On the Complexity of $\#\text{CSP}^d$

Jiabao Lin
Shanghai University of Finance and Economics, China
jiabaolincs@gmail.com

Abstract

Counting CSP^d is the counting constraint satisfaction problem (#CSP in short) restricted to the instances where every variable occurs a multiple of d times. This paper revisits tractable structures in #CSP and gives a complexity classification theorem for $\#\text{CSP}^d$ with algebraic complex weights. The result unifies affine functions (stabilizer states in quantum information theory) and related variants such as the local affine functions, the discovery of which leads to all the recent progress on the complexity of Holant problems.

The Holant is a framework that generalizes counting CSP. In the literature on Holant problems, weighted constraints are often expressed as tensors (vectors) such that projections and linear transformations help analyze the structure. This paper gives an example showing that different classes of tensors distinguished by these algebraic operations may share the same closure property under tensor product and contraction.

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness

Keywords and phrases Constraint satisfaction problem, counting problems, Holant, complexity dichotomy

Digital Object Identifier 10.4230/LIPIcs.ITCS.2021.40

Funding *Jiabao Lin*: This work is supported by Science and Technology Innovation 2030 –"New Generation of Artificial Intelligence" Major Project No.(2018AAA0100903), NSFC grant 61932002, Program for Innovative Research Team of Shanghai University of Finance and Economics (IRTSHUFE) and the Fundamental Research Funds for the Central Universities.

Acknowledgements I would like to thank anonymous referees for their insightful comments and suggestions.

1 Introduction

In the constraint satisfaction problem (CSP), constraints are specified by relations on a finite domain $D = \{0, 1, ..., q-1\}$ with $q \geq 2$. A relation $R \subseteq D^n$ can be seen as a function $f_R : D^n \to \{0, 1\}$ where $f_R(\mathbf{x}) = 1$ if and only if $\mathbf{x} \in R$. To express weighted constraints, we replace relations with complex-valued functions. Let $\mathbb{C}$ denote the set of algebraic complex numbers. Throughout this paper, we refer to them simply as complex numbers. Let $\mathcal{F} = \{f_1, ..., f_l\}$ be a finite function set where $f_i : D^{n_i} \to \mathbb{C}$ with arity $n_i > 0$. Then the weighted counting CSP specified by the set $\mathcal{F}$, denoted by $\#\text{CSP}(\mathcal{F})$, is defined as follows. An input instance I of the problem consists of

- A finite set of variables $V = \{x_1, ..., x_n\}$;
- A finite set of constraints $\{(F_1, \mathbf{x}_1), ..., (F_m, \mathbf{x}_m)\}$ where $F_i \in \mathcal{F}$ and $\mathbf{x}_i \in V^{\text{arity}(F_i)}$ is a tuple of (not necessarily distinct) variables.

Following [4], we say that the instance I *defines* a function of arity n:

$$F_I(x_1, ..., x_n) = \prod_{i=1}^{m} F_i(\mathbf{x}_i).$$

12th Innovations in Theoretical Computer Science Conference (ITCS 2021).
Editor: James R. Lee; Article No. 40; pp. 40:1–40:10
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The output of the problem $\#\mathrm{CSP}(\mathcal{F})$ is the following sum (also called the *partition function*):

$$Z(I) = \sum_{\mathbf{x} \in D^n} F_I(\mathbf{x}).$$

The problem is the counting version of a classical CSP, if we restrict $\mathcal{F}$ to functions with range $\{0, 1\}$. Weighted constraints make the #CSP framework more expressive. For a binary function $f : D^2 \to \mathbb{C}$, $\#\mathrm{CSP}(f)$ is the problem of counting graph homomorphisms into the graph on D with edge weights $f(i, j)$. A wide range of graph parameters can be encoded by graph homomorphisms (see, e.g. [22]), which also play an important role in statistical physics.

The complexity of counting CSP has been intensively studied over the last two decades. Bulatov [3] first gave a complexity dichotomy theorem for unweighted #CSPs: Each problem is either solvable in polynomial time or proved to be #P-hard. Understanding the proof requires knowledge of universal algebra. Later, Dyer and Richerby [18] found a new tractability criterion and their proof is elementary. Based on the techniques developed for unweighted #CSP, the dichotomy was generalized to cover nonnegative [6] and complex weights [4]. Given a function $F : D^n \to \mathbb{C}$, we use $F^{[t]}$, for each $t \in [n] = \{1, ..., n\}$, to denote the following function of arity t:

$$F^{[t]}(x_1, ..., x_t) = \sum_{x_{t+1}, ..., x_n} F(x_1, ..., x_t, x_{t+1}, ..., x_n).$$

And for a function set $\mathcal{F}$, we define the set

$$\mathcal{W}_{\mathcal{F}} = \{F^{[t]} \mid F \text{ is a function defined by an instance of } \#\mathrm{CSP}(\mathcal{F}) \text{ and } 1 \le t \le \text{arity of } F\}.$$

The dichotomy theorem for complex-weighted #CSP is stated as follows.

