf{y})$ has not been set in the first step, $y_1 = 0$ and $f(1, y_2, \dots, y_{2n-1})$ is already set to 1 in the first step. Then set $f(\mathbf{y})$ to be 1. Similarly, if \mathbf{y} satisfies $y_1 = 1$ and $f(0, y_2, \dots, y_{2n-1})$ is already set to 0 in the first step, set $f(\mathbf{y})$ to be 0 if it has not been set in the first step.
We claim that the updated partial function f is still a monotone partial function. Consider $\mathbf{p}, \mathbf{q} \in \{0, 1\}^{2n-1}$ such that $\mathbf{p} \leq \mathbf{q}$. Suppose that $f(\mathbf{p})$ is set to be 1. If it is set in the first step, as we extended the partial function upwards in the first step, $f(\mathbf{q}) = 1$ as well. If $f(\mathbf{p})$ is set to be 1 in the second step, it implies that $f(\mathbf{p}')$ has been set to 1 in the first step, where \mathbf{p}' is obtained from \mathbf{p} by setting p_1 to be 1. Let $\mathbf{q}' \in \{0, 1\}^{2n-1}$ be obtained from \mathbf{q} by setting $q_1 = 1$. As $\mathbf{p}' \leq \mathbf{q}'$, $f(\mathbf{q}')$ has been set to 1 in the first step as well. Thus, $f(\mathbf{q})$ is set to be 1 in the second step. The same argument can be used to show that if $f(\mathbf{q}) = 0$, then $f(\mathbf{p}) = 0$ as well.
3. (Adding influence to 2) For all \mathbf{y} for which $f(\mathbf{y}) = ?$ set $f(\mathbf{y}) = y_2$. The fact that the final function f is monotone follows from observing that any completion of a partial monotone function using a monotone function results in a monotone function.

Finally, our goal is to argue about the Shapley value of the coordinates of the function f . First, we show that the Shapley value of the coordinate 1 in f is $o(1)$. Suppose there exists $\mathbf{p} = (0, y_2, y_3, \dots, y_{2n-1})$ and $\mathbf{q} = (1, y_2, y_3, \dots, y_{2n-1})$ such that $f(\mathbf{p}) = 0$ and $f(\mathbf{q}) = 1$. We claim that both the values $f(\mathbf{p})$ and $f(\mathbf{q})$ are set in the first step of the above procedure. Suppose for contradiction that this is not the case. If neither of them is set in the first step, then they will not be set in the second step either, and in the third step, both of them will be assigned the same value, a contradiction. If exactly one of them is set in the first step, then in the second step, the other value would be set to be equal to it, a contradiction as well. Thus, both the values $f(\mathbf{p})$ and $f(\mathbf{q})$ are set in the first step.

Let $B = \mathcal{B}_g(1) \subseteq \{0, 1\}^{n-1}$ be the boundary of the coordinate 1 in g . As $f(\mathbf{q})$ is set to be 1 in the first step, there exists $\mathbf{x} \in \{0, 1\}^n$ such that $g(\mathbf{x}) = 1$ and $(x_1, x_2, x_3, \dots, x_n) \leq \mathbf{q}$. As \mathbf{x} is not less than or equal to \mathbf{p} , we can conclude that $x_1 = 1$ and $g(0, x_2, x_3, \dots, x_n) = 0$. In other words, $(x_2, x_3, \dots, x_n) \in B$. Similarly, there exists \mathbf{x}' such that $g(\mathbf{x}') = 0$ and $(x'_1, x'_2, x'_3, \dots, x'_n) \geq \mathbf{p}$. By the same argument as above, we can conclude that $(x'_2, x'_3, \dots, x'_n) \in B$. Combining the both, we can conclude that there exist $\mathbf{x}, \mathbf{x}' \in B$ such that $(x_2, x_3, \dots, x_n) \leq (y_2, y_3, \dots, y_{2n-2}) \leq (x'_2, x'_3, \dots, x'_n)$. Note that if the above inequality is true for a $(y_2, y_3, \dots, y_{2n-2})$, we directly get that $(y_2, y_3, \dots, y_{2n-2})$ is in the boundary of the coordinate 1 in f .

