Bounded Degree Nonnegative Counting CSP

JIN-YI CAI and DANIEL P. SZABO, Department of Computer Sciences, University of Wisconsin-Madison, USA

Constraint satisfaction problems (CSP) encompass an enormous variety of computational problems. In particular, all partition functions from statistical physics, such as spin systems, are special cases of counting CSP (#CSP). We prove a complete complexity classification for every counting problem in #CSP with nonnegative valued constraint functions that is valid when every variable occurs a bounded number of times in all constraints. We show that, depending on the set of constraint functions $\mathcal{F}$, every problem in the complexity class #CSP($\mathcal{F}$) defined by $\mathcal{F}$ is *either* polynomial-time computable for all instances without the bounded occurrence restriction, *or* is #P-hard even when restricted to bounded degree input instances. The constant bound in the degree depends on $\mathcal{F}$. The dichotomy criterion on $\mathcal{F}$ is decidable. As a second contribution, we prove a slightly modified but more streamlined decision procedure (from [14]) to test for the tractability of #CSP($\mathcal{F}$). This procedure on an input $\mathcal{F}$ tells us which case holds in the dichotomy for #CSP($\mathcal{F}$). This more streamlined decision procedure enables us to fully classify a family of *directed* weighted graph homomorphism problems. This family contains both P-time tractable problems and #P-hard problems. To our best knowledge, this is the first family of such problems explicitly classified that are not *acyclic*, thereby the Lovász-goodness criterion of Dyer-Goldberg-Paterson [24] cannot be applied.

CCS Concepts: • **Theory of computation** → **Complexity classes**; *Problems, reductions and completeness*; *Algebraic complexity theory*; • **Mathematics of computing** → Enumeration;

Additional Key Words and Phrases: Computational counting complexity, constraint satisfaction problems, counting CSPs, complexity dichotomy, nonnegative counting CSP, graph homomorphisms

ACM Reference format:
Jin-Yi Cai and Daniel P. Szabo. 2024. Bounded Degree Nonnegative Counting CSP. *ACM Trans. Comput. Theory* 16, 2, Article 5 (March 2024), 18 pages.
https://doi.org/10.1145/3632184

1 INTRODUCTION

Constraint Satisfaction Problems (CSPs) have been a subject of immense interest due to their wide applicability and intrinsic elegance. In particular, counting CSPs, or #CSPs, have been an active subject in computational counting complexity [7, 9, 10, 13, 14, 18, 19, 22], including their approximate solutions [20, 28, 30, 41, 42]. Roughly speaking, an (unweighted) constraint satisfaction problem deals with the following scenario, where there is a set of variables, each taking values over some

J.-Y. Cai supported by NSF CCF-1714275.
D. P. Szabo supported by an REU supplement of NSF CCF-1714275.

Authors' address: J.-Y. Cai and D. P. Szabo, Department of Computer Sciences, University of Wisconsin-Madison, 1210 W. Dayton Street Madison, WI 53706-1613, USA; e-mails: jyc@cs.wisc.edu, dszabo2@wisc.edu.

1942-3454/2024/03-ART5 $15.00
https://doi.org/10.1145/3632184

finite domain D, and a set of constraints, each applied on an (ordered) subsequence of these variables. The #CSP problem on an instance asks how many assignments there are of these variables that satisfy all of the given constraints.

Applications of CSP problems are wide-ranging and varied. They range from within computer science to physical sciences such as physics, chemistry, engineering, even music [1, 38, 47, 52]. Within computer science, belief propagation has been a popular research topic in AI, which is ultimately based on some forms of partition function evaluations [5, 29, 39, 40, 46, 48, 51]. The term partition function, which we define formally later, arises from statistical physics, where one can see special cases of (weighted) counting CSPs in the form of spin systems such as the Ising and Potts models, e.g., [25]. In physical sciences as well as in applications within computer science, the instances of counting CSP problems that occur in practice are often with the additional restriction that variables occur a bounded number of times.

To define (unweighted) #CSP problems formally, let D be a finite domain set, Γ be a set of constraint relations Θ_i, where each Θ_i is a relation on D of arity $r_i = r(\Theta_i) \geq 1$. An instance of #CSP(Γ) is then defined by a set X of n variables over D, and a list of constraints Θ from Γ, and for each constraint Θ in the list a sequence of $r(\Theta)$ variables from X that the constraint is applied to. This defines an n-ary relation R in D^n on the input variables where an assignment $(x_1, \ldots, x_n) \in D^n$ is in R iff all constraints are satisfied. For any fixed Γ, the counting CSP problem #CSP(Γ) consists of all input instances using constraint relations from Γ. The computational problem is to compute the size of R given an arbitrary input instance, where the (worst case) computational complexity is measured in terms of size n of the set of variables and the size of the list of constraints. For a finite (fixed) Γ, this can be simplified to just n, up to a polynomial factor. A complexity dichotomy theorem can classify, depending on Γ, the problem #CSP(Γ) as either computable in polynomial-time (P-time), or #P-complete, with no intermediate cases. Typically, the set Γ is a fixed finite set, which defines the #CSP problem—this Γ is the name of the problem. However, in most dichotomy theorems one can allow infinite sets, where in the P-time computable case we assume the specification of the constraints in the instances counts toward the input size, and in the #P-complete case there is a finite subset $\Gamma_0 \subseteq \Gamma$ such that #CSP(Γ_0) is #P-hard.

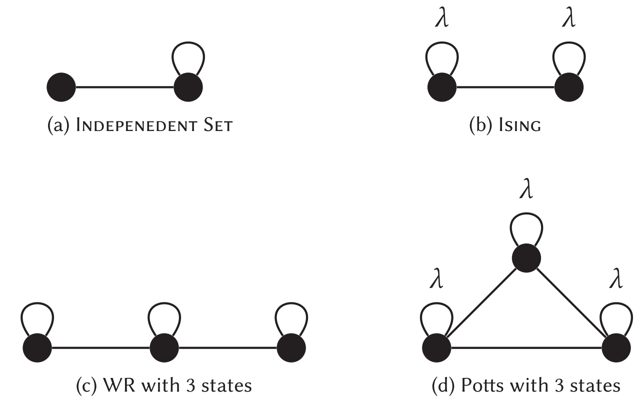
For example, if we let $D = \{0, 1\}$ and $\Gamma = \{\text{OR}_k | k \geq 1\} \cup \{\neq_2\}$, where OR_k is the k-ary OR function, and $\neq_2$ the binary disequality function, then the problem #CSP(Γ) is equivalent to #SAT, the counting Boolean satisfiability problem.

This formulation can be generalized to the weighted setting. In the most general case, the constraint functions can take real or complex values. In this paper we only consider #CSP defined by nonnegatively weighted constraint functions. This means that we replace the constraint language Γ by a set of constraint functions $\mathcal{F}$, where each $f_i \in \mathcal{F}$ has some arity $r_i \geq 1$ and maps D^{r_i} to nonnegative algebraic reals, denoted as $\mathbb{R}_+$.[1] Any given instance I defines a function $F_I : D^n \to \mathbb{R}_+$, such that on each assignment of variables, the value of F_I is the product over the constraint functions in I evaluated on the assignment. The solution to this instance I of #CSP($\mathcal{F}$) is then

$$Z_{\mathcal{F}}(I) = \sum_{(x_1, \ldots, x_n) \in D^n} F_I(x_1, \ldots, x_n). \tag{1}$$

This sum-of-products expression in (1) is called the partition function for an instance of #CSP, with the terminology coming from statistical physics [3]. When all functions in $\mathcal{F}$ are 0-1 valued, then the product is also 0-1 valued and is equivalent to the logical And, and the partition function

[1] Restricting to algebraic numbers is standard in this research area because we wish to state our results in the Turing machine model for strict bit complexity. See [17].



(a) Indepenedent Set

(b) Ising

(c) WR with 3 states

(d) Potts with 3 states

Fig. 1. The graphs corresponding to some well known spin systems.

counts the number of satisfying assignments. Thus, this $Z_{\mathcal{F}}(I)$ generalizes the unweighted case when $\mathcal{F}$ is a set of constraint relations Γ.

As a special case of #CSP, a q-state spin system is a problem on a domain $[q]$ with the constraint language having only a single binary constraint defined by the $q \times q$ interaction matrix A. An instance to this problem is a graph $G = (V, E)$, where the vertices (sites) are considered to be variables (spins) and the edges (bonds) correspond to the interactions between these vertices. The famous Ising model with parameter λ has domain size $q = 2$, and is defined by its interaction matrix $A^{\lambda}_{\text{Ising}} = \left[\begin{smallmatrix} \lambda & 1 \\ 1 & \lambda \end{smallmatrix}\right]$ (see Figure 1(a)). The Potts model (Figure 1(b)) and Widom-Rowlinson model (Figure 1(d)) on 3 states are defined by the following interaction matrices respectively,

$$A^{\lambda}_{\text{3Potts}} = \begin{bmatrix} \lambda & 1 & 1 \\ 1 & \lambda & 1 \\ 1 & 1 & \lambda \end{bmatrix} \quad \text{and} \quad A_{\text{WR}} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}.$$

Familiar problems in computer science can also be expressed in this model; e.g., independent set (IS) is defined by $A_{\text{IS}} = \left[\begin{smallmatrix} 1 & 1 \\ 1 & 0 \end{smallmatrix}\right]$ (Figure 1(a)).

Bulatov [9] proved a sweeping complexity dichotomy for unweighted #CSPs. His proof used deep results from universal algebra. His dichotomy theorem states that #CSP(Γ) is solvable in polynomial-time if Γ satisfies a condition called congruence singularity; it is #P-complete otherwise. Dyer and Richerby [22] gave another proof of this dichotomy using a new P-time tractability criterion, which they proved to be equivalent to congruence singularity.