▶ **Theorem 1** ([4]). *Let $\mathcal{F}$ be a finite set of complex-valued functions. The problem $\#\mathrm{CSP}(\mathcal{F})$ is solvable in polynomial time if the set $\mathcal{W}_{\mathcal{F}}$ satisfies three conditions: the Block Orthogonality condition, the Type Partition condition, and the Mal'tsev condition. Otherwise $\#\mathrm{CSP}(\mathcal{F})$ is #P-hard.*

Roughly speaking, the three conditions require the function defined by any instance to be well-structured, such that the sum of function values can be computed by an efficient algorithm instead of brute-force enumeration. Actually, even if a problem $\#\mathrm{CSP}(\mathcal{F})$ is #P-hard, it is still possible that the algorithm succeeds on a nontrivial subset of instances. In this paper, we consider a special case denoted by $\#\mathrm{CSP}^d$, which was first studied by Huang and Lu [20].

▶ **Definition 2.** *Let $d \ge 1$ be an integer and let $\mathcal{F}$ be a set of complex-valued functions. The problem $\#\mathrm{CSP}^d(\mathcal{F})$ is the restriction of $\#\mathrm{CSP}(\mathcal{F})$ to the instances where every variable occurs a multiple of d times.*

By definition, if $d = 1$, then the problem $\#\mathrm{CSP}^d(\mathcal{F})$ is exactly $\#\mathrm{CSP}(\mathcal{F})$. For a function set $\mathcal{F}$, $\#\mathrm{CSP}^d(\mathcal{F})$ is a subproblem of $\#\mathrm{CSP}(\mathcal{F})$. We consider a subset of $\mathcal{W}_{\mathcal{F}}$:

$$\mathcal{W}^d_{\mathcal{F}} = \{F^{[t]} \mid F \text{ is a function defined by an instance of } \#\mathrm{CSP}^d(\mathcal{F}) \text{ and } 1 \le t \le \text{arity of } F\}.$$

A slight modification of the proof of Theorem 1 yields a unified dichotomy theorem for the $\#\mathrm{CSP}^d$ family.

▶ **Theorem 3.** *Let $d \geq 1$ be an integer and let $\mathcal{F}$ be a finite set of complex-valued functions. The problem $\#\mathrm{CSP}^d(\mathcal{F})$ is solvable in polynomial time if the set $\mathcal{W}^d_{\mathcal{F}}$ satisfies three conditions: the Block Orthogonality condition, the Type Partition condition, and the Mal'tsev condition. Otherwise $\#\mathrm{CSP}^d(\mathcal{F})$ is $\#\mathrm{P}$-hard.*

The tractability criteria survive because, as mentioned before, they are imposed on definable functions and hence not sensitive to how many times a variable appears in an instance. Unfortunately, none of the three conditions is known to be decidable. It is desirable to derive more explicit criteria for constraint functions instead of the functions "generated" by them. However, closed-form formulas or a succinct description of the function values might not exist for arbitrary domains. In graph homomorphisms with real or complex weights [19, 5], the classification of binary functions is explicit but very complicated.

By the definition of #CSP, a variable can occurs arbitrarily many times in an instance. However, in many graph satisfaction problems like matchings, vertices are viewed as constraints and edges as variables. That is, each variable appears exactly twice. Inspired by holographic algorithms [24], Cai, Lu, and Xia [13] proposed the Holant framework.

▶ **Definition 4.** *The problem* $\mathrm{Holant}(\mathcal{F})$ *is the restriction of* $\#\mathrm{CSP}(\mathcal{F})$ *to the instances where every variable occurs exactly twice.*

On the one hand, the Holant is more expressive than #CSP because any $\#\mathrm{CSP}(\mathcal{F})$ is polynomial-time equivalent to the problem $\mathrm{Holant}(\mathcal{F} \cup \{\mathrm{EQ}\})$ where $\mathrm{EQ}(x_1, x_2, x_3) = 1$ if $x_1 = x_2 = x_3$ and otherwise $\mathrm{EQ}(x_1, x_2, x_3) = 0$. One the other hand, by definition, $\mathrm{Holant}(\mathcal{F})$ is a subproblem of $\#\mathrm{CSP}(\mathcal{F})$. The partition function of a bipartite Holant instance is invariant under the operations of the linear group $GL_q(\mathbb{C})$ on constraint functions. These operations are also called *holographic transformations* [24, 13], which turn out to be one of the new sources of tractability [10, 21, 2, 23].

Early study of Holant problems has a similar flavor to that of #CSP (see, e.g. [17, 13, 12]). Based on the dichotomy for a special family of Holant problems, Cai, Lu, and Xia [15] gave an explicit criterion for complex-weighted #CSP on the Boolean domain $\{0, 1\}$ (Boolean #CSP in short).

