On the Complexity of Holant Problems

Heng Guo[1] and Pinyan Lu[2]

1 **School of Mathematical Sciences, Queen Mary University of London, London, UK**
h.guo@qmul.ac.uk

2 **Institute for Theoretical Computer Science, School of Information Management and Engineering, Shanghai University of Finance and Economics, Shanghai, China**
lu.pinyan@mail.shufe.edu.cn

Abstract

In this article we survey recent developments on the complexity of Holant problems. We discuss three different aspects of Holant problems: the decision version, exact counting, and approximate counting.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems, G.2.1 Combinatorics.

Keywords and phrases Computational complexity, Counting complexity, Dichotomy theorems, Approximate counting, Holant problems

Digital Object Identifier 10.4230/DFU.Vol7.15301.159

1 Introduction

Ladner's theorem [53] states that if P $\neq$ NP then there is an infinite hierarchy of intermediate problems that are not polynomial time interreducible. For certain restrictions of these classes, however, dichotomy theorems can be achieved. For NP a dichotomy theorem would state that any problem in the restricted subclass of NP is either in P or NP-complete (or both, in the eventuality that NP equals P.)

The restrictions for which dichotomy theorems are known can be framed in terms of local constraints, most importantly, Constraint Satisfaction Problems (CSP) [58, 28, 4, 5, 6, 33, 38, 27, 37], and Graph Homomorphism Problems [34, 42, 8]. Explicit dichotomy results, where available, manifest a total understanding of the class of computation in question, within polynomial time reduction, and modulo the collapse of the class.

In this article we survey dichotomies in a framework for characterizing local properties that is more general than those mentioned in the previous paragraph, namely the so-called Holant framework [18, 19]. A particular problem in this framework is characterized by a set of signatures as defined in the theory of Holographic Algorithms [65, 64]. The CSP framework can be viewed as a special case of the Holant framework in which equality relations of any arity are always assumed to be available in addition to the stated constraints. The extension from CSP to Holant problems enables us to express certain important problems such as graph matchings, which escape CSP [40] but are expressible in the Holant framework. Moreover, for the same constrain language, Holant problems contain potentially more structure than CSP. Indeed, in the Holant framework, new tractable cases emerge, the most notable among which is holographic algorithms [65].

© Heng Guo and Pinyan Lu;
licensed under Creative Commons License BY
The Constraint Satisfaction Problem: Complexity and Approximability. *Dagstuhl Follow-Ups*, Volume 7, ISBN 978-3-95977-003-3.
Editors: Andrei Krokhin and Stanislav Živný; pp. 159–177
Dagstuhl Publishing
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Germany

The graph matching problem plays an important role in the studies of complexity theory, leading to many exciting developments. Here is an (incomplete) list of related complexity and algorithmic results:

- There is a polynomial time algorithm to decide if a given graph has a perfect matching or not. This is the remarkable blossom algorithm by Edmonds [35]. In fact, in that paper, Edmonds proposed the complexity class P as the class of tractable problems.
- Counting perfect matchings is #P-Complete. This is proved by Valiant [60], right after he defined the class #P [61]. This problem is interesting since it shows that counting version may be much harder than the decision version for the same problem.
- Counting Perfect Matchings for planar graph is polynomial time solvable. This is the famous FKT algorithm [48, 59, 49]. It also serves as the computational primitive for Holographic algorithms [65].
- There is a polynomial time algorithm to compute the parity of the number of (perfect) matchings. The algorithm utilizes the fact that the value of permanent and determinate is the same modular 2. Thus matching problems have an interesting complexity transition from P and $\oplus$P to #P.
- There is a *fully polynomial-time randomized approximation scheme* (FPRAS) for approximately counting matchings. This is one of the first canonical examples of approximate counting [45]. The known algorithm is randomized. Deterministic algorithm is known for bounded degree graphs but open for general graphs [1].
- There is a FPRAS for approximately counting perfect matchings for bipartite graphs. The same algorithm can be used to approximate the permanent of nonnegative matrixes [47]. However it is a long-standing open question to generalize this algorithm to arbitrary graphs (or to show the impossibility for such an algorithm to exist).

From this list, we can see that the graph matching problem often sits right at the boundary between tractability and intractability. In order to understand the boundary of polynomial-time computation through the lens of dichotomy theorems, it is intrinsically important to include matching problems into consideration. Hence the natural framework to express matching problems, namely Holant problems, are more desirable (and more challenging at the same time) to understand than the conventional CSP framework. In this survey, we summarize results for both decision version and counting version of Holant problems with a focus on counting problems, since there is a lot of great progress in the last several years.

The Holant framework is strongly influenced by the development of holographic algorithms and holographic reductions [65, 64, 16, 18]. Indeed, holographic reductions are developed and applied as one of the primary techniques, which has not been used in the study of counting CSP (#CSP) previously. One advantage of the Holant framework is its flexibility. The conventional #CSP can be viewed as a special sub-framework of Holant by assuming that all equality functions are freely available. It is natural to assume other freely available classes of functions, such as the set of unary functions. When all unary functions are present, the framework becomes similar to the "conservative" case of CSP. We emphasize that the Holant framework is more flexible as a priori it assumes less freely available functions than the CSP.

In this survey, we put our attention mainly on Holant problems that cannot be expressed in the CSP framework. We will assume some familiarity to CSP as well as basic and classical dichotomy theorems for the CSP framework. Hence we can focus on new and interesting phenomena which are unique for the Holant framework.

Organization of the Survey

In Section 2, we formally define the framework of Holant Problems and some other basic notations. Section 3 summarizes some results for the decision version of Holant problems. Section 4 is the main section. We carefully discuss complexity dichotomies for the (exact) counting version of the Holant framework. Section 5 surveys some approximate counting results.

2 Definitions and Background

A *signature grid* $\Omega = (G, \mathcal{F}, \pi)$ is a tuple, where $G = (V, E)$ is a graph, $\mathcal{F}$ is a set of functions, and π is a mapping from the vertex set V to $\mathcal{F}$. A function $f \in \mathcal{F}$ with arity k is a mapping $[q]^k \to \mathbb{C}$, and the mapping π satisfies that the arity of $\pi(v)$ (which is a function $f \in \mathcal{F}$) is the same as the degree of v for any $v \in V$. Here we may consider any function with the range of a ring rather than just $\mathbb{C}$, but we choose $\mathbb{C}$ in this survey for clarity. Let $f_v := \pi(v)$ be the function on v. An assignment σ of edges is a mapping $E \to [q]$. The weight of σ is the evaluation $\prod_{v \in V} f_v(\sigma \mid_{E(v)})$, where $E(v)$ denotes the set of incident edges of v. The (counting version of) Holant problem on the instance Ω is to compute the sum of weights of all assignments; namely,

$$\text{Holant}_\Omega = \sum_\sigma \prod_{v \in V} f_v(\sigma \mid_{E(v)}). \tag{1}$$

We also write $\text{Holant}(\Omega; \mathcal{F})$ when we want to emphasize the function set $\mathcal{F}$.

The term Holant was first coined by Valiant in [65] to denote an exponential sum of the above form. Cai, Xia and Lu first formally introduced this framework of counting problems in [18, 19]. We can view each function f_v as a truth table, and then we can represent it by a vector in $\mathbb{C}^{q^{d(v)}}$, or a tensor in $(\mathbb{C}^q)^{\otimes d(v)}$. The vector or the tensor is called the *signature* of a function. When we say "function", we put a slight emphasis on that it is a mapping. When we say "signature", we put a slight emphasis on that it is ready to go through linear transformations. However most of the time in this survey, we use the two terms "function" and "signature" interchangeably without special attention.

A Holant problem is parameterized by a set of functions.

▶ **Definition 1.** Given a set of functions $\mathcal{F}$, we define a counting problem $\text{Holant}(\mathcal{F})$:
Input: A *signature grid* $\Omega = (G, \mathcal{F}, \pi)$;
Output: Holant_Ω.

We will use Pl-Holant$(\mathcal{F})$ to denote the problem where the input graph is planar.

The main goal here is to characterize what kind of function set $\mathcal{F}$ makes the problem $\text{Holant}(\mathcal{F})$ tractable (or hard).

The main focus of this survey is for functions over the Boolean domain $\{0, 1\}$, which we call Boolean functions. We use the following notations to denote some special functions. Let $=_k$ denote the equality function of arity k. Let Δ_s denote the constant unary function which gives value 1 on inputs $s \in [q]$, and 0 on all other inputs. Let $\textsc{ExactOne}_k$ denote the function that is one if the input has Hamming weight 1 and zero otherwise. Let $\mathcal{EO}$ be the set of $\textsc{ExactOne}_k$ functions for all integers k. Then $\text{Holant}(\mathcal{EO})$ is the same as the problem of counting perfect matchings.