A nonnegative matrix is block-rank-1 if it becomes a block-diagonal matrix after a permutation of its rows and a permutation of its columns separately, such that all blocks are rank 1 except for possibly one all-zero block. (Here the blocks in the block-diagonal form of the matrix need not be square matrices.) For example, the following matrix (where blank entries are 0's)

$$\begin{bmatrix} A_{0,0} & & A_{0,2} & & & & & \\ A_{1,0} & & A_{1,2} & & & & & \\ & & & & A_{2,4} & & A_{2,6} & \\ & & & & A_{3,4} & & A_{3,6} & \\ & A_{4,1} & & A_{4,3} & & & & \\ & A_{5,1} & & A_{5,3} & & & & \\ & & & & & A_{6,5} & & A_{6,7} \\ & & & & & A_{7,5} & & A_{7,7} \end{bmatrix} \tag{2}$$

is block-rank-1 if each nonzero rectangle of the form $\left[\begin{smallmatrix} A_{i,j} & A_{i,j'} \\ A_{i',j} & A_{i',j'} \end{smallmatrix}\right]$ has rank 1.

For unweighted #CSP, the Dyer-Richerby condition in [22] for polynomial-time tractability in the dichotomy theorem is *Strong Balance*. Let $d = |D|$ be the domain size. We say a constraint language Γ is *Strongly Balanced* if every n-ary relation R defined by an instance of #CSP(Γ) satisfies the following condition:

For any $a, b \geq 1$ and $c \geq 0$ with $a + b + c \leq n$, the following $d^a \times d^b$ matrix M is block-rank-1:

$$M(\boldsymbol{u}, \boldsymbol{v}) = \left|\{\boldsymbol{w} \in D^c : \exists \boldsymbol{z} \in D^{n-c-b-a} \text{ s.t. } (\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}, \boldsymbol{z}) \in R\}\right|.$$

(If $a + b + c = n$, then the quantified statement "$\exists \boldsymbol{z} \in D^{n-c-b-a}$ such that $(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}, \boldsymbol{z}) \in R$" simply means that $(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}) \in R$.)

If we are dealing with $\mathcal{F}$ rather than Γ, and if $\mathcal{F}$ is not a set of 0-1 valued functions, then the existential quantified statement "$\exists \boldsymbol{z}$" has no meaning. It turns out that there are several equivalent notions of *Balance*, which when $\mathcal{F}$ is restricted to a set of 0-1 valued functions (i.e., when $\mathcal{F}$ can be identified with a constraint language Γ) are all equivalent to the notion of Strong Balance; see Lemma 9.4 in [14]. These notions of Balance do not use existential quantifiers (see Definition 1 in Section 2). These notions are central to the #CSP dichotomies for #CSP($\mathcal{F}$) for nonnegative valued $\mathcal{F}$.

The study of #CSPs is closely related to that of counting graph homomorphisms [4, 27, 35, 36]. For two graphs G and H, a graph homomorphism from G to H is a mapping $f : V(G) \to V(H)$ that preserves vertex adjacency. In other words, if $e = \{u, v\} \in E(G)$ then $e' = \{f(u), f(v)\} \in E(H)$, for all edges e in G. The question of interest in counting complexity is the number of graph homomorphisms from one graph to another, which can also be represented by a partition function. If we let A be the $m \times m$ adjacency 0-1 matrix of the graph H, then the number of homomorphisms from G to H can be represented as a sum-of-products partition function as follows,

$$Z_A(G) = \sum_{f: V(G) \to [m]} \prod_{\{u,v\} \in E(G)} A_{f(u), f(v)}.$$

Partition functions of graph homomorphism can represent important physical spin systems such as the Ising, Potts, or Widom-Rowlinson models, as well as many other well known problems in computer science.

Counting graph homomorphisms is a special case of #CSP. In fact, the vertex-edge incidence graph of G defines an input to a #CSP problem, where vertices $V(G)$ are variables and edges $E(G)$ are (applications of binary) constraints, and the constraint language consists of a single binary relation represented by the adjacency matrix A defining the graph homomorphism problem $G \mapsto Z_A(G)$. Just as in #CSPs, the counting graph homomorphism function $Z_A(G)$ can be generalized from the 0-1 unweighted case to the weighted case where A is a real or complex matrix. It is symmetric for an undirected graph H, in which case we also only consider undirected G; for directed graph homomorphisms, A need not be symmetric.

The first dichotomy on counting graph homomorphisms was due to Dyer and Greenhill [21] for undirected graphs. They showed that there is a simple criterion such that if A satisfies the criterion then $G \mapsto Z_A(G)$ is computable in P-time, otherwise it is #P-complete. In fact they proved that if A does not satisfy the criterion then the problem of evaluating $Z_A(G)$ remains #P-complete even when restricted to graphs G with bounded degree Δ, for some Δ depending on A. Computing $G \mapsto Z_A(G)$ when restricted to graphs G with bounded degree Δ is called EVAL$^{(\Delta)}(A)$. The Dyer-Greenhill dichotomy was extended to the nonnegatively weighted case by Bulatov and Grohe in [8]. This dichotomy was then referenced throughout the field, as many other discoveries, including the results on #CSPs, ended up applying it. However, the hardness part of the proof of the Bulatov-Grohe dichotomy theorem required input graphs that have unbounded degrees. When restricted to bounded degree graphs, the worst case complexity of the Bulatov-Grohe dichotomy was left open for 15 years, until it was finally resolved by Govorov, Cai, and Dyer in [26] for graph homomorphisms with nonnegative weights, and extended by Cai and Govorov in [15] for complex weights. Most problems in statistical physics [6, 11, 31] use bounded degree graphs, and also most of the approximation algorithms work on bounded degree graphs [2, 23, 32, 33, 37, 43–45, 49]. Over the Boolean domain where variables take 0-1 values, it is known that the #CSP dichotomy for complex valued constraint functions holds for input instances where each variable occurs at most three times [16].

It has been an open problem to extend the general domain #CSP dichotomies to include the bounded degree case, i.e., where each variable occurs a bounded number of times. It was open even for the 0-1 unweighted case. For the nonnegative cases, this would be the analogous Govorov-Cai-Dyer extension [26] of the Bulatov-Grohe dichotomy for graph homomorphism, but applied to a much broader class of problems, as graph homomorphism is the special case of #CSP($\mathcal{F}$) where $\mathcal{F}$ consists of a single binary function.

In this paper we prove such a dichotomy for bounded degree nonnegative #CSPs. For any finite domain D, any finite set of nonnegative constraint functions $\mathcal{F}$ on D, and any integer $\Delta \geq 0$, we define #CSP$^{(\Delta)}$($\mathcal{F}$) to be the #CSP problem, where the input consists of n variables $x_1, \ldots, x_n$ over D and a sequence of constraint functions $f_1, \ldots, f_m \in \mathcal{F}$ each applied to a subsequence of the n variables, where each variable x_i appears no more than Δ times among $f_1, \ldots, f_m$. Note that in general, a function $f \in \mathcal{F}$ may occur multiple times among $f_1, \ldots, f_m$. We take $n + m$ as the input size. We prove that the same dichotomy criterion in [14] applies to the bounded degree case: if the P-time tractability criterion is not satisfied, then #CSP$^{(\Delta)}$($\mathcal{F}$) remains #P-hard for some $\Delta > 0$. The dichotomy criterion of [14] will be explained in more detail after we introduce some more definitions in Section 2. These notions are further explicated in Theorem 2, and a more technical statement of Theorem 1 is given in Theorem 3.

Theorem 1. *For any finite domain D and any nonnegatively weighted constraint functions $\mathcal{F}$ on D, if $\mathcal{F}$ satisfies the tractability criterion in [14], then* #CSP($\mathcal{F}$) *is P-time computable; otherwise,* #CSP$^{(\Delta)}$($\mathcal{F}$) *is* #P*-hard* [2] *for some $\Delta > 0$.*

For any fixed finite set $\mathcal{F}$ of constraint functions, the arities of $f \in \mathcal{F}$ are bounded. Viewing any instance as a bipartite graph, with the variables $x_1, \ldots, x_n$ on one side and constraints $f_1, \ldots, f_m \in \mathcal{F}$ on the other, with an edge between x_i and f_j if x_i is an input to the function f_j, we can see that the condition for a #CSP instance to be bounded degree corresponds exactly to this bipartite graph having bounded degree.

Our second contribution in this paper is a slightly modified but more streamlined decision procedure (compared to that of [14]) for polynomial-time tractability. This enables us to fully classify a family of *directed* weighted graph homomorphism problems. This family contains both P-time tractable problems and #P-hard problems. To our best knowledge, this is the first family of such problems explicitly classified that are not *acyclic*, thereby the Lovász-goodness criterion of Dyer-Goldberg-Paterson [24] cannot be applied.

2 BALANCE

Several variants of the *Balance* condition have been used in the study of counting constraint satisfaction problems. In addition to the *Strong Balance* condition [22], the following conditions have been introduced in [14]. Recall that $d = |D|$ denotes the domain size.

[2] The problem #CSP$^{(\Delta)}$($\mathcal{F}$) is also no harder than #P under a polynomial-time Turing reduction for any $\mathcal{F}$. The statement for Theorem 1 does not state #P-complete only for the technical reason that functions in #P by definition take nonnegative integer values while the partition function in (1) may take values more general than nonnegative integers.

Definition 1 (Various notions of Balance). We have the following notions:

(1) (Balance) We say $\mathcal{F}$ is *Balanced* if for any $n \geq 2$, any $a \geq 1$ and $b \geq 1$ with $a + b \leq n$, and any instance I of #CSP($\mathcal{F}$) which defines an n-ary function $F_I(x_1, \ldots, x_n)$ over D^n, the following $d^a \times d^b$ matrix M_I is block-rank-1: The rows and columns of M_I are indexed by tuples $\boldsymbol{u} \in D^a$ and $\boldsymbol{v} \in D^b$ respectively, and

$$M_I(\boldsymbol{u}, \boldsymbol{v}) = \sum_{\boldsymbol{w} \in D^{n-a-b}} F_I(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}),$$

for all $\boldsymbol{u} \in D^a, \boldsymbol{v} \in D^b$. If $a + b = n$ then the sum $\sum_{\boldsymbol{w} \in D^{n-a-b}} F_I(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w})$ is simply $F_I(\boldsymbol{u}, \boldsymbol{v})$.

(2) (Weak Balance) We say $\mathcal{F}$ is *Weakly Balanced* if the definition for Balance holds for $b = 1$.

(3) (Primitive Balance) We say $\mathcal{F}$ is *Primitively Balanced* if the definition for Balance holds for $a = b = 1$.