▶ **Theorem 5** ([15])**.** *Let $\mathcal{F}$ be a set of complex-valued functions on the Boolean domain. Then the problem $\#\mathrm{CSP}(\mathcal{F})$ is solvable in polynomial time if $\mathcal{F} \subseteq \mathcal{P}$ or $\mathcal{F} \subseteq \mathcal{A}$. Otherwise $\#\mathrm{CSP}(\mathcal{F})$ is $\#\mathrm{P}$-hard.*

The definitions of product-type functions $\mathcal{P}$ and affine functions $\mathcal{A}$ are given in Section 3. Affine functions are discovered in the first paper on Holant problems [13], and recently these functions are shown to be equivalent to Clifford gates and stabilizer circuits in quantum computing [11, 1].

Significant progress has been made towards a complexity dichotomy for complex-weighted Holant [7, 21, 2, 23, 14, 9]. However, it remains open even on the Boolean domain. Since it was first proposed, the $\#\mathrm{CSP}^d$ family becomes increasingly important in understanding the relationship between Holant and #CSP. Cai, Lu, and Xia [16] proved a dichotomy theorem for Boolean $\#\mathrm{CSP}^2$ and discovered a new tractable class called *local affine* functions (see Definition 15). This result laid the foundation for all the recent progress on real or complex Holant [2, 8, 23]. In fact, several major Holant dichotomies inevitably go through the $\#\mathrm{CSP}^d$ family whose complexity, before Theorem 3, is known only for some special cases [20, 16, 8]. Moreover, the proofs of these results on $\#\mathrm{CSP}^d$ are complicated, partially because explicit criteria are expected as in Theorem 5.

Although Theorem 3 shows a complexity dichotomy, it is too general to say much about the Boolean domain. Of course we can check the three conditions and derive a simplified version, but there is a direct generalization of Theorem 5. Both product-type and affine functions have a nice closure property: $\mathcal{W}_{\mathcal{P}} \subseteq \mathcal{P}$ and $\mathcal{W}_{\mathcal{A}} \subseteq \mathcal{A}$. This fact and inspiration from Theorem 3 lead to the following theorem.

▶ **Theorem 6.** *Let $d \geq 1$ be an integer and let $\mathcal{F}$ be a set of complex-valued functions on the Boolean domain. Then the problem $\#\mathrm{CSP}^d(\mathcal{F})$ is solvable in polynomial time if $\mathcal{W}^d_{\mathcal{F}} \subseteq \mathcal{P}$ or $\mathcal{W}^d_{\mathcal{F}} \subseteq \mathcal{A}$. Otherwise $\#\mathrm{CSP}^d(\mathcal{F})$ is #P-hard.*

The remainder of this paper is organized as follows. In Section 2, we give a proof of Theorem 3. In Section 3, a simple proof of Theorem 6 is presented. This theorem looks very different from the previous results. For example, holographic transformations seem necessary for the definition of the local affine functions and the algorithm that efficiently solves the problem they define. It will be clear why these transformations disappear in Theorem 6. Some concluding remarks appear in Section 4.

2 On General Finite Domains

This section is devoted to the proof Theorem 3. Most of the work was done in [4] and we only show necessary modifications.

Throughout this section, we assume that functions and relations are defined on a fixed finite domain $D = \{0, 1, ..., q-1\}$. And we use $\mathcal{F}$ to denote a finite set of functions on D. Let $\leq_\mathsf{T}$ denote the polynomial-time Turing reductions.

▶ **Lemma 7.** *For any finite set $\mathcal{G} \subset \mathcal{W}^d_{\mathcal{F}}$, $\#\mathrm{CSP}(\mathcal{G}) \leq_\mathsf{T} \#\mathrm{CSP}^d(\mathcal{F})$.*

Proof. For any function $f \in \mathcal{G} \subset \mathcal{W}^d_{\mathcal{F}}$, there exists some instance I_f of the problem $\#\mathrm{CSP}^d(\mathcal{F})$ such that f is exactly the function defined by I_f. Note that I_f is of constant size, since $\mathcal{G}$ is a finite set.

Let I_1 be an instance of the problem $\#\mathrm{CSP}(\mathcal{G})$. We replace each constraint $(f, \mathbf{x}) \in I_1$ with the instance I_f (in variables $\mathbf{x}$). Then we get a new instance I_2 of the problem $\#\mathrm{CSP}^d(\mathcal{F})$ because the sum of multiples of an integer d is still a multiple of d. It is easy to verify that $Z(I_1) = Z(I_2)$. The size of I_2 is polynomial in that of I_1. ◀

The readers can find the statements of the conditions of Theorem 1 in [4, Subsection 3.1]. We say that the function set $\mathcal{W}^d_{\mathcal{F}}$ violates any of the three conditions, if there exists a finite set $\mathcal{G} \subset \mathcal{W}^d_{\mathcal{F}}$ violates any of the three. The hardness part of Theorem 1 can be summarized as follows.