A function is symmetric iff its function value is preserved under any permutation of its inputs. A symmetric function f on Boolean variables can be expressed by a compact signature $[f_0, f_1, \ldots, f_k]$, where f_i is the value of f on inputs of Hamming weight i. For the

Boolean domain $[2] = \{0, 1\}$, $=_k$ function has the signature $[1, 0, \ldots, 0, 1]$ with $k + 1$ entries and Δ_0 has $[1, 0]$. Moreover, $\textsc{ExactOne}_k$ has signature $[0, 1, 0, \ldots, 0]$ of $k + 1$ entries.

Multiplying a signature $f \in \mathcal{F}$ by a scaler $c \neq 0$ will not change the complexity of $\text{Holant}(\mathcal{F})$. So we always view f and cf as the same signature. In other words, we consider the projective space of vectors or tensors.

Another important property of signatures is degeneracy.

▶ **Definition 2.** A signature is called degenerate iff it can be decomposed into a tensor product of unary signatures.

In particular, a symmetric signature over a Boolean domain is degenerate iff it can be expressed as $\lambda[x, y]^{\otimes k}$.

We use $\text{Holant}\,(\mathcal{F} \mid \mathcal{G})$ to denote the Holant problem over signature grids with a bipartite graph $H = (U, V, E)$, where each vertex in U or V is assigned a signature in $\mathcal{F}$ or $\mathcal{G}$, respectively. Signatures in $\mathcal{F}$ are considered as row vectors (or covariant tensors); signatures in $\mathcal{G}$ are considered as column vectors (or contravariant tensors) (see, for example [30]). In this setting we sometimes write the Holant sum as $\text{Holant}(\Omega; \mathcal{F} \mid \mathcal{G})$ for input Ω. Let $\text{Pl-Holant}\,(\mathcal{F} \mid \mathcal{G})$ denote the Holant problem over signature grids with a planar bipartite graph.

2.1 Holographic Reductions

One key technique for Holant problems is holographic reductions. To introduce the idea, it is convenient to consider bipartite graphs. For a general graph, we can always transform it into a bipartite graph while preserving the Holant value as follows. For each edge in the graph, we replace it by a path of length two. (This operation is called the *2-stretch* of the graph and yields the edge-vertex incidence graph.) Each new vertex is assigned the binary $\textsc{Equality}$ signature $(=_2) = [1, 0, 1]$. Recall that $\text{Holant}\,(\mathcal{F} \mid \mathcal{G})$ denotes the Holant problem over signature grids with a bipartite graph $H = (U, V, E)$, where each vertex in U or V is assigned a signature in $\mathcal{F}$ or $\mathcal{G}$, respectively. Hence we have that $\text{Holant}(\mathcal{F}) \equiv_T \text{Holant}\,(=_2 \mid \mathcal{F})$.

For a 2-by-2 matrix T and a signature set $\mathcal{F}$, define $T\mathcal{F} = \{g \mid \exists f \in \mathcal{F}$ of arity $n,\ g = T^{\otimes n} f\}$, and similarly for $\mathcal{F}T$. Whenever we write $T^{\otimes n} f$ or $T\mathcal{F}$, we view the signatures as column vectors; similarly for $fT^{\otimes n}$ or $\mathcal{F}T$ as row vectors. In the special case that $T = \left[\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right]$, we use $\widehat{\mathcal{F}}$ to denote $T\mathcal{F}$.

Let T be an invertible 2-by-2 matrix. The holographic transformation defined by T is the following operation: given a signature grid $\Omega = (H, \pi)$ of $\text{Holant}\,(\mathcal{F} \mid \mathcal{G})$, for the same bipartite graph H, we get a new grid $\Omega' = (H, \pi')$ of $\text{Holant}\left(\mathcal{F}T \mid T^{-1}\mathcal{G}\right)$ by replacing each signature in $\mathcal{F}$ or $\mathcal{G}$ with the corresponding signature in $\mathcal{F}T$ or $T^{-1}\mathcal{G}$.

▶ **Theorem 3** (Valiant's Holant Theorem [65])**.** *If $T \in \mathbb{C}^{2\times 2}$ is an invertible matrix, then we have* $\text{Holant}(\Omega; \mathcal{F} \mid \mathcal{G}) = \text{Holant}(\Omega'; \mathcal{F}T \mid T^{-1}\mathcal{G})$.

Therefore, an invertible holographic transformation does not change the complexity of the Holant problem in the bipartite setting. Furthermore, there is a special kind of holographic transformation, the orthogonal transformation, that preserves the binary equality and thus can be used freely in the standard setting.

▶ **Theorem 4** (Theorem 2.6 in [20])**.** *If $T \in \mathbf{O}_2(\mathbb{C})$ is an orthogonal matrix (i.e. $TT^T = I_2$), then* $\text{Holant}(\Omega; \mathcal{F}) = \text{Holant}(\Omega'; T\mathcal{F})$.

We frequently apply a holographic transformation defined by the matrix $Z = \frac{1}{\sqrt{2}} \left[\begin{smallmatrix} 1 & 1 \\ i & -i \end{smallmatrix}\right]$ (or sometimes without the nonzero factor of $\frac{1}{\sqrt{2}}$ since this does not affect the complexity). This matrix has the property that the binary EQUALITY signature $(=_2) = [1, 0, 1]$ is transformed to $[1, 0, 1]Z^{\otimes 2} = [0, 1, 0] = (\neq_2)$, the binary DISEQUALITY signature.

By Theorem 3, we have that

$$\begin{aligned}\text{Holant}(\mathcal{F}) &\equiv \text{Holant}\left([1,0,1]T^{\otimes 2} \mid T^{-1}\mathcal{F}\right)\\ \text{Pl-Holant}(\mathcal{F}) &\equiv \text{Pl-Holant}\left([1,0,1]T^{\otimes 2} \mid T^{-1}\mathcal{F}\right),\end{aligned}$$

where $T \in \mathbf{GL}_2(\mathbb{C})$ is nonsingular. This leads to the notion of $\mathcal{C}$-*transformable*.

▶ **Definition 5.** Let $\mathcal{F}$ and $\mathcal{C}$ be two sets of signatures. We say $\mathcal{F}$ is $\mathcal{C}$-transformable if there exists a $T \in \mathbf{GL}_2(\mathbb{C})$ such that $[1, 0, 1]T^{\otimes 2} \in \mathcal{C}$ and $\mathcal{F} \subseteq T\mathcal{C}$.

The following lemma is immediate.

▶ **Lemma 6.** *If $\mathcal{F}$ is $\mathcal{C}$-transformable, then we have the following reductions.*

$$\begin{aligned}\text{Holant}(\mathcal{F}) &\leq_T \text{Holant}(\mathcal{C});\\ \text{Pl-Holant}(\mathcal{F}) &\leq_T \text{Pl-Holant}(\mathcal{C}).\end{aligned}$$

Clearly, if Holant$(\mathcal{C})$ or Pl-Holant$(\mathcal{C})$ is tractable, then Holant$(\mathcal{F})$ or Pl-Holant$(\mathcal{F})$ is tractable for any $\mathcal{C}$-transformable set $\mathcal{F}$.

2.2 Counting Constraint Satisfaction Problems

An instance of counting constraint satisfaction problems (#CSP$(\mathcal{F})$) has the following bipartite view. We have a set of vertices standing for variables and another set for functions (or constraints). Connect a variable vertex to a constraint vertex if the variable appears in the constraint. This bipartite graph is also known as the *constraint graph*. Moreover, each variable can be viewed as an EQUALITY function, as it forces the same value for all adjacent edges. Under this view, we see that

$$\#\text{CSP}(\mathcal{F}) \equiv_T \text{Holant}\left(\mathcal{EQ} \mid \mathcal{F}\right),$$

where $\mathcal{EQ} = \{=_1, =_2, =_3, \dots\}$ is the set of EQUALITY signatures of all arities.

The relationship between #CSP and Holant problems is the following:

$$\begin{aligned}\#\text{CSP}(\mathcal{F}) &\equiv_T \text{Holant}(\mathcal{EQ} \cup \mathcal{F});\\ \text{Pl-}\#\text{CSP}(\mathcal{F}) &\equiv_T \text{Pl-Holant}(\mathcal{EQ} \cup \mathcal{F}).\end{aligned}$$

Reductions from left to right are trivial. For the other direction, we take a signature grid Ω for the problem on the right and create a bipartite signature grid Ω' for the problem on the left such that both signature grids have the same Holant value. We simply create the equivalent bipartite grid Ω'' of Ω by replace each edge with a path of length 2 with $=_2$ in the middle point, as described earlier. Then we contract all EQUALITY signatures that are connected with each other, resulting in Ω' where EQUALITY signatures are on one side and signatures from $\mathcal{F}$ on the other.

3 Decision Version

In the decision version, we focus on the functions taking values in $\{0, 1\}$ and ask the question if the Holant value (as defined in (1)) is zero or not, or equivalently ask if there exists an assignment to satisfy all constraints or not. In this case, a function is a relation and it can also be viewed as a subset of all the possible assignments.