While these three notions may seem to have varying strengths, all three are in fact equivalent by combining the proof in [14] and [34]. See Theorem 2 below. We need the following definition.

Definition 2 (Strong Rectangularity). We say a matrix M is *Rectangular* if after a row permutation and a column permutation it is a block diagonal matrix where all diagonal blocks have no zero entries, except possibly one all zero block. We say a constraint language Γ over D is *Strongly Rectangular* if for any input instance I of #CSP(Γ) which defines an n-ary relation R_I over D^n and for any a and b such that $1 \leq a < b \leq n$, the following $|D|^a \times |D|^{b-a}$ matrix M is rectangular: The rows of M are indexed by $\boldsymbol{u} \in D^a$, the columns of M are indexed by $\boldsymbol{v} \in D^{b-a}$, and

$$M(\boldsymbol{u}, \boldsymbol{v}) = \left| \{\boldsymbol{w} \in D^{n-b} : (\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}) \in R_I\} \right|.$$

THEOREM 2. *The notions of Balance, Weak Balance, and Primitive Balance are equivalent, and can be taken as the P-time tractability criterion of the dichotomy in [14].*

PROOF. It is clear that balance implies weak balance, and weak balance implies primitive balance simply by setting b or a to 1 in Definition 1. We only need to show that primitive balance implies balance, then all three notions are equivalent. For any $\mathcal{F}$ of nonnegative valued constraint functions, Cai, Chen and Lu proved in [14] that (1) if $\mathcal{F}$ is Balanced then it is Weakly Balanced and that the support constraint language of $\mathcal{F}$ satisfies Strong Rectangularity, and (2) the latter two conditions imply that #CSP($\mathcal{F}$) is P-time computable. Here the support constraint language of $\mathcal{F}$ is obtained by taking the support set of each function in $\mathcal{F}$. On the other hand, they also proved that if $\mathcal{F}$ is not Balanced then #CSP($\mathcal{F}$) is #P-hard. Thus, their dichotomy criterion is that $\mathcal{F}$ is Balanced. They also proved in [14] that Primitive Balance implies Weak Balance. Lin and Wang proved in [34] that Weak Balance implies Balance, thus unifying all three notions. □

Remark: We take this opportunity to clarify the dichotomy criteria from [14] for #CSP($\mathcal{F}$). [14] gave two versions of the dichotomy criteria for #CSP($\mathcal{F}$): Theorems 1.1 and 1.2 (pp. 2179-2180 [14]). Theorems 1.1 states that Balance is a complexity dichotomy criterion for #CSP($\mathcal{F}$): if $\mathcal{F}$ satisfies the Balance condition then #CSP($\mathcal{F}$) is P-time computable, otherwise it is #P-hard. Theorems 1.2 states that Weak Balance plus Strong Rectangularity (of the support of $\mathcal{F}$) is also a complexity dichotomy criterion for #CSP($\mathcal{F}$). The proof in [14] is logically based on proving the following two implications: Weak Balance plus Strong Rectangularity imply P-time tractability, and Non-Balance implies #P-hardness. It is easy to see that Balance implies Weak Balance and also Strong Rectangularity, both Theorems 1.1 and 1.2 of [14] follow. If one assumes #P does not collapse to P (a well-accepted hypothesis, without which these dichotomy theorems would not be very meaningful, although still logically valid) then the two dichotomy criteria (Theorems 1.1 and 1.2) in [14] must be equivalent. However, this fact was not proved unconditionally in [14]. Lin and Wang proved unconditionally in [34] that Weak Balance is equivalent to Balance. At the time of writing [14], while it was easy to see that Balance implies Strong Rectangularity, it was not immediately obvious that Weak Balance also implies Strong Rectangularity unconditionally. This is the main reason why Strong Rectangularity was included in the dichotomy criterion for Theorem 1.2 in [14] (another reason is that technically the proof did go through Strong Rectangularity.) With the unconditional proof by Lin and Wang [34], we also know Weak Balance implies Strong Rectangularity, and thus the statement of Theorem 1.2 in [14] can be simplified to just Weak Balance. Finally, it was already proved in [14] that Weak Balance is equivalent to the seemingly even weaker notion of Primitive Balance, thus all three notions are equivalent and are unconditionally the dichotomy criteria as stated in Theorem 2. We will see in Section 4 that Primitive Balance is the crucial notion for the decidability of the dichotomy of #CSP($\mathcal{F}$).

3 BOUNDED DEGREE #CSPS

Lemma 1. *Let M be a nonnegative matrix. If M is not block-rank-1 then neither is MM^T.*

Proof. If every two rows of M are either proportional or their nonzero entries are on disjoint subsets of columns, then M would be block-rank-1. Thus there are rows M_i and M_j, such that they are linearly independent, and the subsets of columns where their nonzero entries occur intersect. Being nonnegative, the latter condition implies that they are not orthogonal. So by the Cauchy-Schwarz inequality we have

$$0 < (M_i \cdot M_j)^2 < (M_i \cdot M_i)(M_j \cdot M_j).$$

Letting $A = MM^T$ we find four nonzero elements $A_{i,i}, A_{i,j} = A_{j,i}$, and $A_{j,j}$ satisfying $A_{i,i}A_{j,j} > A_{i,j}^2 > 0$, so A is not block-rank-1. □

We can now prove our main result, i.e., if a nonnegative constraint set $\mathcal{F}$ does not satisfy the Balance condition, then #CSP$^{(\Delta)}(\mathcal{F})$ is #P-hard for some $\Delta > 0$.

Theorem 3. *If $\mathcal{F}$ is Primitively Balanced, then the problem #CSP($\mathcal{F}$) without degree restriction is computable in polynomial-time, otherwise #CSP$^{(\Delta)}(\mathcal{F})$ is #P-hard for some $\Delta > 0$.*

Proof. As the tractability was already shown in [14], we only need to prove the hardness part. Let $\mathcal{F}$ be any set of nonnegatively weighted constraint functions that is not Primitively Balanced. Then for some instance I on n variables, the $|D| \times |D|$ matrix M defined by

$$M(x_1, x_2) = \sum_{(x_3,\ldots,x_n)\in D^{n-2}} F_I(x_1, x_2, x_3, \ldots, x_n)$$

is not block-rank 1. This instance only uses a finite subset $\mathcal{F}' \subseteq \mathcal{F}$ of constraint functions. In the rest of this proof we will show that #CSP$^{(\Delta)}(\mathcal{F}')$ is #P-hard for some $\Delta > 0$, and therefore we may as well assume $\mathcal{F}$ is a finite set of nonnegatively weighted constraint functions that is not Primitively Balanced.

Let $A = MM^T$. Then the matrix A is symmetric, nonnegative, and not block-rank 1 by Lemma 1. This A defines a graph homomorphism problem. We know from [26] that the bounded degree nonnegative graph homomorphism problem EVAL$^{(\Delta)}(A)$ is #P-hard for some $\Delta > 0$, where the constant Δ depends on A. Here we show a reduction EVAL$^{(\Delta)}(A) \le_P$ #CSP$^{(\Delta')}(\mathcal{F})$, for some $\Delta' > 0$, thereby showing that #CSP$^{(\Delta')}(\mathcal{F})$ is #P-hard for some $\Delta' > 0$.

To show that, consider a graph G with maximum degree at most Δ as input instances of EVAL$^{(\Delta)}(A)$. We can compute the value $Z_A(G)$ by expressing it as the partition function $Z_{\mathcal{F}}(I(G))$ for some instance $I(G)$ of polynomial size in #CSP$^{(\Delta')}(\mathcal{F})$. We will use the instance I that defines the matrix M as having constant size, as it does not depend on G. We construct $I(G)$, an input to #CSP($\mathcal{F}$), with the additional property that every variable occurs at most Δ' times, such that $Z_{\mathcal{F}}(I(G)) = Z_A(G)$, as follows.

We note that each entry in A is a dot product of two row vectors in M, and every entry of M is a sum over $|D|^{n-2}$ evaluations of F_I.

We will define a (binary) gadget, which is an instance of #CSP($\mathcal{F}$) of bounded size, with two specially labelled variables called x^* and x^{**}. Copies of this gadget will be used in the construction of (global) #CSP($\mathcal{F}$) instances. A (binary) gadget may have other variables, but in the global #CSP($\mathcal{F}$) instances all constraints applied to the variables other than x^* and x^{**} in each copy are from within the gadget. We define $I(G)$ by replacing every edge in G by a copy of this gadget. Formally, the construction is as follows, where the gadget simulates the edge weights in A in the #CSP setting.

1. Define a variable x_v over D for every $v \in V(G)$.
2. For each $e = uv \in E(G)$ we add $2n - 3 = 1 + 2(n-2)$ variables y_e, $\mathbf{z}_e = (z_{e,3}, \dots, z_{e,n})$, and $\mathbf{z}'_e = (z'_{e,3}, \dots, z'_{e,n})$ over the same domain D. We then apply two copies of I as constraints, one copy over the variables $(x_u, y_e, \mathbf{z}_e)$ and another over $(x_v, y_e, \mathbf{z}'_e)$. There are no other constraints applied on the variables y_e, $\mathbf{z}_e$ and $\mathbf{z}'_e$.

Thus, one copy of the binary gadget is used to replace each edge $uv \in E(G)$, with the two specially labelled variables x^* and x^{**} identified with x_u and x_v. This gadget defines the following constraint function with input variables x_u and x_v taking values in D,

$$\begin{aligned}
&\sum_{y_e \in D} \left(\sum_{\mathbf{z}_e \in D^{n-2}} F_I(x_u, y_e, \mathbf{z}_e) \sum_{\mathbf{z}'_e \in D^{n-2}} F_I(x_v, y_e, \mathbf{z}'_e) \right) \\
&= \sum_{y_e \in D} M_I(x_u, y_e) M_I(x_v, y_e) \\
&= (MM^T)(x_u, x_v) \\
&= A(x_u, x_v).
\end{aligned}$$

Therefore, the gadget defines the edge constraint function represented by the matrix A, which is exactly the edge weights in $Z_A(G)$.