▶ **Lemma 8** (Lemmas 3.2, 3.4, 3.5 in [4])**.** *If a finite function set $\mathcal{G}$ violates one of the three conditions in Theorem 1, then the problem $\#\mathrm{CSP}(\mathcal{G})$ is #P-hard.*

Hardness Part of Theorem 3. Suppose that a finite set $\mathcal{G} \subset \mathcal{W}^d_{\mathcal{F}}$ violates any of the three conditions. Then the problem $\#\mathrm{CSP}(\mathcal{G})$ is #P-hard by Lemma 8. Moreover, $\#\mathrm{CSP}^d(\mathcal{F})$ is #P-hard because $\#\mathrm{CSP}(\mathcal{G}) \leq_\mathsf{T} \#\mathrm{CSP}^d(\mathcal{F})$. ◀

Now we consider the algorithmic part.

Consider a relation $R \subseteq D^n$ and a map $\varphi : D^3 \to D$. We say the relation R is *closed* under the map φ, if for any three tuples $\mathbf{u} = (u_i), \mathbf{v} = (v_i), \mathbf{w} = (w_i) \in R$, it holds that

$$(\varphi(u_1, v_1, w_1), \varphi(u_2, v_2, w_2), ..., \varphi(u_n, v_n, w_n)) \in R.$$

In this case, we also say φ is a *polymorphism* of the relation R.

▶ **Definition 9** (Mal'tsev Polymorphism). *Suppose that a relation $R \subseteq D^n$ is closed under a map $\varphi : D^3 \to D$. We say that φ is a* Mal'tsev polymorphism *of R, if*

$$\varphi(a, b, b) = \varphi(b, b, a) = a$$

for all $a, b \in D$.

Given a function f, we use R_f denote the relation $\{\mathbf{x} \in D^n \mid f(\mathbf{x}) \neq 0\}$ (called the *support* of f). And for every function $f \in \mathcal{W}^d_{\mathcal{F}}$ of arity $n \geq 2$, a relation $\Omega_f \subseteq D^{2n-2}$ is defined as:

$$(\mathbf{x}, \mathbf{y}) \in \Omega_f \iff f(\mathbf{x}, *) \text{ and } f(\mathbf{y}, *) \text{ are both nonzero and linearly dependent,}$$

where $f(\mathbf{x}, *)$ denotes the q-dimensional vector $(f(\mathbf{x}, 0), f(\mathbf{x}, 1), ..., f(\mathbf{x}, q-1))$. Now we define the set

$$\Lambda^d_{\mathcal{F}} = \{R_f \mid f \in \mathcal{W}^d_{\mathcal{F}}\} \cup \{\Omega_f \mid f \in \mathcal{W}^d_{\mathcal{F}} \text{ of arity} \geq 2\}.$$

▶ **Definition 10** (The Mal'tsev Condition). *All the relations in $\Lambda^d_{\mathcal{F}}$ have a common Mal'tsev polymorphism.*

We have the following lemma after checking the algorithm (say, denoted by $\mathcal{A}$) for Theorem 1.

▶ **Lemma 11.** *Suppose that the set $\mathcal{W}^d_{\mathcal{F}}$ satisfies the three conditions. Then the algorithm $\mathcal{A}$ can solve the problem $\#\mathrm{CSP}^d(\mathcal{F})$ in polynomial time if all the relations in $\Lambda^d_{\mathcal{F}} \cup \{R_f \mid f \in \mathcal{F}\}$ have a common Mal'tsev polymorphism.*

In fact, the Mal'tsev condition already implies the condition in Lemma 11. This completes the algorithmic part of Theorem 3.

▶ **Lemma 12.** *Suppose that the Mal'tsev condition holds. Then all the relations in $\Lambda^d_{\mathcal{F}} \cup \{R_f \mid f \in \mathcal{F}\}$ have a common Mal'tsev polymorphism.*

Proof. The conclusion is trivial if $d = 1$, since $\mathcal{F} \subset \mathcal{W}_{\mathcal{F}}$.

For any function $f : D^n \to \mathbb{C}$, we consider the function $f^d(\mathbf{x}) = (f(\mathbf{x}))^d$ for all $\mathbf{x} \in D^n$. The two functions f and f^d have the same support: $R_f = R_{f^d}$. Then the conclusion follows because $f^d \in \mathcal{W}^d_{\mathcal{F}}$ for every function $f \in \mathcal{F}$. ◀

Some remarks that may help the readers. A relation $R \subseteq D^n$ with a Mal'tsev polymorphism can be of exponential size in n. However, Dyer and Richerby [18] showed that, there is a succinct representation of R determined by the Mal'tsev polymorphism, called the *witness function*, which has linear size in n. Here we do not introduce the definition of the witness function. Given an instance I, the algorithm $\mathcal{A}$ starts with a witness function of the support R_{F_I}. The algorithm for constructing witness functions, by Dyer and Richerby, works no matter how many times a variable occurs but only requires a Mal'tsev polymorphism shared by all the relations R_f for $f \in \mathcal{F}$. Lemma 11 covers the requirement, which is satisfied trivially when $d = 1$ since $\{R_f \mid f \in \mathcal{F}\} \subset \Lambda^1_{\mathcal{F}}$. Later, the instance I is only used for evaluating the function F_I at some points in D^n. To compute the sum $Z(I) = \sum_{\mathbf{x} \in D^n} F_I(\mathbf{x})$, the algorithm $\mathcal{A}$ produces a data structure, called the *row representation*, for each $F^{[t]}_I$ ($t \in [n]$). Now suppose that I is a $\#\mathrm{CSP}^d$ instance. By definition, $F^{[t]}_I \in \mathcal{W}^d_{\mathcal{F}}$ for all $t \in [n]$. To obtain the row representations, it is sufficient to impose the three conditions on $\mathcal{W}^d_{\mathcal{F}}$ under which all the functions and relations involved in the computation are well-structured.