For the decision version of the CSP framework, it is a long standing open question to prove a dichotomy in general. But if we restrict to the Boolean domain, a classification is given by Schaefer [58] as one of the first computational complexity dichotomy theorems. The same dichotomy holds even if we restrict to the instances where each variable appears in at most three constraints. On the other hand, the decision version of Holant is equivalent to CSP where each variable appears at most twice. Its complexity classification, even for the Boolean domain, is still wide open and very interesting. To see why this is challenging, note that the perfect matching problem is a Holant problem defined by the $\textsc{ExactOne}_k$ function (with signature $[0, 1, 0 \dots, 0]$). Deciding the existence of a perfect matching is polynomial time solvable due to Edmonds's remarkable blossom algorithm [36]. However Edmonds's algorithm is highly non-trivial. Indeed it is much more complicated than any of the tractable cases in Schaefer's dichotomy for the CSP framework, and utilizes a lot of special structures intrinsic to the problem. It is hard to rule out the possibility of other similar tractable problems. The following family of Δ-matroid relations turn out to be the main obstacle. Let e_i denote the unit vector which is 0 on all indices other than i, on which its entry is 1.

▶ **Definition 7.** Let M be a subset of $\{0, 1\}^d$. It is called a Δ-*matroid* if for any pair of vectors $x, y \in M$ that differ on some index i, either $x \oplus e_i \in M$, or there exists another index $j \neq i$ on which x and y also differ and $x \oplus e_i \oplus e_j \in M$.

We say a relation R (or a $\{0, 1\}$ valued function f) is a Δ-matroid if the set of allowed assignments of R (or the set of inputs x such that $f(x) = 1$) is a Δ-matroid.

It is easy to verify that the perfect matching function ($\textsc{ExactOne}_k$ function) is a Δ-matroid according to the definition, but there are many more functions that are Δ-matroids. In particular, it was shown that there are functions in this family which cannot be expressed by composition of $\textsc{ExactOne}_k$ functions. Feder showed the following hardness result: unless all relations in F are Δ-matroids, the decision $\text{Holant}(\mathcal{F})$ has the same complexity as $\text{CSP}(\mathcal{F})$ [39]. Based on this hardness result, we have the following classification result.

▶ **Theorem 8.** *All decision* $\text{Holant}(\mathcal{F})$ *problems are divided into three classes according to* $\mathcal{F}$*:*

1. *Every function in* $\mathcal{F}$ *is a* Δ*-matroid;*
2. *If CSP(*$\mathcal{F}$*) is tractable according to the dichotomy classification of Schaefer [58], then* $\text{Holant}(\mathcal{F})$ *is also tractable;*
3. *Otherwise,* $\text{Holant}(\mathcal{F})$*is NP-complete.*

The only remaining open case is the complexity of Δ-matroid functions in the Holant framework. A number of tractable classes of Δ-matroids have been identified [26, 39, 29, 41, 50], but it seems to be still quite challenging to settle the complexity for the whole class. A recurring theme is the connection between Δ-matroids and matching problems. Here we mention two known tractable classes of Δ-matroids.

For a symmetric relation $[f_0, f_1, \dots, f_k]$ where $f_i \in \{0, 1\}$, the number of consecutive 0's between two 1's is called a gap. For example, $[0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0]$ have two gaps with length 2 and 3 respectively, while the one 0 in the beginning and two 0's at the end are not viewed as gaps. It is not difficulty to verify that, a symmetric relation is a Δ-matroid if and only if it has no gap with length larger than one. For all symmetric Δ-matroid relations, the decision Holant problem is tractable [26].

▶ **Theorem 9.** *If all the relations in* $\mathcal{F}$ *are symmetric and* Δ*-matroid, i.e. the largest gap of any relation in* $\mathcal{F}$ *has length* 1*, then there is a polynomial time algorithm to decide* $\text{Holant}(\mathcal{F})$*.*

Another broad family of tractable functions is called even Δ-matroid relations, shown recently by Kazda, Kolmogorov, and Rolínek [50]. A Δ-matroid relation M is called even if all vectors in M have the same parity of their Hamming weights; that is, they all have even Hamming weights or all have odd Hamming weights.

▶ **Theorem 10.** *If F contains only even Δ-matroid relations, then decision* Holant$(\mathcal{F})$ *can be solved in polynomial time.*

This tractability result for even Δ-matroid relations leads to a complete complexity dichotomy of Boolean CSP on planar graphs (see [31, 50]).

4 Exact Counting

There has been a lot of progress in understanding the complexity of computing the Holant sum exactly. In particular, we have a thorough understanding of Holant problems defined by Boolean symmetric functions, even if the input is restricted to planar graphs and the weights are complex.

The following theorem is a combination of [11, 9], the culminating results from a long line of research [23, 19, 21, 51, 52, 15, 14, 13, 44]. We are interested in general input graphs as well as planar graphs. The Holant framework was first proposed to systematically study the power of Valiant's holographic algorithms [65], which was designed to solve counting problems in planar graphs. When inputs are planar graphs, the problem is denoted by Pl-Holant$(\mathcal{F})$.

▶ **Theorem 11.** *Let $\mathcal{F}$ be a set of Boolean symmetric functions with complex weights.* Holant$(\mathcal{F})$ *either has a polynomial time algorithm, or is* $\#\mathbf{P}$*-hard to compute. This dichotomy also holds for* Pl-Holant$(\mathcal{F})$ *(but the tractable criterion is different).*

In fact, we know more than merely that the dichotomy holds. (This is non-trivial due to Ladner's Theorem [53]) We have a complete explicit criteria for tractable sets of functions in Theorem 11. In order to describe the criteria, we will first introduce some families of functions that appear as tractable cases in the dichotomy theorem.

4.1 Tractable Families

We summarize several known sets of tractable Boolean functions with complex weights. The first one is very simple. If all signatures are degenerate or binary, then the problem is tractable.

For a binary signature, define its matrix as

$$M_f := \begin{bmatrix} f(00) & f(01) \\ f(10) & f(11) \end{bmatrix}. \tag{2}$$

Connecting f to g via one edge gives another signature h with the matrix $M_h = M_f M_g$.

▶ **Lemma 12.** *Let $\mathcal{F}$ be a set of complex weighted symmetric signatures in Boolean variables. Then* Holant$(\mathcal{F})$ *is computable in polynomial time if all non-degenerate signatures in $\mathcal{F}$ are of arity at most 2.*

Proof. We first replace degenerate signatures by a bunch of equivalent unary signatures. Then any instance of Holant$(\mathcal{F})$ can be decomposed into paths and cycles. The Holant is a product of all paths and cycles.

For a path, we remove the two endpoints, leaving a binary signature f composed by a series of binary signatures. Compute the signature matrix M_f of f by multiplying all binary signatures along the path. Then the Holant is vM_fu^{T}, where v and u are the two unary signatures at endpoints.

For a cycle, we arbitrarily break an edge getting a path with two dangling edges. Similar to the above case, we multiply matrices of all binary signatures along this path, getting M. The trace of M is the Holant. ◀

We further note that for a binary signature f and $T \in \mathbb{C}^{2\times 2}$, let $g = fT^{\otimes 2}$. Then

$$M_g = TM_fT^{\mathsf{T}}. \tag{3}$$

This can be seen by viewing T as a binary, and then treating g as connecting T, f, and T^{T} sequentially.

4.1.1 Affine Signatures

▶ **Definition 13** (Definition 3.1 in [25]). A k-ary function $f(x_1, \ldots, x_k)$ is *affine* if it has the form

$$\lambda \cdot \chi_{Ax=0} \cdot i^{\sum_{j=1}^{n} \langle \mathbf{v}_j, x\rangle},$$

where $\lambda \in \mathbb{C}$, $x = (x_1, x_2, \ldots, x_k, 1)^{\mathsf{T}}$, A is a matrix over $\mathbb{F}_2$, $\mathbf{v}_j$ is a vector over $\mathbb{F}_2$, and χ is a 0-1 indicator function such that $\chi_{Ax=0}$ is 1 if and only if $Ax = 0$. Note that the dot product $\langle \mathbf{v}_j, x\rangle$ is calculated over $\mathbb{F}_2$, while the summation $\sum_{j=1}^{n}$ on the exponent of $i = \sqrt{-1}$ is evaluated as a sum mod 4 of 0-1 terms. We use $\mathcal{A}$ to denote the set of all affine functions.

The matrix A defines an affine space which is the support of the signature f (and hence the name). Notice that there is no restriction on the number of rows in the matrix A. It is permissible that A is the zero matrix so that $\chi_{Ax=0} = 1$ holds for all x. An equivalent way to express the exponent of i is as a quadratic polynomial where all cross terms have an even coefficient (cf. [7]).