Since G has bounded degrees, and I has constant size, $I(G)$ also has bounded degrees. The variables y_e are "local" to each edge $e \in E(G)$ in the sense that there are no other constraints on them except in the definition of the gadget for this edge e, which has constant size. The same is true for $\mathbf{z}_e$ and $\mathbf{z}'_e$.

Also, the size of $I(G)$ is linear in the size of G, so this is a polynomial-time reduction. That $Z_{\mathcal{F}}(I(G)) = Z_A(G)$ follows from the fact that A is the edge constraint function by our construction. □

Remark: The relationship of Δ from [26] to Δ' in the proof of Theorem 3 is as follows. Assuming $\mathcal{F}$ is not Primitively Balanced, we have an instance I which defines the matrix M. This instance I has a constant size and a constant maximum degree d (this depends on $\mathcal{F}$ but not the input graph G to $\text{EVAL}^{(\Delta)}(A)$, where $A = MM^T$). If G has maximum degree at most Δ, then the construction in the proof reveals that the maximum degree of $I(G)$ is at most $\max\{d\Delta, 2d\}$. Thus, we can take $\Delta' = \max\{d\Delta, 2d\}$. We also note that being not block-rank 1 also includes the possibility of being not rectangular.

The constant Δ in the statement of Theorem 1 (equivalently in Theorem 3) depends on $\mathcal{F}$. (This is called Δ' in the proof of Theorem 3.) This dependence is already present in [26] that $\text{EVAL}^{(\Delta)}(A)$

is #P-hard for some $\Delta > 0$, where the constant Δ depends on A but it is not explicitly given. In addition, there is also the dependence of d on $\mathcal{F}$ in the proof of Theorem 3. In [21], Dyer and Greenhill conjectured that a universal constant $\Delta = 3$ suffices for $\text{EVAL}^{(\Delta)}(A)$ where A is a 0-1 symmetric matrix. This is still open. It is open whether a universal constant Δ, or a constant that only depends on the domain size $|D|$, may suffice for even the 0-1 case, for both $\text{EVAL}^{(\Delta)}(A)$ and $\#\text{CSP}^{(\Delta)}(\mathcal{F})$. In [16] it is known that the constant 3 suffices for the Boolean domain. Xia [50] proved that a universal Δ *does not* exist for $\text{EVAL}^{(\Delta)}(A)$ for complex symmetric matrices A, assuming #P does not collapse to P.

4 EFFECTIVE DICHOTOMY AND A FAMILY OF DIRECTED GH

The condition of Balance (Definition 1) in the dichotomy refers to all instances I of $\#\text{CSP}(\mathcal{F})$, which is an infinitary statement. Thus, it is not immediate that the tractability condition in Theorem 3 is decidable. However, the condition is the same as the one in [14] for the unbounded degree case, and in that paper a decision procedure is given. Here we give a slight modification of the same decision procedure for the dichotomy in Theorem 3. This form is more symmetric and allows us to apply the following procedure more effectively.

Theorem 4. *The polynomial-time tractability condition of balance in Theorem 3 can be tested by the following two conditions. Measured in the size of D and $\mathcal{F}$, this shows that the decision problem for testing balance is in* NP.

(A) There is a Mal'tsev polymorphism $\varphi : D^3 \to D$ for (the support of) every function in $\mathcal{F}$. This means that φ preserves all relations defined as the support of some function in $\mathcal{F}$ (this is called a polymorphism), and satisfies $\varphi(a, a, b) = \varphi(b, a, a) = b$ for all $a, b \in D$ (Mal'tsev property). The existence of such a mapping φ is equivalent to Strong Rectangularity.

(B) For all $\alpha \neq \beta, \kappa \neq \lambda \in D$ there is a bijection $\pi : D^6 \to D^6$ satisfying the following three properties:

(1) $\pi((\alpha, \alpha, \alpha, \beta, \beta, \beta)) = (\alpha, \alpha, \alpha, \beta, \beta, \beta)$.

(2) $\pi((\kappa, \lambda, \kappa, \lambda, \kappa, \lambda)) = (\lambda, \kappa, \lambda, \kappa, \lambda, \kappa)$.

(3) Any function $f \in \mathcal{F}$ with arity r is invariant under π, that is, for any sequence $(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_r)$ of length r of 6-tuples where $\boldsymbol{y}_i = (y_{i,1}, \ldots, y_{i,6}) \in D^6$ for $1 \leq i \leq r$, the following holds:

$$\prod_{j \in [6]} f(y_{1,j}, \ldots, y_{r,j}) = \prod_{j \in [6]} f(\pi(\boldsymbol{y}_1)_j, \ldots, \pi(\boldsymbol{y}_r)_j). \tag{3}$$

The only difference in the statement of this decision criterion compared to the one stated in [14] is a more symmetric expression for condition *2* of (B). The proof closely follows the proof in [14]; for completeness, we will give a proof in an appendix. Alternatively, one can apply a perm $\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 3 & 2 & 4 & 6 & 5 \end{pmatrix}$ and show that it is an automorphism of a relational structure $(\mathfrak{D}, \mathfrak{F})$, which is defined to be the 6th power of $\#\text{CSP}(\mathcal{F})$, to derive this form of the decision criterion from the form proved in [14]. [3] However, the more symmetric form in Theorem 4 makes the criterion more easily applicable. We demonstrate this by proving new tractable and intractable cases of directed graph homomorphisms that were previously unknown. The tractability or intractability of these problems were decidable in principle by the previous method in [14] which was an extension of the method by Dyer and Richerby for the unweighted case [22]; however, the decision procedure of the previous method seemed in practice too complicated to be useful.

Dyer, Goldberg and Paterson in [24] proved a decidable complexity dichotomy for (unweighted) directed graph homomorphisms that is restricted to directed *acyclic* graphs. Their polynomial-time tractability criterion is an interesting condition of being *layered* and *Lovász-good* for directed

[3] In [14], the corresponding condition *2* of (B) is stated as $\pi((\kappa, \kappa, \lambda, \lambda, \lambda, \kappa)) = (\lambda, \lambda, \kappa, \kappa, \kappa, \lambda)$.

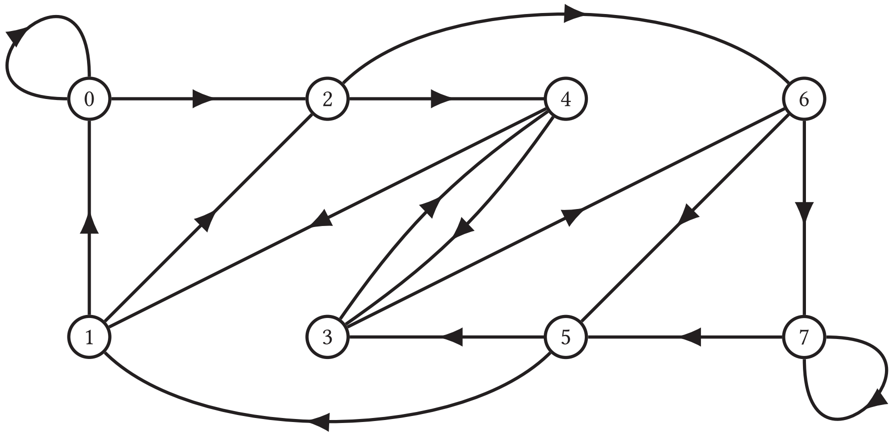


Fig. 2. A tractable binary relation represented by a directed graph. The adjacency matrix is given in Equation (4).

acyclic graphs. They state in [24] that "An interesting feature of the dichotomy, which is absent from previously-known dichotomy results, is that there is a rich supply of tractable graphs H with complex structure". Going beyond directed *acyclic* graphs, as it is done with Theorem 3 and 4, is expected to yield even more polynomial-time tractable problems. However, up until now we don't have any interesting concrete examples. (Part of the reason is probably that testing for the tractability criterion, while decidable, is not a simple matter; see below.) The dichotomy theorem in this paper applies more generally without the *acyclicity* restriction. We now give a family of non-acyclic directed graph homomorphism problems that we can completely classify using our tractability criterion in Theorem 4. To our best knowledge, this is the first such explicit family that can be classified, going beyond the Lovász-goodness criterion [24].

To start, if we take all nonzero $A_{i,j} = 1$ in Equation (2), we get a binary relation that defines a polynomial-time tractable problem. This represents an adjacency matrix of a directed graph H illustrated in Figure 2. In fact, we can give an infinite family of tractable #CSP based on a weighted binary constraint function given in Equation (4). These problems were considered in [12], where it was shown that, while the complexity of the #CSP defined by these relations is provably decidable, the decision criterion was yet too complicated, therefore for what values of $A_{i,j}$ in Equation (2) the problem it defines is tractable for #CSP was not resolved.

We will show that a nonnegative binary constraint function given in the form of Equation (2) with positive entries $A_{i,j}$ defines a tractable #CSP iff the constraint function is a positive multiple of the function in Equation (4)

$$A = \begin{bmatrix} 1 & & v & & & & & \\ u & & uv & & & & & \\ & & & & y & & yv & \\ & & & & yu & & yuv & \\ & x & & xv & & & & \\ & xu & & xuv & & & & \\ & & & & & z & & zv \\ & & & & & zu & & zuv \end{bmatrix} \tag{4}$$

for some positive reals u, v, x, y, z, with the condition that $z = xy$.

The #CSP problem it defines is on a domain of size 8 and has a constraint function set $\mathcal{F}$ consisting of a single binary (but not symmetric) constraint function given by the matrix in (4). The directed graph defined by the support of the function is not acyclic.

We first prove the relation in (4) for any positive u, v, x, y and $z = xy$ defines a tractable problem. After that we prove the reverse direction.

To apply Theorem 4, we treat $D = \{0, \dots, 7\}$ as a vector space $(\mathrm{GF}[2])^3$ of size 8, represented by three bit strings $\{0,1\}^3$. Then we can take the Mal'tsev polymorphism $\varphi : D^3 \to D$ where $\varphi(x, y, z) = x - y + z$. (Here $-$ is the same as $+$ in $\mathrm{GF}[2^3]$.) It is Mal'tsev because $\varphi(a, a, b) = \varphi(b, a, a) = b$ for all $a, b \in D$. The directed edge relation given by the matrix (2) (by setting all 16 nonzero entries $A_{i,j} = 1$) is given symbolically as follows:

$$A_{i,j} = 1 \text{ where } i = i_1 i_2 i_3, j = j_1 j_2 j_3 \in D \iff j_1 = i_2 \text{ and } j_3 = i_1. \tag{5}$$

One can easily check that φ is a polymorphism, i.e., for any (x, x'), (y, y') and (z, z'), if $A_{x,x'} = A_{y,y'} = A_{z,z'} = 1$ then $A_{\varphi(x,y,z),\varphi(x',y',z')} = 1$.