3 On the Boolean Domain

In this section, all the functions and relations are defined on the Boolean domain $D = \{0, 1\}$.

We start with the definition of product-type functions and affine functions. Let $\mathrm{EQ}(x, y)$ denote the equality function: $\mathrm{EQ}(x, y) = 1$ if $x = y$, otherwise $\mathrm{EQ}(x, y) = 0$. And let $\mathrm{NE}(x, y)$ denote the disequality function: $\mathrm{NE}(x, y) = 1 - \mathrm{EQ}(x, y)$.

▶ **Definition 13** (Product-Type Functions). *A function is of* product type *if it is defined by a* #CSP *instance where every constraint function is a unary function or the binary function* EQ *or* NE*. Let* $\mathcal{P}$ *denote the set of all product-type functions.*

A Boolean relation is *affine* if it is the set of solutions to a system of linear equations over the field $\mathbb{Z}_2$. We say that f has affine support if its support is affine. Recall that the support of f is the relation $R_f = \{\mathbf{x} \in \{0, 1\}^n \mid f(\mathbf{x}) \neq 0\}$.

▶ **Definition 14** (Affine Functions). *A function f of arity n is* affine *if its support is affine and there is a constant $\lambda \in \mathbb{C}$ such that for all $\mathbf{x} \in R_f$,*

$$f(\mathbf{x}) = \lambda \cdot i^{Q(\mathbf{x})},$$

where $i = \sqrt{-1}$ and Q is a homogeneous quadratic polynomial

$$Q(x_1, ..., x_n) = \sum_{i=1}^{n} a_i x_i^2 + 2 \sum_{1 \leq i < j \leq n} b_{ij} x_i x_j$$

with $a_i \in \mathbb{Z}_4$ and $b_{ij} \in \{0, 1\}$. We use $\mathcal{A}$ to denote the set of all affine functions.

Proof of Theorem 6. Suppose that there are two functions $f, g \in \mathcal{W}_{\mathcal{F}}^d$ (they can be the same) such that $f \notin \mathcal{P}$ and $g \notin \mathcal{A}$. By Theorem 5, the problem $\#\mathrm{CSP}(\{f, g\})$ is #P-hard. Then $\#\mathrm{CSP}^d(\mathcal{F})$ is also #P-hard since $\#\mathrm{CSP}(\{f, g\}) \leq_{\mathrm{T}} \#\mathrm{CSP}^d(\mathcal{F})$ by Lemma 7.

Now we assume that $\mathcal{W}_{\mathcal{F}}^d \subseteq \mathcal{P}$ or $\mathcal{W}_{\mathcal{F}}^d \subseteq \mathcal{A}$. In both cases, for any instance I, the function F_I (say, of arity n) has affine support and the linear system for the support can be constructed efficiently from that of the constraint functions. There are two cases:

- $F_I \in \mathcal{P}$. We can determine the variable dependence on the support: $x_i = x_j$ or $x_i \neq x_j$ or they are independent. Then the evaluation of the partition function $Z(I)$ reduces to a trivial case where every constraint is unary.
- $F_I \in \mathcal{A}$. We can obtain the explicit formula (in Definition 14) for the function F_I, by evaluating it at $O(n^2)$ many points, using the instance I. Then the algorithm for Theorem 5 is able to compute $Z(I)$. See [15] for more details.

Therefore, the partition function is computable in polynomial time. ◀

The remainder of this section is devoted to the connection between Theorem 6 and the dichotomy for Boolean $\#\mathrm{CSP}^2$ in [16]. Before this, we need to introduce some notations and definitions.

A function of arity $n \geq 2$ can be expressed as a $2^{n-r} \times 2^r$ matrix ($0 \leq r \leq n$), denoted by $M_r(f)$. The rows and columns are indexed by $\mathbf{x} \in \{0, 1\}^{n-r}$ and $\mathbf{y} \in \{0, 1\}^r$, respectively, and $f(\mathbf{x}, \mathbf{y})$ is the $(\mathbf{x}, \mathbf{y})^{\mathrm{th}}$ entry of the matrix $M_r(f)$. In particular, when $r = 0$, $M_r(f)$ is a column vector of dimension 2^n. In the following, we do not distinguish a function from its matrix representations. The integer r will be clear in matrix multiplication.

Given two matrices A and B, we use $A \otimes B$ to denote their tensor product.