Observe that the boundary of coordinate 1 in g is the set of vectors (x_2, x_3, \dots, x_n) such that $\frac{49}{100}n \leq \sum_{j=2}^n x_j \leq \frac{51}{100}n$. By the previous argument, we can deduce that the boundary $B' = \mathcal{B}_f(1)$ of the coordinate 1 in f is the set of vectors $\mathbf{y} = (y_2, y_3, \dots, y_{2n-1})$ that satisfy the following property: The number of $i \in [n-1]$ such that both $y_{2i} = y_{2i+1} = 1$ is at least $\frac{49}{100}n$. Similarly, the number of $i \in [n-1]$ such that $y_{2i} = y_{2i+1} = 0$ is at least $\frac{49}{100}n$. Observe that this implies that we require that $\frac{49}{50}n \leq \sum_{j=2}^{2n-1} y_j \leq \frac{51}{50}n$. However, for every integer l such that $\frac{49}{50}n \leq l \leq \frac{51}{50}n$, when we sample a uniformly random vector $\mathbf{y} = (y_2, y_3, \dots, y_{2n-1})$ with $\sum_{j=2}^{2n-1} y_j = l$, the probability that the number of $i \in [n-1]$ such that both $y_{2i} = y_{2i+1} = 1$ is at least $\frac{49}{100}n$ is $o(\frac{1}{n})$. Thus, using Equation (1), we can infer that the Shapley value of the coordinate 1 in f is $o(1)$.

We now show that the coordinate 2 has $\Omega(1)$ Shapley value in f . Consider $\mathbf{y} = (y_1, y_2, \dots, y_{2n-1})$ such that $\frac{49n}{50} < \text{hw}(\mathbf{y}) \leq \frac{51n}{50}$. If the number of i such that both $y_{2i} = y_{2i+1} = 1$ is less than $\frac{49}{100}n$, we have $(y_1, y_3, \dots, y_{2n-1}) \in \mathcal{B}_f(2)$. However, for every integer l such that $\frac{49}{50}n \leq l \leq \frac{51}{50}n$, when we sample a uniformly random \mathbf{y} with

$\text{hw}(\mathbf{y}) = l$, with probability $1 - o(1)$, the number of i such that both $y_{2i} = y_{2i+1} = 1$ is less than $\frac{49}{100}n$. Thus, using Equation (1), we can infer that the Shapley value of the coordinate 2 is $\Omega(1)$ in the function f . Finally, by symmetry, we can observe that $\Phi_f(i) = \Phi_f(3)$ for all $i \geq 3$, and thus, $\Phi_f(i) = o(1)$ for all $i \geq 3$. ◀

References

- 1 Per Austrin, Venkatesan Guruswami, and Johan Håstad. $(2+\epsilon)$ -SAT is NP-hard. *SIAM J. Comput.*, 46(5):1554–1573, 2017.
- 2 Libor Barto. Cyclic operations in promise constraint satisfaction problems. *Dagstuhl workshop The Constraint Satisfaction Problem: Complexity and Approximability, Schloss Dagstuhl, Germany*, 2018. Available at https://www2.karlin.mff.cuni.cz/~barto/Articles/Barto_Dagstuhl18.pdf.
- 3 Libor Barto. Personal communication, 2018-20.
- 4 Libor Barto, Jakub Bulín, Andrei Krokhin, and Jakub Opršal. Algebraic approach to promise constraint satisfaction. *arXiv preprint*, 2018. [arXiv:1811.00970](https://arxiv.org/abs/1811.00970).
- 5 Joshua Brakensiek and Venkatesan Guruswami. New hardness results for graph and hypergraph colorings. In *31st Conference on Computational Complexity, CCC 2016*, pages 14:1–14:27, 2016.
- 6 Joshua Brakensiek and Venkatesan Guruswami. Promise constraint satisfaction: Structure theory and a symmetric Boolean dichotomy. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018*, pages 1782–1801, 2018.
- 7 Joshua Brakensiek, Venkatesan Guruswami, and Sai Sandeep. Conditional dichotomy of boolean ordered promise csps. *arXiv preprint*, 2021. [arXiv:2102.11854](https://arxiv.org/abs/2102.11854).
- 8 Joshua Brakensiek, Venkatesan Guruswami, Marcin Wrochna, and Stanislav Zivný. The power of the combined basic linear programming and affine relaxation for promise constraint satisfaction problems. *SIAM J. Comput.*, 49(6):1232–1248, 2020.
- 9 Mark Braverman, Subhash Khot, and Dor Minzer. On rich 2-to-1 games. In *12th Innovations in Theoretical Computer Science Conference, ITCS 2021*, volume 185 of *LIPIcs*, pages 27:1–27:20, 2021.
- 10 Andrei A. Bulatov. A dichotomy theorem for nonuniform CSPs. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017*, pages 319–330. IEEE Computer Society, 2017.
- 11 Andrei A. Bulatov, Peter Jeavons, and Andrei A. Krokhin. Classifying the complexity of constraints using finite algebras. *SIAM J. Comput.*, 34(3):720–742, 2005.
- 12 Anindya De, Ilias Diakonikolas, and Rocco A. Servedio. The inverse Shapley value problem. *Games Econ. Behav.*, 105:122–147, 2017.
- 13 Irit Dinur, Subhash Khot, Guy Kindler, Dor Minzer, and Muli Safra. On non-optimally expanding sets in grassmann graphs. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018*, pages 940–951. ACM, 2018.
- 14 Irit Dinur, Subhash Khot, Guy Kindler, Dor Minzer, and Muli Safra. Towards a proof of the 2-to-1 games conjecture? In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018*, pages 376–389. ACM, 2018.
- 15 Tomás Feder and Moshe Y. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through datalog and group theory. *SIAM J. Comput.*, 28(1):57–104, 1998.
- 16 Miron Ficak, Marcin Kozik, Miroslav Olsák, and Szymon Stankiewicz. Dichotomy for symmetric boolean PCSPs. In *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019*, volume 132 of *LIPIcs*, pages 57:1–57:12, 2019.
- 17 Peter Jeavons. On the algebraic structure of combinatorial problems. *Theor. Comput. Sci.*, 200(1-2):185–204, 1998.