For the second requirement (B), there are $|D|^6! = 262144! > 10^{1306590}$ bijections from D^6 to D^6, so it is infeasible to enumerate them. However, the following map π works.

Let $M : \{0,1\}^6 \to \{0,1\}^6$ be the bijection that swaps (010101) and (101010) and acts as the identity on the rest. In particular, M preserves the Hamming weight. Let M_i be the ith output bit of M, and let $x_1, x_2, \dots, x_6 \in D$ where each x_i consists of three bits $x_i = a_i b_i c_i \in \{0,1\}^3$.

We write $\boldsymbol{x} = (x_1, \dots, x_6) = (a_1 b_1 c_1, \dots, a_6 b_6 c_6) \in D^6$. We will also represent $\boldsymbol{x}$ bitwise using $\boldsymbol{a} = (a_1, \dots, a_6) \in \{0,1\}^6$, $\boldsymbol{b} = (b_1, \dots, b_6) \in \{0,1\}^6$, and $\boldsymbol{c} = (c_1, \dots, c_6) \in \{0,1\}^6$, and write $\boldsymbol{x} = \boldsymbol{abc}$.

Then we define

$$\begin{aligned} \pi(\boldsymbol{x}) &= \pi(x_1, x_2, \dots, x_6) = \pi(a_1 b_1 c_1, a_2 b_2 c_2, \dots, a_6 b_6 c_6) \\ &= (M_1(\boldsymbol{a}) M_1(\boldsymbol{b}) M_1(\boldsymbol{c}),\ M_2(\boldsymbol{a}) M_2(\boldsymbol{b}) M_2(\boldsymbol{c}),\ \dots,\ M_6(\boldsymbol{a}) M_6(\boldsymbol{b}) M_6(\boldsymbol{c})), \end{aligned}$$

where we recall that M_i denotes the ith output bit of the function $M : \{0,1\}^6 \to \{0,1\}^6$ that swaps (010101) and (101010), and is the identity on the rest.

Thus, we have defined $\pi : D^6 \to D^6$ in such a way that, if we write the input $\boldsymbol{x} = (x_1, x_2, \dots, x_6) \in D^6$ as a 3×6 matrix over $\{0,1\}$ where each x_j is a written as a column

$$\begin{bmatrix} a_1 & a_2 & \cdots & a_6 \\ b_1 & b_2 & \cdots & b_6 \\ c_1 & c_2 & \cdots & c_6 \end{bmatrix},$$

then the output of π is also a 3×6 matrix over $\{0,1\}$ obtained by applying M to each row. In particular, the ith row of the output only depends on the ith row of the input.

Claim 1. *The mapping $\pi : D^6 \to D^6$ satisfies properties 1, 2, and 3 of (B) in Theorem 4 for all $\alpha \neq \beta, \kappa \neq \lambda \in D$.* [4]

Proof. (of Claim 1) Property 1 holds by construction, because M fixes pointwise 0^6, 1^6, $0^3 1^3$, $1^3 0^3$. It satisfies property 2 because in addition it swaps (010101) and (101010).

For property 3 Equation (3) is expressed as (more details are given below)

$$u^{\sum c_i} v^{\sum d_i} x^{\sum a_i} y^{\sum b_i} \left(\frac{z}{xy}\right)^{\sum a_i b_i} = u^{\sum M_i(\boldsymbol{c})} v^{\sum M_i(\boldsymbol{d})} x^{\sum M_i(\boldsymbol{a})} y^{\sum M_i(\boldsymbol{b})} \left(\frac{z}{xy}\right)^{\sum M_i(\boldsymbol{a}) M_i(\boldsymbol{b})} \tag{6}$$

[4] In Theorem 4 the mapping π may depend on $\alpha \neq \beta$, $\kappa \neq \lambda \in D$, but the π in Claim 1 is in fact the same for all α, β, κ, λ.

where M_i denotes the ith output bit of M, all sums (as well as those below) range from $i = 1$ to 6. Because M preserves Hamming weight, we get $\sum_{i=1}^{6} a_i = \sum_{i=1}^{6} M_i(\boldsymbol{a})$, and similarly for $\boldsymbol{b}, \boldsymbol{c}, \boldsymbol{d}$, and since $z = xy$, this equation holds. □

We now investigate *all* possible tractable cases represented by a matrix A in the form (2) with positive entries $A_{i,j}$. By the necessary condition of Balance applied to the binary function A itself, all relevant four 2×2 blocks must be of rank 1, in order to be tractable, i.e., it takes the form in (4) with some positive u, v, x, y and z, up to a global positive factor. We prove that, up to a global positive factor, a nonnegative matrix A in (2) with the given support structure defines a tractable partition function $Z_{\mathcal{F}}(\cdot)$ where $\mathcal{F} = \{A\}$ iff A has the form in (4) for some positive reals u, v, x, y and $z = xy$; otherwise $Z_{\mathcal{F}}(\cdot)$ is #P-hard.

This #CSP problem has $\mathcal{F}$ consisting of a single binary (nonsymmetric) constraint function defined by the matrix A. By its support structure and the Mal'tsev polymorphism we already satisfied condition (A) of Theorem 4. So, the problem is tractable if and only if for all $\alpha \neq \beta, \kappa \neq \lambda \in D$, there is a bijection $\pi : D^6 \to D^6$ that satisfies the following three properties:

(1) $\pi((\alpha, \alpha, \alpha, \beta, \beta, \beta)) = (\alpha, \alpha, \alpha, \beta, \beta, \beta)$.
(2) $\pi((\kappa, \lambda, \kappa, \lambda, \kappa, \lambda)) = (\lambda, \kappa, \lambda, \kappa, \lambda, \kappa)$.
(3) For the binary function represented by the matrix A, and any 6-tuples $\boldsymbol{x}, \boldsymbol{y} \in D^6$, where $\boldsymbol{x} = (x_1, \ldots, x_6)$ and $\boldsymbol{y} = (y_1, \ldots, y_6)$, we have the following invariance under π,

$$\prod_{i \in [6]} A_{x_i, y_i} = \prod_{i \in [6]} A_{\pi(\boldsymbol{x})_i, \pi(\boldsymbol{y})_i}. \tag{7}$$

Suppose there is a bijection $\pi : D^6 \to D^6$ that satisfies these properties. Let $\pi(x_1, \ldots, x_6) = (\pi_1(x_1, \ldots, x_6), \ldots, \pi_6(x_1, \ldots, x_6))$, where $\pi_i : D^6 \to D$, and $\pi_i(x_1, \ldots, x_6)$ is the ith output entry in D of π. Denote the three bits of $\pi_i(x_1, \ldots, x_6)$ as $f_i(x_1, \ldots, x_6)$, $g_i(x_1, \ldots, x_6)$, and $h_i(x_1, \ldots, x_6)$. To satisfy property 3, we need each π_i to preserve the edge relation 5, i.e., preserve the support set. Since π is a bijection, if we verify that a nonzero LHS of (7) implies a nonzero RHS of (7), we will also have proved that it maps a zero LHS to a zero RHS; thus it preserves the support set. So, consider arbitrary

$$\boldsymbol{x} = (x_1, \ldots x_6) = (a_1 b_1 c_1, \ldots, a_6 b_6 c_6) \in D^6, \ \boldsymbol{y} = (y_1, \ldots, y_6) = (b_1 d_1 a_1, \ldots, b_6 d_6 a_6) \in D^6.$$

This is a generic pair of tuples such that $A_{x_i, y_i} \neq 0$, for $1 \leq i \leq 6$. We need $A_{\pi_i(\boldsymbol{x}), \pi_i(\boldsymbol{y})} \neq 0$ for each i. As before we will also represent $\boldsymbol{x}$ bitwise using $\boldsymbol{a} = (a_1, \ldots, a_6) \in \{0,1\}^6$, $\boldsymbol{b} = (b_1, \ldots, b_6) \in \{0,1\}^6$, $\boldsymbol{c} = (c_1, \ldots, c_6) \in \{0,1\}^6$ and $\boldsymbol{d} = (d_1, \ldots, d_6) \in \{0,1\}^6$, and write $\boldsymbol{x} = \boldsymbol{abc}$ and $\boldsymbol{y} = \boldsymbol{bda}$.

Therefore, by the edge relation, we have $f_i(\boldsymbol{abc}) = h_i(\boldsymbol{bda})$. Hence, f_i is independent of the third part of the input $\boldsymbol{c}$. Also, $g_i(\boldsymbol{abc}) = f_i(\boldsymbol{bda})$, so f_i is also independent of the second part of the input, and therefore is in fact a function on the first part of the input only. Thus, there is a function $f_i' : \{0,1\}^6 \to \{0,1\}$, such that $f_i(\boldsymbol{abc}) = f_i'(\boldsymbol{a})$. Then, from $f_i'(\boldsymbol{a}) = h_i(\boldsymbol{bda})$, we know that h_i is actually a function of its third part of the input only. From $g_i(\boldsymbol{abc}) = f_i'(\boldsymbol{b})$, we know that g_i is a function of its second part of the input only. Thus, there are functions $g_i', h_i' : \{0,1\}^6 \to \{0,1\}$, such that $g_i(\boldsymbol{abc}) = g_i'(\boldsymbol{b})$ and $h_i(\boldsymbol{abc}) = h_i'(\boldsymbol{c})$. Putting these together, we see $f_i'(\boldsymbol{a}) = g_i'(\boldsymbol{a}) = h_i'(\boldsymbol{a})$. Since $\boldsymbol{a} \in \{0,1\}^6$ is arbitrary, we get $f_i' = g_i' = h_i'$. We now rename these as $M_i := f_i' = g_i' = h_i'$. In other words, $\pi : D^6 \to D^6$ has the form $\pi = (\pi_1, \pi_2, \ldots, \pi_6)$ where $\pi_i(\boldsymbol{abc}) = M_i(\boldsymbol{a}) M_i(\boldsymbol{b}) M_i(\boldsymbol{c})$. We will name the mapping $M = (M_1, M_2, \ldots, M_6) : \{0,1\}^6 \to \{0,1\}^6$ with M_i being its ith bit output. Since π is a bijection, so must be M.