It is known that the set of non-degenerate symmetric signatures in $\mathcal{A}$ is precisely the nonzero signatures ($\lambda \neq 0$) in $\mathcal{F}_1 \cup \mathcal{F}_2 \cup \mathcal{F}_3$ with arity at least 2, where $\mathcal{F}_1$, $\mathcal{F}_2$, and $\mathcal{F}_3$ are three families of signatures defined as

$$\begin{aligned}
\mathcal{F}_1 &= \left\{\lambda\left([1,0]^{\otimes k} + i^r[0,1]^{\otimes k}\right) \mid \lambda \in \mathbb{C}, k = 1,2,\ldots, r = 0,1,2,3\right\}, \\
\mathcal{F}_2 &= \left\{\lambda\left([1,1]^{\otimes k} + i^r[1,-1]^{\otimes k}\right) \mid \lambda \in \mathbb{C}, k = 1,2,\ldots, r = 0,1,2,3\right\}, \\
\mathcal{F}_3 &= \left\{\lambda\left([1,i]^{\otimes k} + i^r[1,-i]^{\otimes k}\right) \mid \lambda \in \mathbb{C}, k = 1,2,\ldots, r = 0,1,2,3\right\}.
\end{aligned} \tag{4}$$

We explicitly list these signatures up to an arbitrary constant multiple from $\mathbb{C}$, see Table 1.

The tractability of $\mathcal{A}$ is first shown in [25]. It was later generalized to arbitrary domain size using Gauss sums [7]. Together with Lemma 6, we have the following.

▶ **Lemma 14.** *Let $\mathcal{F}$ be any set of symmetric, complex-valued signatures in Boolean variables. If $\mathcal{F}$ is $\mathcal{A}$-transformable, then* $\operatorname{Holant}(\mathcal{F})$ *is computable in polynomial time.*

4.1.2 Product-Type Signatures

▶ **Definition 15** (Definition 3.3 in [25]). A function is of *product type* if it can be expressed as a product of unary functions, binary equality functions ($[1,0,1]$), and binary disequality functions ($[0,1,0]$). We use $\mathcal{P}$ to denote the set of product-type functions.

Table 1 List of all non-degenerate affine signatures.

1.	$[1, 0, \ldots, 0, \pm 1]$;	$(\mathcal{F}_1, r = 0, 2)$
2.	$[1, 0, \ldots, 0, \pm i]$;	$(\mathcal{F}_1, r = 1, 3)$
3.	$[1, 0, 1, 0, \ldots, 0 \text{ or } 1]$;	$(\mathcal{F}_2, r = 0)$
4.	$[1, -i, 1, -i, \ldots, (-i) \text{ or } 1]$;	$(\mathcal{F}_2, r = 1)$
5.	$[0, 1, 0, 1, \ldots, 0 \text{ or } 1]$;	$(\mathcal{F}_2, r = 2)$
6.	$[1, i, 1, i, \ldots, i \text{ or } 1]$;	$(\mathcal{F}_2, r = 3)$
7.	$[1, 0, -1, 0, 1, 0, -1, 0, \ldots, 0 \text{ or } 1 \text{ or } (-1)]$;	$(\mathcal{F}_3, r = 0)$
8.	$[1, 1, -1, -1, 1, 1, -1, -1, \ldots, 1 \text{ or } (-1)]$;	$(\mathcal{F}_3, r = 1)$
9.	$[0, 1, 0, -1, 0, 1, 0, -1, \ldots, 0 \text{ or } 1 \text{ or } (-1)]$;	$(\mathcal{F}_3, r = 2)$
10.	$[1, -1, -1, 1, 1, -1, -1, 1, \ldots, 1 \text{ or } (-1)]$.	$(\mathcal{F}_3, r = 3)$

An alternate definition for $\mathcal{P}$, implicit in [22], is the tensor closure of signatures with support on two complementary bit vectors. It is easily shown (cf. Lemma A.1 in the full version of [44]) that if f is a symmetric signature in $\mathcal{P}$, then f is degenerate, binary DISEQUALITY $\neq_2$, or $[a, 0, \ldots, 0, b]$ for some $a, b \in \mathbb{C}$.

The tractability of $\mathcal{P}$ is due to a straightforward propagation algorithm (see, for example [25]). Together with Lemma 6, we have the following.

▶ **Lemma 16.** *Let $\mathcal{F}$ be any set of symmetric, complex-valued signatures in Boolean variables. If $\mathcal{F}$ is $\mathcal{P}$-transformable, then* Holant$(\mathcal{F})$ *is computable in polynomial time.*

4.1.3 Vanishing Signatures

Vanishing signatures define Holant problems where the Holant sum is always 0.

▶ **Definition 17.** A set of signatures $\mathcal{F}$ is called *vanishing* if $\operatorname{Holant}(\Omega; \mathcal{F}) = 0$ for every signature grid Ω. A signature f is called *vanishing* if the singleton set $\{f\}$ is vanishing.

A useful way to understand vanishing signatures is via a low rank tensor decomposition. To state these decompositions, we use the following definition.

▶ **Definition 18.** Let S_n be the symmetric group of degree n. Then for positive integers t and n with $t \leq n$ and unary signatures $v, v_1, \ldots, v_{n-t}$, we define

$$\operatorname{Sym}_n^t(v; v_1, \ldots, v_{n-t}) = \sum_{\pi \in S_n} \bigotimes_{k=1}^{n} u_{\pi(k)},$$

where the ordered sequence $(u_1, u_2, \ldots, u_n) = (\underbrace{v, \ldots, v}_{t \text{ copies}}, v_1, \ldots, v_{n-t})$.

With this notation we can define the vanishing degree.

▶ **Definition 19.** A nonzero symmetric signature f of arity n has *positive vanishing degree* $k \geq 1$, denoted by $\operatorname{vd}^+(f) = k$, if $k \leq n$ is the largest positive integer such that there exists $n - k$ unary signatures $v_1, \ldots, v_{n-k}$ such that

$$f = \operatorname{Sym}_n^k([1, i]; v_1, \ldots, v_{n-k}).$$

If f cannot be expressed as such a symmetrization form, we define $\operatorname{vd}^+(f) = 0$. If f is the all zero signature, define $\operatorname{vd}^+(f) = n + 1$.

We define *negative vanishing degree* vd^- similarly, using $-i$ instead of i.

It is possible that both $\mathrm{vd}^+(f)$ and $\mathrm{vd}^-(f)$ are nonzero. For example, $\mathrm{vd}^+(=_2) = \mathrm{vd}^-(=_2) = 1$.

The following theorem completely characterizes symmetric vanishing signatures. It is proved in [11]. For $\sigma \in \{+, -\}$, let $\mathcal{V}^\sigma := \{f \mid 2\,\mathrm{vd}^\sigma(f) > \mathrm{arity}(f)\}$.

▶ **Theorem 20.** *Let $\mathcal{F}$ be a set of symmetric signatures. Then $\mathcal{F}$ is vanishing if and only if $\mathcal{F} \subseteq \mathcal{V}^+$ or $\mathcal{F} \subseteq \mathcal{V}^-$.*

Obviously, vanishing signatures define tractable Holant problems. The algorithm is simple – just output 0! However, vanishing signatures can be combined with other functions and remain tractable. The following two lemma are shown in [11].

▶ **Lemma 21.** *Let $\sigma = +$ or $-$. Let $\mathcal{F}$ be a set of complex weighted symmetric signatures in Boolean variables. Then* $\mathrm{Holant}(\mathcal{F})$ *is computable in polynomial time if $\mathcal{F} \subseteq \mathcal{V}^\sigma \cup \{f \mid \mathrm{vd}^\sigma(f) \geq 1 \,\&\, \mathrm{arity}(f) = 2\}$.*

▶ **Lemma 22.** *Let $\sigma = +$ or $-$. Let $\mathcal{F}$ be a set of complex weighted symmetric signatures in Boolean variables. Then* $\mathrm{Holant}(\mathcal{F})$ *is computable in polynomial time if any non-degenerate signature $f \in \mathcal{F}$ satisfies that* $\mathrm{vd}^\sigma(f) \geq \mathrm{arity}(f) - 1$.

Lemma 21 can be understood as putting vanishing signatures and certain kind of binary signatures together remain tractable. Lemma 22 can be understood as highly vanishing signatures ($\mathrm{vd}^\sigma(f) \geq \mathrm{arity}(f) - 1$) can be put together with all unary signatures and remain tractable, since unary signatures automatically satisfy the condition $\mathrm{vd}^\sigma(f) \geq \mathrm{arity}(f) - 1 = 0$.