- 18 Peter Jeavons, David A. Cohen, and Marc Gyssens. Closure properties of constraints. *J. ACM*, 44(4):527–548, 1997.
- 19 Gil Kalai. Social indeterminacy. *Econometrica*, 72(5):1565–1581, 2004.
- 20 Subhash Khot. Hardness results for coloring 3-colorable 3-uniform hypergraphs. In *The 43rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2002*, pages 23–32. IEEE, 2002.
- 21 Subhash Khot. On the power of unique 2-prover 1-round games. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing, STOC 2002*, pages 767–775. ACM, 2002.
- 22 Subhash Khot, Dor Minzer, and Muli Safra. On independent sets, 2-to-2 games, and grassmann graphs. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017*,, pages 576–589. ACM, 2017.
- 23 Subhash Khot, Dor Minzer, and Muli Safra. Pseudorandom sets in Grassmann graph have near-perfect expansion. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018*, pages 592–601. IEEE Computer Society, 2018.
- 24 Andrei A. Krokhnin and Jakub Opršal. The complexity of 3-colouring H -colourable graphs. In *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019*, pages 1227–1239. IEEE Computer Society, 2019.
- 25 G. A. Margulis. Probabilistic characteristics of graphs with large connectivity (in Russian). *Probl. Pered. Inform.*, 10:101–108, 1974.
- 26 Tomasz P. Michalak, Aadithya V. Karthik, Piotr L. Szczepanski, Balaraman Ravindran, and Nicholas R. Jennings. Efficient computation of the Shapley value for game-theoretic network centrality. *J. Artif. Intell. Res.*, 46:607–650, 2013.
- 27 Ramasuri Narayanam and Yadati Narahari. A Shapley value-based approach to discover influential nodes in social networks. *IEEE Trans Autom. Sci. Eng.*, 8(1):130–147, 2011.
- 28 Ryan O’Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014.
- 29 Jan Petr. Monotone functions avoiding majorities, 2020. Undergraduate Thesis. Univerzita Karlova, Matematicko-fyzikální fakulta.
- 30 Nicholas Pippenger. Galois theory for minors of finite functions. *Discrete Mathematics*, 254(1-3):405–419, 2002.
- 31 Lucio Russo. An approximate zero-one law. *Z. Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 61(1):129–139, 1982.
- 32 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. ACM, 1978.
- 33 L. S. Shapley and Martin Shubik. A method for evaluating the distribution of power in a committee system. *American Political Science Review*, 48(3):787–792, 1954.
- 34 Robert Weber. Probabilistic values for games. Cowles Foundation Discussion Papers 471R, Cowles Foundation for Research in Economics, Yale University, 1977. URL: <https://EconPapers.repec.org/RePEc:cwl:cwldpp:471r>.
- 35 Marcin Wrochna and Stanislav Zivný. Improved hardness for H -colourings of G -colourable graphs. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020*, pages 1426–1435. SIAM, 2020.
- 36 Dmitriy Zhuk. A proof of the CSP dichotomy conjecture. *J. ACM*, 67(5):30:1–30:78, 2020.