Now we pick $\alpha = \left[\begin{smallmatrix} 0 \\ 1 \\ 0 \end{smallmatrix}\right], \beta = \left[\begin{smallmatrix} 1 \\ 0 \\ 0 \end{smallmatrix}\right], \kappa = \left[\begin{smallmatrix} 1 \\ 0 \\ 1 \end{smallmatrix}\right], \lambda = \left[\begin{smallmatrix} 0 \\ 1 \\ 1 \end{smallmatrix}\right] \in D$. (We write them as column vectors to visually aid the readers.) Then clearly $\alpha \neq \beta, \kappa \neq \lambda$. We have $(\alpha, \alpha, \alpha, \beta, \beta, \beta) = \left[\begin{smallmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{smallmatrix}\right]$. For any π defined by a bijection M as above, it satisfies property 1. above iff M pointwise fixes 000000, 000111 and 111000. We also have $(\kappa, \lambda, \kappa, \lambda, \kappa, \lambda) = \left[\begin{smallmatrix} 1&0&1&0&1&0 \\ 0&1&0&1&0&1 \\ 1&1&1&1&1&1 \end{smallmatrix}\right]$, and $(\lambda, \kappa, \lambda, \kappa, \lambda, \kappa) = \left[\begin{smallmatrix} 0&1&0&1&0&1 \\ 1&0&1&0&1&0 \\ 1&1&1&1&1&1 \end{smallmatrix}\right]$. Hence π satisfies property 2. above iff M fixes 111111 and swaps 010101 with 101010. Below we assume M is a bijection that satisfies these properties.

It is easy to verify that for any bijection $M : \{0,1\}^6 \to \{0,1\}^6$, the mapping π defined above preserves the support (defined by nonzero values of the LHS in (7)). Since M is a bijection, in the following we only need to verify that (7) holds for any nonzero LHS of (7) (as any zero LHS automatically has a zero RHS).

Now we show that Equation (7) in property 3 is the same as (6).

To see that, take any nonzero of the LHS in (7) with $\boldsymbol{x} = \boldsymbol{abc}$ and $\boldsymbol{y} = \boldsymbol{bda}$, then the LHS is evaluated as

$$\begin{aligned} &\prod_{i \in [6]} (u^{c_i} v^{d_i})^{(1-a_i)(1-b_i)} (y u^{c_i} v^{d_i})^{(1-a_i)b_i} (x u^{c_i} v^{d_i})^{a_i(1-b_i)} (z u^{c_i} v^{d_i})^{a_i b_i} \\ &= \prod_{i \in [6]} u^{c_i} v^{d_i} x^{a_i} y^{b_i} \left(\frac{z}{xy}\right)^{a_i b_i} \\ &= u^{\sum c_i} v^{\sum d_i} x^{\sum a_i} y^{\sum b_i} \left(\frac{z}{xy}\right)^{\sum a_i b_i} \end{aligned}$$

The expression for the RHS is nearly identical, with $M_i(\boldsymbol{a})$ substituting a_i, and so on.

Now it is clear that if $z = xy$, then the partition function $Z_{\mathcal{F}}(\cdot)$ is tractable, witnessed by any π defined by a bijection on $\{0,1\}^6$ that preserves Hamming weight, pointwise fixes 000000, 111111, 000111, 111000, and swaps 010101 and 101010.

Next, assume $z \neq xy$. We can multiply both sides of (6) over all 2^6 possible $\boldsymbol{c}$ and all 2^6 possible possible $\boldsymbol{d}$, and using the fact that M is a bijection, to get

$$u^{6 \cdot 2^{11}} v^{6 \cdot 2^{11}} x^{2^{12} \sum a_i} y^{2^{12} \sum b_i} \left(\frac{z}{xy}\right)^{2^{12} \sum a_i b_i} = u^{6 \cdot 2^{11}} v^{6 \cdot 2^{11}} x^{2^{12} \sum M_i(\boldsymbol{a})} y^{2^{12} \sum M_i(\boldsymbol{b})} \left(\frac{z}{xy}\right)^{2^{12} \sum M_i(\boldsymbol{a}) M_i(\boldsymbol{b})},$$

which is equivalent to

$$x^{\sum a_i} y^{\sum b_i} \left(\frac{z}{xy}\right)^{\sum a_i b_i} = x^{\sum M_i(\boldsymbol{a})} y^{\sum M_i(\boldsymbol{b})} \left(\frac{z}{xy}\right)^{\sum M_i(\boldsymbol{a}) M_i(\boldsymbol{b})}. \tag{8}$$

Then multiplying over all 2^6 possible 6-tuples $\boldsymbol{b}$,

$$x^{2^6 \sum a_i} y^{6 \cdot 2^{11}} \left(\frac{z}{xy}\right)^{2^5 \sum a_i} = x^{2^6 \sum M_i(\boldsymbol{a})} y^{6 \cdot 2^{11}} \left(\frac{z}{xy}\right)^{2^5 \sum M_i(\boldsymbol{a})},$$

we get $\left(\frac{xz}{y}\right)^{\sum a_i} = \left(\frac{xz}{y}\right)^{\sum M_i(\boldsymbol{a})}$.

Suppose $xz \neq y$, it follows that $\sum_{i=1}^6 a_i = \sum_{i=1}^6 M_i(\boldsymbol{a})$ for any $\boldsymbol{a}$, i.e., M preserves Hamming weight. Then it follows from (6) that

$$\left(\frac{z}{xy}\right)^{\sum a_i b_i} = \left(\frac{z}{xy}\right)^{\sum M_i(\boldsymbol{a}) M_i(\boldsymbol{b})}.$$

But if we take $\boldsymbol{a} = 000111$ and $\boldsymbol{b} = 010101$, we have $M(\boldsymbol{a}) = \boldsymbol{a}$ and $M(\boldsymbol{b}) = 101010$. Then $\sum_{i=1}^6 a_i b_i = 2$, but $\sum_{i=1}^6 M_i(\boldsymbol{a}) M_i(\boldsymbol{b}) = 1$. This is a contradiction to (6), since $z \neq xy$. Hence we conclude that the partition function $Z_{\mathcal{F}}(\cdot)$ is #P-hard.

Next, suppose that $xz = y$, then (8) is simplified to

$$x^{\sum a_i} y^{\sum b_i} x^{-2\sum a_i b_i} = x^{\sum M_i(\boldsymbol{a})} y^{\sum M_i(\boldsymbol{b})} x^{-2\sum M_i(\boldsymbol{a}) M_i(\boldsymbol{b})}. \tag{9}$$

Multiplying over all possible $\boldsymbol{a}$, this becomes

$$x^{6\cdot 2^5} y^{2^6 \sum b_i} x^{-2\cdot 2^5 \sum b_i} = x^{6\cdot 2^5} y^{2^6 \sum M_i(\boldsymbol{b})} x^{-2\cdot 2^5 \sum M_i(\boldsymbol{b})}.$$

This simplifies to

$$\left(\frac{y}{x}\right)^{\sum b_i} = \left(\frac{y}{x}\right)^{\sum M_i(\boldsymbol{b})}.$$

If $x \neq y$, M preserves weight and we are done by the same argument as for when $xz \neq y$.

Otherwise if $x = y$, (9) becomes

$$x^{\sum a_i + \sum b_i - 2\sum a_i b_i} = x^{\sum M_i(\boldsymbol{a}) + \sum M_i(\boldsymbol{b}) - 2\sum M_i(\boldsymbol{a}) M_i(\boldsymbol{b})}.$$

This is equivalent to

$$x^{\frac{1}{2} - \frac{1}{2}\sum (2a_i - 1)(2b_i - 1)} = x^{\frac{1}{2} - \frac{1}{2}\sum (2M_i(\boldsymbol{a}) - 1)(2M_i(\boldsymbol{b}) - 1)},$$

which can be written as

$$x^{\sum a_i' b_i'} = x^{\sum M_i(\boldsymbol{a})' M_i(\boldsymbol{b})'}$$

where $a_i' = 2a_i - 1 \in \{-1, 1\}$, and similarly for b_i', and $M_i(\boldsymbol{a})'$, $M_i(\boldsymbol{b})'$. We can fix the same $\boldsymbol{a}$ and $\boldsymbol{b}$ as above, $\boldsymbol{a} = 000111$ and $\boldsymbol{b} = 010101$, and this gives $\boldsymbol{a}' = (-1, -1, -1, 1, 1, 1)$ and $\boldsymbol{b}' = (-1, 1, -1, 1, -1, 1)$. Then we get $2 = \sum_{i=1}^{6} a_i' b_i' \neq \sum_{i=1}^{6} M_i(\boldsymbol{a})' M_i(\boldsymbol{b})' = -2$. Thus we must have $x = 1$, and then $y = 1$ and $z = 1$, which contradicts $z \neq xy$. We have proved that if $z \neq xy$ the partition function $Z_{\mathcal{F}}(\cdot)$ is #P-hard.

APPENDIX: PROOF OF THEOREM 4

For completeness, in this appendix we give a proof of Theorem 4, which follows closely the proof in [14].

First, condition (A) is equivalent to Strong Rectangularity of Γ, the support constraint language of $\mathcal{F}$. To verify this condition one simply exhaustively searches for a mapping $D^3 \to D$ with the stated properties, called a Mal'tsev polymorphism of Γ; see [9, 22] for Mal'tsev polymorphisms and this equivalence.