4.1.4 Matchgate Signatures

Matchgates were introduced by Valiant [63, 62] to give polynomial-time algorithms for a collection of counting problems over planar graphs. As the name suggests, problems expressible by matchgates can be reduced to computing a weighted sum of perfect matchings. The latter problem is tractable over planar graphs by Kasteleyn's algorithm [49]. Historically the algorithm was first found by Temperley and Fisher for $\mathbb{Z}^2$ [59] and independently by Kasteleyn [48]. It was later generalized to general planar graphs by Kastelyn [49]. Hence sometimes it is also called the FKT algorithm. These counting problems are naturally expressed in the Holant framework using *matchgate signatures*, denoted by $\mathcal{M}$. Thus Pl-Holant($\mathcal{M}$) is tractable.

Formally, recall that $\mathcal{EO}$ is the set of $\textsc{ExactOne}_k$ functions for all integers k. Let $\mathcal{WEO}$ be the set of weighted $\textsc{ExactOne}_k$ functions for all k. Then $\mathcal{M}$ contains signatures that can be realized as an $\mathcal{WEO}$-gate (realizable by functions in the set $\mathcal{WEO}$). Holographic transformations extend the reach of the FKT algorithm even further by Lemma 6, as stated below.

▶ **Lemma 23.** *Let $\mathcal{F}$ be any set of symmetric, complex-valued signatures in Boolean variables. If $\mathcal{F}$ is $\mathcal{M}$-transformable, then* Pl-Holant($\mathcal{F}$) *is computable in polynomial time.*

Matchgate signatures are characterized by the matchgate identities (for an up-to-date treatment, see [10] for the identities and a self-contained proof). Any matchgate signature f must satisfy the *parity condition*, which asserts that the support of f has to contain entries of only even or odd Hamming weights, but not both. For symmetric matchgates, they have 0 for every other entry and form a geometric progression with the remaining entries. We explicitly list all the symmetric signatures in $\mathcal{M}$ (see [10]).

▶ **Proposition 24.** *Let f be a symmetric signature in $\mathcal{M}$. Then there exists $a, b \in \mathbb{C}$ and $n \in \mathbb{N}$ such that f takes one of the following forms:*
1. $[a^n, 0, a^{n-1}b, 0, \ldots, 0, ab^{n-1}, 0, b^n]$ *(of arity $2n \geq 2$);*
2. $[a^n, 0, a^{n-1}b, 0, \ldots, 0, ab^{n-1}, 0, b^n, 0]$ *(of arity $2n+1 \geq 1$);*
3. $[0, a^n, 0, a^{n-1}b, 0, \ldots, 0, ab^{n-1}, 0, b^n]$ *(of arity $2n+1 \geq 1$);*
4. $[0, a^n, 0, a^{n-1}b, 0, \ldots, 0, ab^{n-1}, 0, b^n, 0]$ *(of arity $2n+2 \geq 2$).*

In the last three cases with $n = 0$, the signatures are $[1, 0]$, $[0, 1]$, and $[0, 1, 0]$. Any multiple of these is also a matchgate signature.

Note that perfect matching signatures, $[0, 1, 0, \cdots, 0]$, and their reversal are special cases when $b = 0$ or $a = 0$ in the last two cases.

Similar to vanishing signatures, signatures in $\mathcal{M}$ have low rank decompositions as well.

▶ **Proposition 25.** *Let f be a symmetric signature in $\mathcal{M}$ of arity n. Then there exist $a, b, \lambda \in \mathbb{C}$ such that f takes one of the following forms:*
1. $[a, b]^{\otimes n} + [a, -b]^{\otimes n} = \begin{cases} 2[a^n, 0, a^{n-2}b^2, 0, \ldots, 0, b^n] & n \text{ is even,} \\ 2[a^n, 0, a^{n-2}b^2, 0, \ldots, 0, ab^{n-1}, 0] & n \text{ is odd;} \end{cases}$
2. $[a, b]^{\otimes n} - [a, -b]^{\otimes n} = \begin{cases} 2[0, a^{n-1}b, 0, a^{n-3}b^3, 0, \ldots, 0, ab^{n-1}, 0] & n \text{ is even,} \\ 2[0, a^{n-1}b, 0, a^{n-3}b^3, 0, \ldots, 0, b^n] & n \text{ is odd;} \end{cases}$
3. $\lambda \operatorname{Sym}_n^{n-1}([1, 0]; [0, 1]) = [0, \lambda, 0, \ldots, 0]$;
4. $\lambda \operatorname{Sym}_n^{n-1}([0, 1]; [1, 0]) = [0, \ldots, 0, \lambda, 0]$.

The understanding of matchgates was further developed in [17], which characterized, for every symmetric signature, the set of holographic transformations under which the transformed signature becomes a matchgate signature.

4.1.5 An Extra Planar Tractable Case

In [9], towards a complete planar dichotomy theorem, a new tractable case was found for planar graphs.

Recall that $\mathcal{EO}$ is the set of functions $\textsc{ExactOne}_k$ for all arities k, and $Z = \left[\begin{smallmatrix} 1 & 1 \\ i & -i \end{smallmatrix}\right]$. Let $\mathcal{EO}'$ be the set of inverses of $\textsc{ExactOne}_k$ for all arities k. Namely, $f \in \mathcal{EO}'$ requires the input to have hamming weight exactly (arity -1). Let $\mathcal{WEQ}$ denotes the set of weighted equality functions. Moreover, let $\mathcal{F}^*$ denote $\mathcal{F}$ with all degenerate signatures $[a, b]^{\otimes m}$ replaced by unary $[a, b]$. Then we have the following lemma [9].

▶ **Lemma 26.** *Let $\mathcal{F}$ be a set of symmetric Boolean functions. If $\mathcal{F} \subseteq Z\mathcal{P} \cup Z(\mathcal{EO})$ or $\mathcal{F} \subseteq Z\mathcal{P} \cup Z(\mathcal{EO}')$, and the greatest common divisor of the arities of the signatures in $\mathcal{F}^* \cap Z(\mathcal{WEQ})$ is at least 5, then* Holant$(\mathcal{F})$ *is tractable.*

The algorithm of this special case is a recursive procedure to find edges that either have to be a particular value, or do not have satisfying assignments. We can show that either these edges show up in the graph, or the instance falls into one of the tractable cases above. The existence of these edges (namely, when the instance is not solvable by previous cases) is due to the degree rigidity of a planar graph. (For example, the average degree of a planar graph cannot be more than 6.)

4.2 The Full Dichotomy

After introducing tractable families, we can finally state the dichotomy theorem in full detail.

▶ **Theorem 27.** *Let $\mathcal{F}$ be any set of symmetric, complex-valued functions in Boolean variables. Then* Pl-Holant$(\mathcal{F})$ *is* $\#\mathbf{P}$*-hard unless $\mathcal{F}$ satisfies one of the following conditions:*

1. *All non-degenerate signatures in $\mathcal{F}$ are of arity at most 2;*
2. *$\mathcal{F}$ is $\mathcal{A}$-transformable;*
3. *$\mathcal{F}$ is $\mathcal{P}$-transformable;*
4. $\mathcal{F} \subseteq \mathcal{V}^{\sigma} \cup \{f \mid \mathrm{vd}^{\sigma}(f) \geq 1 \& \, \mathrm{arity}(f) = 2\}$ *for some* $\sigma \in \{+, -\}$*;*
5. *Any non-degenerate signature $f \in \mathcal{F}$ satisfies* $\mathrm{arity}(f) - \mathrm{vd}^{\sigma}(f) \leq 1$ *for some* $\sigma \in \{+, -\}$.
6. *$\mathcal{F}$ is $\mathcal{M}$-transformable;*
7. $\mathcal{F} \subseteq Z\mathcal{P} \cup Z(\mathcal{E}\mathcal{O})$ *or* $\mathcal{F} \subseteq Z\mathcal{P} \cup Z(\mathcal{E}\mathcal{O}')$*, and the greatest common divisor of the arities of the signatures in* $\mathcal{F}^* \cap Z(\mathcal{W}\mathcal{E}\mathcal{Q})$ *is at least* 5.

In each exceptional case, Pl-Holant$(\mathcal{F})$ *is computable in polynomial time. If $\mathcal{F}$ satisfies conditions 1 to 5, then* Holant$(\mathcal{F})$ *is computable in polynomial time without planarity; otherwise* Holant$(\mathcal{F})$ *is* $\#\mathbf{P}$*-hard.*

4.3 Beyond Boolean and Symmetric Functions

In full generality, we would like to understand the complexity of Holant problems defined by any set of functions, rather than just symmetric Boolean functions. However, the understanding of those Holant problems is far from complete.

Still in the Boolean domain, the best dichotomy result we know of regarding asymmetric functions is [22]. A crucial constraint for the result of [22] is that it requires unary functions to be available freely. This corresponds to the "conservative" case in the study of CSP problems. We use Holant$^*(\mathcal{F})$ to denote these problems.