Let $\alpha = \frac{1+i}{\sqrt{2}}$ where $i = \sqrt{-1}$. Then $\alpha^2 = i$. And let $M_x = \left[\begin{smallmatrix} 1 & 0 \\ 0 & x \end{smallmatrix}\right]$ for $x \in \mathbb{C}$. We define the following set of functions:

$$\mathcal{A}^\alpha = \{M_\alpha^{\otimes n} g \,|\, g \in \mathcal{A} \text{ and } n \text{ is the arity of } g\},$$

which is a tractable class for $\#\text{CSP}^2$.

Recall that $R_f = \{\mathbf{x} \in \{0,1\}^n \,|\, f(\mathbf{x}) \neq 0\}$ for a function f of arity n.

▶ **Definition 15** (Local Affine Functions). *A function f (of arity n) is called* local affine, *if for every element $(s_1, s_2, ..., s_n) \in R_f$,*

$$(M_{\alpha^{s_1}} \otimes M_{\alpha^{s_2}} \otimes \cdots \otimes M_{\alpha^{s_n}}) f \in \mathcal{A},$$

where $\alpha^s = 1$ if $s = 0$, $\alpha^s = \alpha$ if $s = 1$. The set of all local affine functions is denoted by $\mathcal{L}$.

▶ **Theorem 16** ([16]). *Let $\mathcal{F}$ be a set of functions on the Boolean domain. If $\mathcal{F} \subseteq \mathcal{C}$ for $\mathcal{C} \in \{\mathcal{P}, \mathcal{A}, \mathcal{A}^\alpha, \mathcal{L}\}$, then the problem $\#\text{CSP}^2(\mathcal{F})$ is solvable in polynomial time. Otherwise $\#\text{CSP}^2(\mathcal{F})$ is $\#$P-hard.*

Since the theorem above is a special case of Theorem 6, the two tractability criteria should be compatible. In fact, we have the following observation.

▶ **Lemma 17.** *If $\mathcal{F} \subseteq \mathcal{C}$ for $\mathcal{C} \in \{\mathcal{P}, \mathcal{A}, \mathcal{A}^\alpha, \mathcal{L}\}$, then $\mathcal{W}^2_\mathcal{F} \subseteq \mathcal{P}$ or $\mathcal{W}^2_\mathcal{F} \subseteq \mathcal{A}$.*

The algorithms in [16] for the two classes $\mathcal{A}^\alpha$ and $\mathcal{L}$ start with local transformations induced by the matrix M_α, such that every constraint function of an instance I becomes affine and hence the function F_I is also affine. However, as stated in Lemma 17, F_I is already affine. Before proving the lemma, we need some preparations.

▶ **Lemma 18** (Closure Property). *Let $\mathcal{C} = \mathcal{P}$ or $\mathcal{C} = \mathcal{A}$, then $\mathcal{W}_\mathcal{F} \subseteq \mathcal{C}$ for any set $\mathcal{F} \subseteq \mathcal{C}$. In particular, for any n-ary function $f \in \mathcal{C}$:*

- *$f^{[t]} \in \mathcal{C}$ for all $t \in [n]$;*
- *$f_\pi \in \mathcal{C}$ for any permutation π on $[n]$, where $f_\pi(x_1, x_2, ..., x_n) = f(x_{\pi(1)}, x_{\pi(2)}, ..., x_{\pi(n)})$.*

Proof. We only show that $g(x_1, ..., x_{n-1}) = \sum_{x_n \in \{0,1\}} f(x_1, ..., x_{n-1}, x_n) \in \mathcal{C}$ for any n-ary function $f \in \mathcal{C}$. The case $\mathcal{C} = \mathcal{A}$ was proved in [11, Lemma 3.1]. Now suppose that $f \in \mathcal{P}$. By the definition of product-type functions, it is sufficient to consider the case where f has support

$$R_f \subseteq \{(u_1, ..., u_n), (1 - u_1, ..., 1 - u_n)\} \text{ for some } u_i \in \{0,1\}.$$

It then follows that $R_g \subseteq \{(u_1, ..., u_{n-1}), (1 - u_1, ..., 1 - u_{n-1})\}$. Thus $g \in \mathcal{P}$. ◀

▶ **Lemma 19** (Closure under Matrix Multiplication). *Let $f \in \mathcal{A}$ be a function of arity n. And let $g \in \mathcal{A}$ be a function of arity m. Then it holds that*

$$M_r(f) M_{m-r}(g) \in \mathcal{A},$$

for all $0 \le r \le \min\{n, m\}$.

Proof. For any $0 \le r \le \min\{n, m\}$, we consider the function of arity $n + m - 2r$:

$$\begin{aligned} & h(x_1, ..., x_{n-r}, y_1, ..., y_{m-r}) \\ = & \sum_{z_1, ..., z_r \in \{0,1\}} f(x_1, ..., x_{n-r}, z_1, ..., z_r) g(z_1, ..., z_r, y_1, ..., y_{m-r}). \end{aligned}$$

Then it follows that $M_{m-r}(h) = M_r(f) M_{m-r}(g)$. By Lemma 18, we have $h \in \mathcal{A}$. ◀

▶ **Lemma 20** (Closure under Tensor Product). *Let $A, B \in \mathcal{A}$ be two matrices. Then $A \otimes B \in \mathcal{A}$.*