For condition (B), again, searching for a possible mapping $\pi : D^6 \to D^6$ with the stated properties is clearly in NP. We now show that it is equivalent to the following:

> Primitive Balance Given Strong Rectangularity: Given D and $\mathcal{F}$ and assume the support constraint language of Γ is Strongly Rectangular. For every instance I of #CSP($\mathcal{F}$), the square $|D| \times |D|$ matrix M defined by I is block-rank-1; and this holds *if and only if* for all $\alpha \neq \beta \in D$ and $\kappa \neq \lambda \in D$,
>
> $$M(\alpha,\kappa)^2 M(\beta,\lambda)^2 M(\alpha,\lambda) M(\beta,\kappa) = M(\alpha,\lambda)^2 M(\beta,\kappa)^2 M(\alpha,\kappa) M(\beta,\lambda). \tag{10}$$
>
> Here, we assume the instance I has n variables and the rows and the columns of the matrix M are indexed by $x \in D$ and $y \in D$, and
>
> $$M(x, y) = \sum_{x_3, \ldots, x_n \in D} F_I(x, y, x_3, \ldots, x_n), \qquad \text{for all } x, y \in D. \tag{11}$$

We note that Equation (10) can be written as

$$M(\alpha,\kappa) M(\beta,\lambda) M(\alpha,\lambda) M(\beta,\kappa) \det \begin{bmatrix} M(\alpha,\kappa) & M(\alpha,\lambda) \\ M(\beta,\kappa) & M(\beta,\lambda) \end{bmatrix} = 0.$$

From this it is easy to see its equivalence to block-rank-1.

We next reformulate the property in terms of a new pair $(\mathfrak{D}, \mathfrak{F})$ which is called the *6-th power* of $(D, \mathcal{F})$.

(1) The new domain $\mathfrak{D} = D^6$, and we use $\mathfrak{s} = (s_1, \ldots, s_6)$ to denote an element in $\mathfrak{D}$, where $s_i \in D$.
(2) $\mathfrak{F} = \{g_1, \ldots, g_h\}$ has the same number of functions as $\mathcal{F}$ and every g_i, $i \in [h]$, has the same arity r_i as f_i. Function $g_i : \mathfrak{D}^{r_i} \to \mathbb{R}_+$ is constructed explicitly from f_i as follows:

$$g_i(\mathfrak{s}_1, \ldots, \mathfrak{s}_{r_i}) = \prod_{j \in [6]} f_i(s_{1,j}, \ldots, s_{r_i,j}), \qquad \text{for all } \mathfrak{s}_1, \ldots, \mathfrak{s}_{r_i} \in \mathfrak{D} = D^6.$$

An input instance I of $(D, \mathcal{F})$ over n variables $(x_1, \ldots, x_n)$ naturally defines an input instance $\mathfrak{I}$ of $(\mathfrak{D}, \mathfrak{F})$ over n variables $(y_1, \ldots, y_n)$ as follows: for each tuple $(f, i_1, \ldots, i_r) \in I$, add a tuple $(g, i_1, \ldots, i_r)$ to $\mathfrak{I}$, where $g \in \mathfrak{F}$ corresponds to $f \in \mathcal{F}$. Similarly, we let $G : \mathfrak{D}^n \to \mathbb{R}_+$ denote the n-ary function that $\mathfrak{I}$ defines:

$$G(y_1, \ldots, y_n) = \prod_{(g, i_1, \ldots, i_r) \in \mathfrak{I}} g(y_{i_1}, \ldots, y_{i_r}), \qquad \text{for all } y_1, \ldots, y_n \in \mathfrak{D}.$$

Let $\mathfrak{a}, \mathfrak{b}, \mathfrak{c}$ denote the following three specific elements from $\mathfrak{D}$:

$$\mathfrak{a} = (\alpha, \alpha, \alpha, \beta, \beta, \beta), \quad \mathfrak{b} = (\kappa, \lambda, \kappa, \lambda, \kappa, \lambda), \quad \mathfrak{c} = (\lambda, \kappa, \lambda, \kappa, \lambda, \kappa).$$

Since $\alpha \neq \beta$ and $\kappa \neq \lambda$, $\mathfrak{a}, \mathfrak{b}, \mathfrak{c}$ are three distinct elements in $\mathfrak{D}$. For each $\mathfrak{s} \in \mathfrak{D}$, let

$$\text{hom}_{\mathfrak{s}}(\mathfrak{I}) \overset{\text{def}}{=} \sum_{y_3, \ldots, y_n \in \mathfrak{D}} G(\mathfrak{a}, \mathfrak{s}, y_3, \ldots, y_n), \qquad \text{for every instance } \mathfrak{I} \text{ of } (\mathfrak{D}, \mathfrak{F}).$$

Let $\mathfrak{I}$ be the instance of $(\mathfrak{D}, \mathfrak{F})$ that corresponds to I, and let M be the $|D| \times |D|$ matrix defined in (11). Then, these choices of $\mathfrak{a}$, $\mathfrak{b}$, and $\mathfrak{c}$ are such that

$$\text{hom}_{\mathfrak{b}}(\mathfrak{I}) = M(\alpha, \kappa)^2 M(\beta, \lambda)^2 M(\alpha, \lambda) M(\beta, \kappa) \quad \text{and}$$
$$\text{hom}_{\mathfrak{c}}(\mathfrak{I}) = M(\alpha, \lambda)^2 M(\beta, \kappa)^2 M(\alpha, \kappa) M(\beta, \lambda),$$

after multiplying out the expressions and reassembling the sums. As a result, we have the following reformulation of the decision problem:

$$M \text{ satisfies (10) for all } I \iff \text{hom}_{\mathfrak{b}}(\mathfrak{I}) = \text{hom}_{\mathfrak{c}}(\mathfrak{I}) \text{ for all } \mathfrak{I}$$

The next reformulation considers sums over *injective* tuples only. We say $(y_1, \ldots, y_n) \in \mathfrak{D}^n$ is an injective tuple if $y_i \neq y_j$ for all $i \neq j \in [n]$. We use Y_n to denote the set of injective n-tuples. We now define functions $\text{mon}_{\mathfrak{s}}(\mathfrak{I})$, which are sums over injective tuples: For each $\mathfrak{s} \in \mathfrak{D}$, let

$$\text{mon}_{\mathfrak{s}}(\mathfrak{I}) \overset{\text{def}}{=} \sum_{(\mathfrak{a}, \mathfrak{s}, y_3, \ldots, y_n) \in Y_n} G(\mathfrak{a}, \mathfrak{s}, y_3, \ldots, y_n), \qquad \text{for every instance } \mathfrak{I} \text{ of } (\mathfrak{D}, \mathfrak{F}).$$

The following lemma is from [22] [Lemma 41] which shows that $\text{hom}_{\mathfrak{b}}(\mathfrak{I}) = \text{hom}_{\mathfrak{c}}(\mathfrak{I})$ for all $\mathfrak{I}$ if and only if the same equation holds for the sums over injective tuples. The proof uses the Möbius inversion.

Lemma 2 ([22], Lemma 41). $\text{hom}_{\mathfrak{b}}(\mathfrak{I}) = \text{hom}_{\mathfrak{c}}(\mathfrak{I})$ *for all* $\mathfrak{I}$ *if and only if* $\text{mon}_{\mathfrak{b}}(\mathfrak{I}) = \text{mon}_{\mathfrak{c}}(\mathfrak{I})$ *for all* $\mathfrak{I}$.

The condition of the following lemma can be verified in NP, and exactly the same proof for Lemma 6.6 in [14] works here (although the definitions of $\mathfrak{b}$ and $\mathfrak{c}$ are different. [5])

[5] There was an unfortunate typo in the statement of Lemma 6.6 in [14], where $\pi(\mathfrak{a}) = \mathfrak{a}$, $\pi(\mathfrak{b}) = \mathfrak{c}$ was mistakenly stated as $\pi(\mathfrak{a}) = \pi(\mathfrak{a})$, $\pi(\mathfrak{b}) = \pi(\mathfrak{c})$.

Lemma 3. $\text{mon}_{\mathfrak{b}}(\mathfrak{I}) = \text{mon}_{\mathfrak{c}}(\mathfrak{I})$ *for all* $\mathfrak{I}$ *if and only if there exists a bijection* π *from the domain* $\mathfrak{D}$ *to itself* (*called an* automorphism *of* $(\mathfrak{D}, \mathfrak{F})$) *such that* $\pi(\mathfrak{a}) = \mathfrak{a}$, $\pi(\mathfrak{b}) = \mathfrak{c}$, *and for every* r*-ary function* $g \in \mathfrak{F}$, *we have*

$$g(y_1, \ldots, y_r) = g\left(\pi(y_1), \ldots, \pi(y_r)\right), \qquad \textit{for all } y_1, \ldots, y_r \in \mathfrak{D}. \tag{12}$$

This completes the proof of Theorem 4.

ACKNOWLEDGMENTS

We thank the Editor and the three anonymous referees for their careful reading of the initial submission and valuable comments, all of which have helped us improve the presentation.

REFERENCES

[1] Carlo Baldassi, C. Borgs, J. Chayes, A. Ingrosso, Carlo Lucibello, Luca Saglietti, and R. Zecchina. 2016. Unreasonable effectiveness of learning neural networks: From accessible states and robust ensembles to basic algorithmic schemes. *Proceedings of the National Academy of Sciences* 113 (2016), E7655–E7662.

[2] Alexander Barvinok. 2016. *Combinatorics and Complexity of Partition Functions.* Vol. 30. Springer. https://doi.org/10.1007/978-3-319-51829-9

[3] Rodney J. Baxter. 2016. *Exactly Solved Models in Statistical Mechanics.* Elsevier.

[4] Christian Borgs, Jennifer Chayes, László Lovász, Vera T. Sós, and Katalin Vesztergombi. 2006. Counting graph homomorphisms. In *Topics in Discrete Mathematics.* Springer, 315–371.

[5] A. Braunstein, M. Mézard, and R. Zecchina. 2005. Survey propagation: An algorithm for satisfiability. *Random Struct. Algorithms* 27 (2005), 201–226.

[6] Graham R. Brightwell and Peter Winkler. 1999. Graph homomorphisms and phase transitions. *Journal of Combinatorial Theory, Series B* 77, 2 (1999), 221–262. https://doi.org/10.1006/jctb.1999.1899

[7] Andrei Bulatov, Martin Dyer, Leslie Ann Goldberg, Markus Jalsenius, Mark Jerrum, and David Richerby. 2012. The complexity of weighted and unweighted #CSP. *J. Comput. System Sci.* 78, 2 (2012), 681–688. https://doi.org/10.1016/j.jcss.2011.12.002 Games in Verification.