For asymmetric functions, we need to be careful to state the result. Let $\langle \mathcal{F} \rangle$ of a set $\mathcal{F}$ denote its tensor closure; namely, $\langle \mathcal{F} \rangle$ is the minimum set containing $\mathcal{F}$, closed under tensor product. This closure exists, being the set of all functions obtained by taking a finite sequence of tensor products from $\mathcal{F}$.

Let $\mathcal{T}$ be the set of all unary and binary functions. Let $\mathcal{E}$ be the set of all functions f such that f is zero except on two inputs $(x_1, \ldots, x_n)$ and $(1 - x_1, \ldots, 1 - x_n)$. In other words, $f \in \mathcal{E}$ iff its support is contained in a pair of complementary points. We think of $\mathcal{E}$ as a generalized form of equality functions. Let $\mathcal{M}$ be the set of all functions f such that f is zero except on $n + 1$ inputs whose Hamming weight is at most 1, where n is the arity of f. We think of $\mathcal{M}$ as a generalized form of matchings.

Recall that $Z = \left[\begin{smallmatrix} 1 & 1 \\ i & -i \end{smallmatrix}\right]$. Let $Z' = \left[\begin{smallmatrix} 1 & 1 \\ -i & i \end{smallmatrix}\right]$. Then we have the following theorem [22].

▶ **Theorem 28.** *Let $\mathcal{F}$ be any set of complex valued functions in Boolean variables. The problem* Holant$^*(\mathcal{F})$ *is polynomial time computable, if (1) $F \subseteq \langle \mathcal{T} \rangle$, or (2) there exists an orthogonal matrix H such that $\mathcal{F} \subseteq H$, or (3) $\mathcal{F} \subseteq \langle Z\mathcal{E} \rangle$ or $\mathcal{F} \subseteq \langle Z'\mathcal{E} \rangle$, or (4) $\mathcal{F} \subseteq \langle Z\mathcal{M} \rangle$ or $\mathcal{F} \subseteq \langle Z'\mathcal{E} \rangle$. In all other cases,* Holant$^*(\mathcal{F})$ *is* $\#\mathbf{P}$*-hard. The dichotomy is still true even if the inputs are restricted to planar graphs.*

Going beyond the Boolean domain, [24] gives a dichotomy theorem regarding Holant* problems defined by a single ternary symmetric function over a domain of size 3. The statement is rather technical and we refer the interested readers to [24] for details.

For even larger domain sizes, we know the complexity of counting k-edge-colourings over (planar) d-regular graphs for any pair of integers (k, d) [12]. Edge colourings are special cases of Holant problems where the domain size is k and the constraint on the vertex requires that all inputs are distinct. Following this path a dichotomy is known for Holant problems defined by ternary functions and with certain high symmetry [12]. This symmetry requirement is inspired by the All-Distinct constraint of edge colorings. For simplicity here we only state the edge coloring result and refer the reader to [12] for the more technical dichotomy theorem.

▶ **Theorem 29.** *Counting k-edge-colourings is $\#\mathbf{P}$-hard over planar d-regular (multi-)graphs if $k \geq d \geq 3$.*

Note that if $d \leq 2$ the problem is trivial, and if $k < d$ there is no such colourings.

5 Approximate Counting

In this last section, we study the approximation version of counting problems. For any given parameter $\epsilon > 0$, the algorithm outputs a number $\hat{Z}$ such that $(1-\epsilon)Z \leq \hat{Z} \leq (1+\epsilon)Z$, where Z is the accurate Holant summation of the input instance. We also require that the running time of the algorithm is bounded by $poly(n, 1/\epsilon)$, where n is the number of vertices of the given graph. This is called a fully polynomial-time approximation scheme (FPTAS). The randomized relaxation of FPTAS is called fully polynomial-time randomized approximation scheme (FPRAS), which uses random bits in the algorithm and requires that the final output is within the range $[(1-\epsilon)Z, (1+\epsilon)Z]$ with high probability.

Recall that we may view Holant problems as a CSP where each variable appears at most twice. The CSP problem with a degree bound is not necessarily of the same computational complexity as the problem without the degree bound even if the degree bound is larger than 2. New and interesting tractable families show up. For degree bounds larger than 2, a partial classification was known (see, for example [32]). On the other hand, the situation of Holant (bounded degree 2) is wide open, and it seems that there are many more tractable problems. Here we list a number of interesting ones.

Matching. There is an FPRAS for counting the number of matchings, even with weights [45].

Parity Function. A parity function is a symmetric function of form $[a, b, a, b, \cdots]$. If the constraint in each vertex is a parity function, there is an FPRAS for computing the partition function for any weighted graphs [46]. By transforming to this Holant problem (which was called the "subgraph world" problem in [46]) of parity functions, an FPRAS for ferromenaginic Ising model was given by Jerrum and Sinclair [46]

SAT. For SAT instances where each variable appears in at most two clauses, there is an FPRAS to count the number of satisfying assignments [3].

Not-All-Equal. Let $\textsc{NotAllEqual}_k$ be the symmetric function that is 0 if the input has Hamming weights 0 or k, and 1 otherwise; namely its signature is $[0, 1, 1, \cdots, 1, 0]$ with $k+1$ entries. Let $\mathcal{NAE}$ be the set of $\textsc{NotAllEqual}_k$ functions for all integers k. There is an FPRAS for Holant($\mathcal{NAE}$) [57].

5.1 Winding

One powerful approach to design approximate counting algorithms is Markov Chain Monte Carlo (MCMC). The key step is to prove that the Markov chain is rapidly mixing, namely, it is very close to the stationary distribution after polynomial number of steps. Canonical paths argument, introduced by [45, 46] is one of the two main tools (the other one is coupling) to prove rapid mixing of the Markov chain. To make use of canonical paths, one needs to design paths between each pair of states for the Markov chain and prove that the overall congestion at each transition of the Markov chain is low. However, it is typically a very difficult task to come up with a low congestion routing, especially because there are usually exponentially many states corresponding to the Markov chain.

There are some successful examples such as the matching problem mentioned above. The symmetric difference of two matchings of a graph is a disjoint union of paths and cycles. Then, the natural and successful canonical paths for matchings is "(un-)winding" the edges one by one following an arbitrary order of these paths and cycles. Another important successful example is the "subgraph world" problem (or the parity function problem, as described in the last section) transformed from ferromagnetic Ising model [46]. For this problem, the symmetric difference of two configurations can be any graphs. The key observation is that we may utilize the path-cycle decomposition. Jerrum and Sinclair's canonical paths simply do an arbitrary path-cycle decomposition and unwind edges following these paths and cycles. Since the constraint in each vertex for that problem is simply the parity function, one can prove that these canonical paths indeed have low congestion.

In an unpublished manuscript [57], McQuillan proposed a beautiful generalization of this path-cycle decomposition idea called winding. The idea was further developed in [43]. Here is the definition of windable functions.

▶ **Definition 30.** For any finite set J and any configuration $x \in \{0,1\}^J$, define $\mathcal{M}_x$ to be the set of partitions of $\{i \mid x_i = 1\}$ into pairs and at most one singleton. A function $f : \{0,1\}^J \to \mathbb{R}^+$ is **windable** if there exist values $B(x,y,M) \geq 0$ for all $x, y \in \{0,1\}^J$ and all $M \in \mathcal{M}_{x \oplus y}$ satisfying:

1. $f(x)f(y) = \sum_{M \in \mathcal{M}_{x \oplus y}} B(x,y,M)$ for all $x, y \in \{0,1\}^J$, and
2. $B(x,y,M) = B(x \oplus S, y \oplus S, M)$ for all $x, y \in \{0,1\}^J$ and all $S \in M \in \mathcal{M}_{x \oplus y}$.

Here $x \oplus S$ denotes the vector obtained by changing x_i to $1 - x_i$ for the one or two elements i in S.[1]

To get a rapidly mixing Markov chain, we assign two values to the two half-edges of each edge. We call it consistent if the two values are the same. A normal edge assignment is therefore an assignment of half-edges without inconsistency. We call these assignments perfect. A near-perfect assignment is one where there are two inconsistencies along edges. Windable functions admit a rapidly mixing Markov chain, by moving between perfect assignments and near-perfect assignments. Due to the design of the algorithm, the number of inconsistencies cannot be one.

However since we enlarge the state space slightly, merely a rapidly mixing Markov chain is not sufficient to guarantee a polynomial time algorithm. We also need to be able to hit perfect assignments with at least inverse polynomial probability. This can be stated as a bound between the Holant sum of all perfect assignments and that of all near-perfect assignments. More precisely, recall (1). The weight of an assignment can be naturally extend to near-perfect assignments, and their Holant sum is to simply add all weights up. Denote by Z_0 the Holant of all perfect assignments, and Z_2 that of all near-perfect assignments. Then we need to ensure that Z_2/Z_0 is bounded above by a polynomial for the MCMC algorithm to run in polynomial time.