Proof. Let $f, g \in \mathcal{A}$ be two functions such that $A = M_r(f)$ and $B = M_t(f)$ for some integers $r, t \geq 0$. Consider the function

$$h(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{w}) = f(\mathbf{x}, \mathbf{z}) g(\mathbf{y}, \mathbf{w})$$

where $\mathbf{z} \in \{0,1\}^r$ and $\mathbf{w} \in \{0,1\}^t$. Then $A \otimes B = M_{r+t}(h)$. ◀

▶ **Lemma 21.** *Let f be a function of arity n. Suppose that there exist matrices $A_1, A_2, ..., A_n \in \left\{ \left[\begin{smallmatrix} 1 & 0 \\ 0 & i^r \end{smallmatrix}\right] \mid r = 0,1,2,3 \right\}$ where $i = \sqrt{-1}$, such that $(A_1 \otimes A_2 \otimes \cdots \otimes A_n) f \in \mathcal{A}$. Then $f \in \mathcal{A}$.*

Proof. Set $A = A_1 \otimes A_2 \otimes \cdots \otimes A_n$. The matrix A is invertible and $A^{-1} = A_1^{-1} \otimes A_2^{-1} \otimes \cdots \otimes A_n^{-1} \in \mathcal{A}$ by Lemma 20. Then it follows that $f = (A^{-1}A)f = A^{-1}(Af) \in \mathcal{A}$, according to Lemma 19. ◀

Now we are ready to prove Lemma 17.

Proof of Lemma 17. Due to the closure property of product-type and affine functions (Lemma 18), the two cases $\mathcal{F} \subseteq \mathcal{A}^\alpha$ and $\mathcal{F} \subseteq \mathcal{L}$ remain to be verified. Furthermore, by Lemma 18, we only need to check the definable functions.

Let I be an instance of $\#\mathrm{CSP}^2(\mathcal{F})$. Suppose that the instance I has n variables $\{x_1, x_2, ..., x_n\}$ and m constraints $\{(f_1, \mathbf{x}_1), (f_2, \mathbf{x}_2), ..., (f_m, \mathbf{x}_m)\}$. Then the function defined by I is

$$F_I(x_1, ..., x_n) = f_1(\mathbf{x}_1) f_2(\mathbf{x}_2) \cdots f_m(\mathbf{x}_m).$$

Suppose that each variable x_j occurs k_j times in the instance I. By definition, k_j is even for each $j \in [n]$ and we set $k = k_1 + k_2 + \cdots + k_n$.

Consider the function of arity k: $g = f_1 \otimes f_2 \otimes \cdots \otimes f_m$. There is a permutation π on $[k]$ such that

$$F_I(x_1, ..., x_n) = g_\pi(\underbrace{x_1, ..., x_1}_{k_1 \text{ times}}, \underbrace{x_2, ..., x_2}_{k_2 \text{ times}}, ..., \underbrace{x_n, ..., x_n}_{k_n \text{ times}}).$$

Suppose that $\mathcal{F} \subseteq \mathcal{L}$. We show that $F_I \in \mathcal{A}$. If the function F_I is identically zero, then we are done. Suppose not. Then there exist $s_1, s_2, ..., s_n \in \{0,1\}$ such that

$$F_I(s_1, s_2, ..., s_n) = g_\pi(\underbrace{s_1, ..., s_1}_{k_1 \text{ times}}, \underbrace{s_2, ..., s_2}_{k_2 \text{ times}}, ..., \underbrace{s_n, ..., s_n}_{k_n \text{ times}}) \neq 0.$$

By the definition of local affine functions, we have

$$(M_{\alpha^{s_1}}^{\otimes k_1} \otimes M_{\alpha^{s_2}}^{\otimes k_2} \otimes \cdots \otimes M_{\alpha^{s_n}}^{\otimes k_n}) g_\pi \in \mathcal{A}. \quad (*)$$

The relation between the two functions F_I and g_π shows that

$$(M_{\alpha^{s_1}}^{k_1} \otimes M_{\alpha^{s_2}}^{k_2} \otimes \cdots \otimes M_{\alpha^{s_n}}^{k_n}) F_I \in \mathcal{A}.$$

For each $j \in [n]$, $M_{\alpha^{s_j}}^{k_j} \in \left\{ \left[\begin{smallmatrix} 1 & 0 \\ 0 & i^r \end{smallmatrix}\right] \mid r = 0,1,2,3 \right\}$, since k_j is even. By Lemma 21, we have $F_I \in \mathcal{A}$.

Now suppose that $\mathcal{F} \subseteq \mathcal{A}^\alpha$. However, this case has been considered, because the relation $(*)$ holds by setting $s_1 = s_2 = \cdots = s_n = 1$. This completes the proof. ◀

4 Conclusion

Local affine functions partially reflect the difficulty in proving a Holant dichotomy: Nice structures hide in strange supports and we lack powerful tools.