[8] Andrei Bulatov and Martin Grohe. 2005. The complexity of partition functions. *Theoretical Computer Science* 348, 2 (2005), 148–186. https://doi.org/10.1016/j.tcs.2005.09.011 *Automata, Languages and Programming: Algorithms and Complexity* (ICALP-A 2004).

[9] Andrei A. Bulatov. 2008. Also in J. ACM 60.5 (October 2013). The complexity of the counting constraint satisfaction problem. In *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008 (Lecture Notes in Computer Science, Vol. 5125).* Springer, 646–661. https://doi.org/10.1007/978-3-540-70575-8_53

[10] Andrei A. Bulatov and Víctor Dalmau. 2007. Towards a dichotomy theorem for the counting constraint satisfaction problem. *Information and Computation* 205, 5 (2007), 651–678. https://doi.org/10.1016/j.ic.2006.09.005

[11] Robert Burton and Jeffrey E. Steif. 1994. Non–uniqueness of measures of maximal entropy for subshifts of finite type. *Ergodic Theory and Dynamical Systems* 14, 2 (1994), 213–235. https://doi.org/10.1017/S0143385700007859

[12] Jin-Yi Cai and Xi Chen. 2010. Also in Comput. Complex. 28.3 (2019). A decidable dichotomy theorem on directed graph homomorphisms with non-negative weights. In *51st Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA.* IEEE Computer Society, 437–446. https://doi.org/10.1109/FOCS.2010.49

[13] Jin-Yi Cai and Xi Chen. 2012. Also in J. ACM 64.3 (2017). Complexity of counting CSP with complex weights. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, Howard J. Karloff and Toniann Pitassi (Eds.). ACM, 909–920. https://doi.org/10.1145/2213977.2214059

[14] Jin-Yi Cai, Xi Chen, and Pinyan Lu. 2011. Also in SIAM J. Comput. 45.6 (2016). Non-negatively weighted #CSP: An effective complexity dichotomy. In *Proceedings of the 26th Annual IEEE Conference on Computational Complexity, CCC 2011, San Jose, California, USA, June 8-10, 2011.* IEEE Computer Society, 45–54. https://doi.org/10.1109/CCC.2011.32

[15] Jin-Yi Cai and Artem Govorov. 2020. Dichotomy for graph homomorphisms with complex values on bounded degree graphs. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, Sandy Irani (Ed.). IEEE, 1103–1111. https://doi.org/10.1109/FOCS46700.2020.00106

[16] Jin-Yi Cai, Pinyan Lu, and Mingji Xia. 2014. The complexity of complex weighted boolean #CSP. *J. Comput. Syst. Sci.* 80, 1 (2014), 217–236. https://doi.org/10.1016/j.jcss.2013.07.003

[17] Jin-Yi Cai and Xi Chen. 2017. *Complexity Dichotomies for Counting Problems: Volume 1, Boolean Domain.* Cambridge University Press.

[18] Nadia Creignou and Miki Hermann. 1996. Complexity of generalized satisfiability counting problems. *Information and Computation* 125, 1 (1996), 1–12.

[19] Nadia Creignou, Sanjeev Khanna, and Madhu Sudan. 2001. *Complexity Classifications of Boolean Constraint Satisfaction Problems*. Society for Industrial and Applied Mathematics. https://doi.org/10.1137/1.9780898718546

[20] Irit Dinur, Elchanan Mossel, and Oded Regev. 2009. Conditional hardness for approximate coloring. *SIAM J. Comput.* 39, 3 (2009), 843–873. https://doi.org/10.1137/07068062X

[21] Martin Dyer and Catherine Greenhill. 2000. The complexity of counting graph homomorphisms. *Random Structures & Algorithms* 17, 3-4 (2000), 260–289. https://doi.org/10.1002/1098-2418(200010/12)17:3/4<260::AID-RSA5>3.0.CO;2-W

[22] Martin Dyer and David Richerby. 2013. An effective dichotomy for the counting constraint satisfaction problem. *SIAM J. Comput.* 42, 3 (2013), 1245–1274. https://doi.org/10.1137/100811258

[23] Martin E. Dyer, Alan M. Frieze, and Mark Jerrum. 2002. On counting independent sets in sparse graphs. *SIAM J. Comput.* 31, 5 (2002), 1527–1541. https://doi.org/10.1137/S0097539701383844

[24] Martin E. Dyer, Leslie Ann Goldberg, and Mike Paterson. 2006. Also in J. ACM 54.6 (December 2007). On counting homomorphisms to directed acyclic graphs. In *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 4051)*. Springer, 38–49. https://doi.org/10.1007/11786986_5

[25] Leslie Ann Goldberg and Mark Jerrum. 2015. A complexity classification of spin systems with an external field. *Proceedings of the National Academy of Sciences* 112, 43 (2015), 13161–13166. https://doi.org/10.1073/pnas.1505664112

[26] Artem Govorov, Jin-Yi Cai, and Martin Dyer. 2023. A dichotomy for bounded degree graph homomorphisms with nonnegative weights. *J. Comput. System Sci.* 132 (2023), 1–15. https://doi.org/10.1016/j.jcss.2022.09.002

[27] P. Hell and J. Nesetril. 2004. *Graphs and Homomorphisms*. OUP Oxford. https://books.google.com/books?id=bJXWV-qK7kYC

[28] Johan Håstad. 2001. Some optimal inapproximability results. *J. ACM* 48, 4 (July 2001), 798–859. https://doi.org/10.1145/502090.502098

[29] A. Ihler and David A. McAllester. 2009. Particle belief propagation. In *AISTATS*.

[30] Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. 2007. Optimal inapproximability results for MAX-CUT and other 2-variable CSPs? *SIAM J. Comput.* 37, 1 (2007), 319–357. https://doi.org/10.1137/S0097539705447372

[31] J. L. Lebowitz and G. Gallavotti. 1971. Phase transitions in binary lattice gases. *J. Math. Phys.* 12, 7 (1971), 1129–1133.

[32] Liang Li, Pinyan Lu, and Yitong Yin. 2012. Approximate counting via correlation decay in spin systems. In *Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 922–940.

[33] Liang Li, Pinyan Lu, and Yitong Yin. 2013. Correlation decay up to uniqueness in spin systems. In *Proceedings of the Twenty-fourth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 67–84.

[34] Jiabao Lin and Hanpin Wang. 2018. The complexity of Boolean Holant problems with nonnegative weights. *SIAM J. Comput.* 47, 3 (2018), 798–828. https://doi.org/10.1137/17M113304X

[35] László Lovász. 1967. Operations with structures. *Acta Mathematica Academiae Scientiarum Hungarica* 18, 3-4 (1967), 321–328.

[36] László Lovász. 2012. *Large Networks and Graph Limits*. Colloquium Publications, Vol. 60. American Mathematical Society. I–XIV, 1–475 pages.

[37] Pinyan Lu and Yitong Yin. 2016. Approximating the partition function of two–spin systems. In *Encyclopedia of Algorithms*. 117–123. https://doi.org/10.1007/978-1-4939-2864-4_750

[38] M. MacDonald and Mark S. Seidenberg. 2006. Constraint satisfaction accounts of lexical and sentence comprehension.

[39] Elitza Maneva, Elchanan Mossel, and Martin J. Wainwright. 2007. A new look at survey propagation and its generalizations. *J. ACM* 54, 4 (July 2007), 17–es. https://doi.org/10.1145/1255443.1255445

[40] Nima Noorshams and M. Wainwright. 2013. Belief propagation for continuous state spaces: Stochastic message-passing with quantitative guarantees. *J. Mach. Learn. Res.* 14 (2013), 2799–2835.

[41] Prasad Raghavendra. 2008. Optimal algorithms and inapproximability results for every CSP?. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing* (Victoria, British Columbia, Canada) *(STOC '08)*. Association for Computing Machinery, New York, NY, USA, 245–254. https://doi.org/10.1145/1374376.1374414

[42] P. Raghavendra and D. Steurer. 2009. How to round any CSP. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*. 586–594.

[43] Alistair Sinclair, Piyush Srivastava, and Marc Thurley. 2014. Approximation algorithms for two–state anti–ferromagnetic spin systems on bounded degree graphs. *Journal of Statistical Physics* 155, 4 (2014), 666–686.

[44] Alistair Sinclair, Piyush Srivastava, and Yitong Yin. 2013. Spatial mixing and approximation algorithms for graphs with bounded connective constant. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*. IEEE, 300–309.

[45] Allan Sly and Nike Sun. 2012. The computational hardness of counting in two-spin models on d-regular graphs. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*. IEEE, 361–369.

[46] L. Song, A. Gretton, D. Bickson, Y. Low, and Carlos Guestrin. 2011. Kernel belief propagation. In *AISTATS*.

[47] Mauricio Toro, C. Rueda, Carlos Agón, and G. Assayag. 2016. GELISP: A framework to represent musical constraint satisfaction problems and search strategies. *Journal of Theoretical and Applied Information Technology* 86 (2016), 327–331.

[48] M. Wainwright, T. Jaakkola, and A. Willsky. 2005. MAP estimation via agreement on trees: Message-passing and linear programming. *IEEE Transactions on Information Theory* 51 (2005), 3697–3717.

[49] Dror Weitz. 2006. Counting independent sets up to the tree threshold. In *Proceedings of the Thirty-eighth Annual ACM Symposium on Theory of Computing*. 140–149.

[50] Mingji Xia. 2010. Holographic reduction: A domain changed application and its partial converse theorems. In *Automata, Languages and Programming, 37th International Colloquium, ICALP 2010, Bordeaux, France, July 6-10, 2010, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 6198)*. Springer, 666–677. https://doi.org/10.1007/978-3-642-14165-2_56

[51] Jonathan S. Yedidia, W. Freeman, and Yair Weiss. 2000. Generalized belief propagation. In *NIPS*.

[52] Hai-Jun Zhou. 2013. Spin glass approach to the feedback vertex set problem. *The European Physical Journal B* 86 (2013), 1–9.

Received 5 March 2023; revised 1 November 2023; accepted 6 November 2023