▶ **Theorem 31.** *There exists an FPRAS to compute the partition function of* $\mathrm{Holant}(\mathcal{F})$ *if all the functions in* $\mathcal{F}$ *are windable and* Z_2/Z_0 *is bounded above by a polynomial of the input size.*

One can verify that the matching constraint [45] and the parity function [46] are indeed windable. Thus both of these two FPRASes can be viewed as special cases of Theorem 31.

[1] This definition is taken from [43], which simplifies the original definition from [57].

However for a general function, it is still quite difficulty to tell whether it is windable or not. A clear characterization was given for all symmetric functions in [43] by solving a set of linear equations. With this powerful approach and characterization in hand, one can design a number of new FPRAS for approximate counting by simply verifying that the local constraint functions are windable. One such example is counting b-matchings, which is a natural generalization of matchings. A subset of edges for a graph is called a b-matching if every vertex is incident to at most b edges in the set. Hence 1-matching is the conventional definition of matching for a graph. Huang, Lu, and Zhang [43] showed that there exists an FPRAS to count b-matchings when $b \leq 7$ for any graphs.

Another problem one can resolve using this approach is a generalization of the edge cover problem. A subset of edges for a graph is called an edge cover if every vertex is incident to at least one edge in the set. Previously, MCMC based approximation algorithm for counting edge covers was only known for 3-regular graphs [2] by Bezáková and Rummler. In fact, they also used canonical paths to get rapid mixing and used path-cycle decompositions to construct canonical paths. However, due to the lack of a systematic approach, Bezáková and Rummler stopped at the special case of 3-regular graphs. Using the winding approach and the systematic characterization of windable functions, one can show that there exist a convex combination of path-cycle decompositions which works for general graphs [43]. Moreover, one can generalize it to b-edge-covers by requiring that every vertex is incident to at least b edges in the set. This approach yields an FPRAS to count b-edge-covers for $b \leq 2$ [43]. We note that FPTAS based on the correlation decay technique for counting edge covers for general graphs was known [54, 55]. However, it seems that the correlation decay approach have intrinsic difficulties for 2-edge-covers.

It is still open whether there exists an FPRAS for counting b-matchings for $b > 7$ or counting b-edge-covers for $b > 2$.

5.2 Fibonacci Functions

Correlation decay is another idea based on which one may design approximate counting algorithms. This approach has the advantage of yielding deterministic algorithms, namely FPTAS. Here we present FPTAS for a family of functions called Fibonacci Functions.

Fibonacci Functions by themselves are tractable, as they are $\mathcal{P}$-transformable (see Lemma 16). We extend the framework a bit by allowing edge weights. An edge-weighted Holant instance $\Omega = (G, \{f_v | v \in V\}, \{\lambda_e | e \in E\})$ is a tuple defined as follows. $G = (V, E)$ is a graph. f_v is a function with arity d_v: $\{0,1\}^{d_v} \to \mathbb{R}^+$, where d_v is the degree of v and $\mathbb{R}^+$ denotes non-negative real numbers. Edge weight λ_e is a mapping $\{0,1\} \to \mathbb{R}^+$. A configuration σ of edges is a mapping $E \to \{0,1\}$ and has a weight

$$w_\Omega(\sigma) = \prod_{e \in E} \lambda_e(\sigma(e)) \prod_{v \in V} f_v(\sigma\mid_{E(v)}),$$

where $E(v)$ denotes the set of incident edges of v. The counting problem on the instance Ω is to compute the partition function (or the Holant sum):

$$Z(\Omega) = \sum_{\sigma} \left(\prod_{e \in E} \lambda_e(\sigma(e)) \prod_{v \in V} f_v(\sigma\mid_{E(v)}) \right).$$

We use Holant$(\mathcal{F}, \Lambda)$ to denote the problem of computing the above quantity, where all functions are from $\mathcal{F}$ and all edge weights are from the set Λ. This version and its complexity are slightly different from the decision version and exactly counting as described in the previous two sections.

A Fibonacci function f is a symmetric function $[f_0, f_1, \ldots, f_k]$, satisfying that $f_i = cf_{i-1} + f_{i-2}$ for some constant c. For example, the parity function $[a, b, a, b, \ldots]$ is a special Fibonacci function with $c = 0$. If there is no edge weights (or equivalently all the weights are equal to 1) and all the node functions are Fibonacci functions with the same parameter c, we have a polynomial time algorithm to compute the partition function exactly [18]. If we allow edges to have non-trivial weights or different functions to have different parameters in Fibonacci gates, then the exact counting problem becomes #P-hard [19, 11]. Thus, it is interesting to study the problem in the approximation setting. Indeed, these edge-weighted Holant problems have connections with ferromagnetic 2-spin systems. For more details, see [56].

We use $\mathcal{F}_c^{p,q}$ to denote a subfamily of $\mathcal{F}_c$ such that $f_{i+1} \geq pf_i$ and $f_{i+1} \leq qf_i$ for all $i = 0, 1, \cdots, d-1$. When the upper bound q is not given, we simply write $\mathcal{F}_c^p$. We use $\mathcal{F}_{c_1,c_2}^{p,q}$ to denote $\bigcup_{c_1 \leq c \leq c_2} \mathcal{F}_c^{p,q}$. We use $\Lambda_{\lambda_1,\lambda_2}$ to denote the set of edge weights λ_e such that $\lambda_1 \leq \lambda_e \leq \lambda_2$.

Lu, Wang and Zhang [56] give the following algorithms.

▶ **Theorem 32.** *For any $c > 0$ and $p > 0$, there exists $\lambda_1(p, c) < 1$ and $\lambda_2(p, c) > 1$ such that there is an FPTAS for* $\text{Holant}(\mathcal{F}_c^p, \Lambda_{\lambda_1(p,c),\lambda_2(p,c)})$.

▶ **Theorem 33.** *Let $p > 0$. Then there is an FPTAS for* $\text{Holant}(\mathcal{F}_{1.17,+\infty}^p, \Lambda_{1,+\infty})$.

References

1 Mohsen Bayati, David Gamarnik, Dimitriy Katz, Chandra Nair, and Prasad Tetali. Simple deterministic approximation algorithms for counting matchings. In *Proceedings of STOC*, pages 122–127, 2007.

2 Ivona Bezáková and William A Rummler. Sampling edge covers in 3-regular graphs. In *Mathematical Foundations of Computer Science 2009*, pages 137–148. Springer, 2009.

3 Russ Bubley and Martin Dyer. Graph orientations with no sink and an approximation for a hard case of no. sat. Technical report, Association for Computing Machinery, New York, NY (United States), 1997.

4 Andrei A. Bulatov. A dichotomy theorem for constraint satisfaction problems on a 3-element set. *J. ACM*, 53(1):66–120, 2006. `doi:10.1145/1120582.1120584`.

5 Andrei A. Bulatov. The complexity of the counting constraint satisfaction problem. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *ICALP (1)*, volume 5125 of *Lecture Notes in Computer Science*, pages 646–661. Springer, 2008. `doi:10.1007/978-3-540-70575-8_53`.

6 Andrei A. Bulatov and Víctor Dalmau. Towards a dichotomy theorem for the counting constraint satisfaction problem. In *FOCS*, pages 562–571. IEEE Computer Society, 2003. URL: `http://csdl.computer.org/comp/proceedings/focs/2003/2040/00/20400562abs.htm`.

7 Jin-Yi Cai, Xi Chen, Richard J. Lipton, and Pinyan Lu. On tractable exponential sums. In *FAW*, pages 148–159. Springer Berlin Heidelberg, 2010.

8 Jin-Yi Cai, Xi Chen, and Pinyan Lu. Graph homomorphisms with complex values: A dichotomy theorem. In Samson Abramsky, Cyril Gavoille, Claude Kirchner, Friedhelm Meyer auf der Heide, and Paul G. Spirakis, editors, *ICALP (1)*, volume 6198 of *Lecture Notes in Computer Science*, pages 275–286. Springer, 2010. `doi:10.1007/978-3-642-14165-2_24`.

9 Jin-Yi Cai, Zhiguo Fu, Heng Guo, and Tyson Williams. A Holant dichotomy: Is the FKT algorithm universal? In *FOCS*, pages 249–260, 2015.