Theorem 3 and Theorem 6 give a unified complexity dichotomy for the whole $\#\mathrm{CSP}^d$ family. Being abstract enough, they reveal that essentially there is no new tractable structure for $d > 1$. This fact is obtained by considering barriers to efficient evaluations of the partition functions, but not simply the set of constraint functions which defines the problem though. Moreover, the proof of Theorem 6 is much simpler than those of the partial results on $\#\mathrm{CSP}^d$.

It is not clear whether or not the dichotomies in this paper can help the study of Holant problems at large. They are much more conceptual than existing dichotomies and techniques for the Holant.

References

1 Miriam Backens. A new holant dichotomy inspired by quantum computation. In *44th International Colloquium on Automata, Languages,and Programming, ICALP 2017*, pages 16:1–16:14, 2017. doi:10.4230/LIPIcs.ICALP.2017.16.

2 Miriam Backens. A complete dichotomy for complex-valued Holant^c. In *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, pages 12:1–12:14, 2018. doi:10.4230/LIPIcs.ICALP.2018.12.

3 Andrei A. Bulatov. The complexity of the counting constraint satisfaction problem. *J. ACM*, 60(5):34, 2013. doi:10.1145/2528400.

4 Jin-Yi Cai and Xi Chen. Complexity of counting CSP with complex weights. *J. ACM*, 64(3):19:1–19:39, 2017. doi:10.1145/2822891.

5 Jin-Yi Cai, Xi Chen, and Pinyan Lu. Graph homomorphisms with complex values: A dichotomy theorem. *SIAM J. Comput.*, 42(3):934–1029, 2013. doi:10.1137/110840194.

6 Jin-Yi Cai, Xi Chen, and Pinyan Lu. Nonnegative weighted #CSP: An effective complexity dichotomy. *SIAM J. Comput.*, 45(6):2177–2198, 2016. doi:10.1137/15m1032314.

7 Jin-Yi Cai, Zhiguo Fu, Heng Guo, and Tyson Williams. A Holant dichotomy: Is the FKT algorithm universal? In *IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 1259–1276, 2015. doi:10.1109/focs.2015.81.

8 Jin-Yi Cai, Zhiguo Fu, and Shuai Shao. From Holant to quantum entanglement and back. In *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, pages 22:1–22:16, 2020. doi:10.4230/LIPIcs.ICALP.2020.22.

9 Jin-Yi Cai, Heng Guo, and Tyson Williams. The complexity of counting edge colorings and a dichotomy for some higher domain Holant problems. In *55th IEEE Annual Symposium on Foundations of Computer Science*, pages 601–610, 2014. doi:10.1109/FOCS.2014.70.

10 Jin-Yi Cai, Heng Guo, and Tyson Williams. A complete dichotomy rises from the capture of vanishing signatures. *SIAM J. Comput.*, 45(5):1671–1728, 2016. doi:10.1137/15m1049798.

11 Jin-Yi Cai, Heng Guo, and Tyson Williams. Clifford gates in the holant framework. *Theor. Comput. Sci.*, 745:163–171, 2018. doi:10.1016/j.tcs.2018.06.010.

12 Jin-Yi Cai, Sangxia Huang, and Pinyan Lu. From Holant to #CSP and back: Dichotomy for Holant^c problems. *Algorithmica*, 64(3):511–533, 2012. doi:10.1007/s00453-012-9626-6.

13 Jin-Yi Cai, Pinyan Lu, and Mingji Xia. Holant problems and counting CSP. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, pages 715–724, 2009. doi:10.1145/1536414.1536511.

14 Jin-Yi Cai, Pinyan Lu, and Mingji Xia. Dichotomy for Holant* problems with domain size 3. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1278–1295, 2013. doi:10.1137/1.9781611973105.93.

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

16 Jin-Yi Cai, Pinyan Lu, and Mingji Xia. Dichotomy for real Holant^c problems. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 1802–1821, 2018. doi:10.1137/1.9781611975031.118.

17 Martin E. Dyer, Leslie A. Goldberg, and Mark Jerrum. The complexity of weighted Boolean #CSP. *SIAM J. Comput.*, 38(5):1970–1986, 2009. doi:10.1137/070690201.

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

19 Leslie A. Goldberg, Martin Grohe, Mark Jerrum, and Marc Thurley. A complexity dichotomy for partition functions with mixed signs. *SIAM J. Comput.*, 39(7):3336–3402, 2010. doi:10.1137/090757496.

20 Sangxia Huang and Pinyan Lu. A dichotomy for real weighted Holant problems. *Computational Complexity*, 25(1):255–304, 2016. doi:10.1007/s00037-015-0118-3.

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

22 László Lovász. *Large Networks and Graph Limits*, volume 60 of *Colloquium Publications*. American Mathematical Society, 2012. URL: http://www.ams.org/bookstore-getitem/item=COLL-60.

23 Shuai Shao and Jin-Yi Cai. A dichotomy for real Boolean Holant problems. *To appear at FOCS*, 2020. arXiv:2005.07906.

24 Leslie G. Valiant. Holographic algorithms. *SIAM J. Comput.*, 37(5):1565–1594, 2008. doi:10.1137/070682575.