10 Jin-Yi Cai and Aaron Gorenstein. Matchgates revisited. *Theory Comput.*, 10(7):167–197, 2014.

11 Jin-Yi Cai, Heng Guo, and Tyson Williams. A complete dichotomy rises from the capture of vanishing signatures. In *STOC*, pages 249–260, 2013.
12 Jin-Yi Cai, Heng Guo, and Tyson Williams. The complexity of counting edge colorings and a dichotomy for some higher domain Holant problems. In *FOCS*, pages 601–610, 2014.
13 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.
14 Jin-Yi Cai and Michael Kowalczyk. Spin systems on k-regular graphs with complex edge functions. *Theor. Comput. Sci.*, 461:2–16, 2012.
15 Jin-Yi Cai and Michael Kowalczyk. Partition functions on k-regular graphs with $\{0,1\}$-vertex assignments and real edge functions. *Theor. Comput. Sci.*, 494(0):63–74, 2013.
16 Jin-Yi Cai and Pinyan Lu. Holographic algorithms: From art to science. *J. Comput. Syst. Sci.*, 77(1):41–61, 2011. doi:10.1016/j.jcss.2010.06.005.
17 Jin-Yi Cai and Pinyan Lu. Holographic algorithms: From art to science. *J. Comput. Syst. Sci.*, 77(1):41–61, 2011.
18 Jin-Yi Cai, Pinyan Lu, and Mingji Xia. Holographic algorithms by fibonacci gates and holographic reductions for hardness. In *FOCS'08: Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, Washington, DC, USA, 2008. IEEE Computer Society.
19 Jin-Yi Cai, Pinyan Lu, and Mingji Xia. Holant problems and counting CSP. In Michael Mitzenmacher, editor, *STOC*, pages 715–724. ACM, 2009. doi:10.1145/1536414.1536511.
20 Jin-Yi Cai, Pinyan Lu, and Mingji Xia. Computational complexity of Holant problems. *SIAM J. Comput.*, 40(4):1101–1132, 2011.
21 Jin-Yi Cai, Pinyan Lu, and Mingji Xia. A computational proof of complexity of some restricted counting problems. *Theor. Comput. Sci.*, 412(23):2468–2485, 2011.
22 Jin-Yi Cai, Pinyan Lu, and Mingji Xia. Dichotomy for Holant* problems of boolean domain. In *SODA'11: Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, 2011.
23 Jin-Yi Cai, Pinyan Lu, and Mingji Xia. Holographic reduction, interpolation and hardness. *Computational Complexity*, 21(4):573–604, 2012.
24 Jin-Yi Cai, Pinyan Lu, and Mingji Xia. Dichotomy for Holant* problems with domain size 3. In *SODA*, pages 1278–1295, 2013.
25 Jin-Yi Cai, Pinyan Lu, and Mingji Xia. The complexity of complex weighted Boolean #CSP. *J. Comput. System Sci.*, 80(1):217–236, 2014.
26 Gérard Cornuéjols. General factors of graphs. *Journal of Combinatorial Theory, Series B*, 45(2):185–198, 1988.
27 N. Creignou, S. Khanna, and M. Sudan. *Complexity classifications of boolean constraint satisfaction problems.* SIAM Monographs on Discrete Mathematics and Applications, 2001.
28 Nadia Creignou and Miki Hermann. Complexity of generalized satisfiability counting problems. *Inf. Comput.*, 125(1):1–12, 1996.
29 Victor Dalmau and Daniel K Ford. Generalized satisfiability with limited occurrences per variable: A study through delta-matroid parity. In *International Symposium on Mathematical Foundations of Computer Science*, pages 358–367. Springer, 2003.
30 C. T. J. Dodson and T. Poston. *Tensor Geometry.* Graduate Texts in Mathematics 130. Springer-Verlag, New York, 1991.
31 Zdeněk Dvořák and Martin Kupec. On planar boolean csp. In *International Colloquium on Automata, Languages, and Programming*, pages 432–443. Springer, 2015.
32 Martin E. Dyer, Leslie Ann Goldberg, Markus Jalsenius, and David Richerby. The complexity of approximating bounded-degree Boolean #CSP. *Inf. Comput.*, 220:1–14, 2012. doi:10.1016/j.ic.2011.12.007.

33 Martin E. Dyer, Leslie Ann Goldberg, and Mark Jerrum. The complexity of weighted boolean #CSP. *CoRR*, abs/0704.3683, 2007. URL: `http://arxiv.org/abs/0704.3683`.

34 Martin E. Dyer, Leslie Ann Goldberg, and Mike Paterson. On counting homomorphisms to directed acyclic graphs. *J. ACM*, 54(6), 2007. `doi:10.1145/1314690.1314691`.

35 Jack Edmonds. Maximum matching and a polyhedron with $0,1$ vertices. *J. of Res. the Nat. Bureau of Standards*, 69 B:125–130, 1965.

36 Jack Edmonds. Paths, trees, and flowers. *Canadian Journal of mathematics*, 17(3):449–467, 1965.

37 John Faben. The complexity of counting solutions to generalised satisfiability problems modulo k. *CoRR*, abs/0809.1836, 2008.

38 T. Feder and M. Y. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through Datalog and group theory. *SIAM Journal on Computing*, 28(1):57–104, 1999.

39 Tomás Feder. Fanout limitations on constraint systems. *Theoretical Computer Science*, 255(1):281–293, 2001.

40 M. Freedman, L. Lovász, and A. Schrijver. Reflection positivity, rank connectivity, and homomorphism of graphs. *J. AMS*, 20:37–51, 2007.

41 James F Geelen, Satoru Iwata, and Kazuo Murota. The linear delta-matroid parity problem. *Journal of Combinatorial Theory, Series B*, 88(2):377–398, 2003.

42 Leslie Ann 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`.

43 Lingxiao Huang, Pinyan Lu, and Chihao Zhang. Canonical paths for mcmc: from art to science. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 514–527. SIAM, 2016.

44 Sangxia Huang and Pinyan Lu. A dichotomy for real weighted Holant problems. In *IEEE Conference on Computational Complexity*, pages 96–106. IEEE Computer Society, 2012.

45 Mark Jerrum and Alistair Sinclair. Approximating the permanent. *SIAM J. Comput.*, 18(6):1149–1178, 1989. `doi:10.1137/0218077`.

46 Mark Jerrum and Alistair Sinclair. Polynomial-time approximation algorithms for the ising model. *SIAM Journal on Computing*, 22(5):1087–1116, 1993.

47 Mark Jerrum, Alistair Sinclair, and Eric Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *Journal of the ACM*, 51:671–697, July 2004. `doi:10.1145/1008731.1008738`.

48 P. W. Kasteleyn. The statistics of dimers on a lattice. *Physica*, 27:1209–1225, 1961.

49 P. W. Kasteleyn. Graph theory and crystal physics. In (F. Harary, editor, *Graph Theory and Theoretical Physics*, pages 43–110. Academic Press, London, 1967.

50 Alexandr Kazda, Vladimir Kolmogorov, and Michal Rolínek. Even delta-matroids and the complexity of planar boolean CSPs. *arXiv preprint arXiv:1602.03124*, 2016.

51 Michael Kowalczyk. Classification of a class of counting problems using holographic reductions. In Hung Q. Ngo, editor, *COCOON*, volume 5609 of *Lecture Notes in Computer Science*, pages 472–485. Springer, 2009. `doi:10.1007/978-3-642-02882-3_47`.

52 Michael Kowalczyk and Jin-Yi Cai. Holant problems for regular graphs with complex edge functions. *In the proceeding of STACS*, 2010.

53 Richard E. Ladner. On the structure of polynomial time reducibility. *J. ACM*, 22(1):155–171, 1975.

54 Chengyu Lin, Jingcheng Liu, and Pinyan Lu. A simple FPTAS for counting edge covers. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 341–348, 2014. `doi:10.1137/1.9781611973402.25`.

55 Jingcheng Liu, Pinyan Lu, and Chihao Zhang. FPTAS for counting weighted edge covers. In *Algorithms – ESA 2014 – 22th Annual European Symposium, Wroclaw, Poland, September 8-10, 2014. Proceedings*, pages 654–665, 2014.

56 Pinyan Lu, Menghui Wang, and Chihao Zhang. FPTAS for weighted Fibonacci gates and its applications. In *ICALP*, pages 787–799, 2014.

57 Colin McQuillan. Approximating holant problems by winding. *arXiv preprint arXiv:1301.2880*, 2013.

58 T. J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the tenth annual ACM symposium on Theory of computing*, page 226. ACM, 1978.

59 H. N. V. Temperley and M. E. Fisher. Dimer problem in statistical mechanics-an exact result. *Philosophical Magazine*, 6:1061–1063, 1961.

60 Leslie G. Valiant. The complexity of computing the permanent. *Theor. Comput. Sci.*, 8:189–201, 1979.

61 Leslie G. Valiant. The complexity of enumeration and reliability problems. *SIAM J. Comput.*, 8(3):410–421, 1979.

62 Leslie G. Valiant. Expressiveness of matchgates. *Theor. Comput. Sci.*, 289(1):457–471, 2002.

63 Leslie G. Valiant. Quantum circuits that can be simulated classically in polynomial time. *SIAM J. Comput.*, 31(4):1229–1254, 2002. URL: `http://epubs.siam.org/sam-bin/dbq/article/37702`.

64 Leslie G. Valiant. Accidental algorthims. In *FOCS'06: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 509–517, Washington, DC, USA, 2006. IEEE Computer Society. `doi:10.1109/FOCS.2006.7`.

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