



ELSEVIER

Contents lists available at ScienceDirect

Linear Algebra and its Applications

journal homepage: www.elsevier.com/locate/laa

Holographic algorithms by Fibonacci gates[☆]

Jin-Yi Cai^{a,*}, Pinyan Lu^{b,2}, Mingji Xia^c

^a Computer Sciences Department, University of Wisconsin – Madison, 1210 West Dayton Street, Madison, WI 53706, U.S.A.

^b Microsoft Research Asia, #999, Zi Xing Road, Min Hang District, Shanghai, 200241, P.R. China

^c Institute of Software, Chinese Academy of Sciences, #4 South Fourth Street, Zhong Guan Cun, Beijing 100190, P.R. China

ARTICLE INFO

Article history:

Available online xxxxx

Submitted by V. Mehrmann

Keywords:

Fibonacci gates
Holographic algorithm
Counting problems
Dichotomy theorem
Signature theory
Matchgates

ABSTRACT

We introduce Fibonacci gates as a polynomial time computable primitive, and develop a theory of holographic algorithms based on these gates. The Fibonacci gates play the role of matchgates in Valiant's theory (Valiant (2008) [19]). They give rise to polynomial time computable counting problems on general graphs, while matchgates mainly work over planar graphs only. We develop a signature theory and characterize all realizable signatures for Fibonacci gates. For bases of arbitrary dimensions we prove a basis collapse theorem. We apply this theory to give new polynomial time algorithms for certain counting problems. We also use this framework to prove that some slight variations of these counting problems are #P-hard. Holographic algorithms with Fibonacci gates prove to be useful as a general tool for classification results of counting problems (dichotomy theorems (Cai et al. (2009) [7])).

© 2011 Elsevier Inc. All rights reserved.

1. Introduction

Leslie Valiant [18, 19] introduced a marvelously original algorithmic design technique, called matchgate computations and holographic algorithms. This methodology has produced some exotic P-time algorithms for counting problems.

Roughly speaking, a counting problem in complexity theory is as follows: An input instance such as a graph is given, the question is to count the number of certain configurations or solutions. For

[☆] Some results in this paper have been reported in a preliminary conference paper [6].

* Corresponding author.

E-mail addresses: jyc@cs.wisc.edu (J.-Y. Cai), pinyanl@microsoft.com (P. Lu), [xmjljx@gmail.com](mailto:xmjlxj@gmail.com) (M. Xia).

¹ Supported by NSF CCF-0830488 and CCF-0511679.

² Work done in part while the author was a graduate student at Tsinghua University.

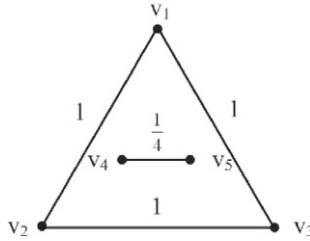


Fig. 1. A matchgate with signature $\frac{1}{4}(0, 1, 1, 0, 1, 0, 0, 1)$.

example, the counting problem can be to count the number of perfect matchings in an input graph. Frequently it is possible to translate this quantity as a pairing of a vector in a vector space (often of dimension exponential in the size of the input) with a vector in the dual space. This in general gives only an exponential time algorithm for the problem (computing this pairing takes exponential time). However, sometimes it is possible to evaluate this pairing much more efficiently (in time polynomial in the size of the input). *Holographic algorithms* try to achieve this speed-up of computation.

There are two main ingredients to Valiant’s theory. The first is to use planar perfect matchings and Pfaffians to encode and process information. The second is to create a weighted sum of exponential size with these perfect matchings in a “holographic mix”, and achieve exponential cancelations in the process. Ultimately it is reduced to the problem of counting the number of perfect matchings over planar graphs, and can be solved in polynomial time via the Fisher–Kasteleyn–Temperley (FKT) algorithm [12–14].

Holographic algorithms are found for a variety of problems, ranging from certain restricted Satisfiability type problems, to Vertex Cover, to other graph problems such as edge orientation and node/edge deletion. The full version of Satisfiability (SAT) is “Input: a Boolean formula; Output: the number of satisfying assignments”. The full version of Vertex Cover (Vc) is “Input: a graph $G(V, E)$; Output: the number of $S \subseteq V$ such that every $e \in E$ is incident to some $v \in S$ ”. The P-time algorithms only solve these problems on restricted classes of inputs. Nevertheless, the restricted versions were not known previously to be in P, and some minor variations are known to be NP-hard. For example, let #PI-Rtw-Mon-3CNF denote the problem of counting the number of satisfying assignments, where the input formula is in 3-Conjunctive-Normal-Form, each variable appears twice (called read-twice) and without negation (called monotone), and furthermore the incidence graph of this formula (which variable appears in which clause) is planar. This restricted SAT problem is still #P-complete, which means that it is as hard as the unrestricted SAT problem. Moreover, the restricted problem mod 2, #₂PI-Rtw-Mon-3CNF, which decides whether the number of satisfying assignments is even or odd, is NP-hard. Using holographic algorithms Valiant [20] showed that surprisingly, counting modulo 7 for this restricted SAT problem, #₇PI-Rtw-Mon-3CNF, is in P.

We will illustrate with an example the idea of creating a “holographic mix” by values of perfect matchings. Fig. 1 shows a finite planar weighted graph Γ , called a matchgate, with 3 external nodes v_1, v_2, v_3 . For any $b_1 b_2 b_3 \in \{0, 1\}^3$, we can remove a subset $S = \{v_i | b_i = 1\}$ and all incident edges; we denote the remaining graph $\Gamma - S$. Now we compute the “perfect matching polynomial” $\sum_M \prod_{e \in M} w(e)$, where M ranges over all perfect matchings of $\Gamma - S$, and $w(e)$ is the weight of the edge e . This value will be denoted by $\Gamma_{b_1 b_2 b_3}$. Putting them together we get a “signature” of the matchgate $(\Gamma_{b_1 b_2 b_3}) = \frac{1}{4}(0, 1, 1, 0, 1, 0, 0, 1)_{b_1 b_2 b_3 \in \{0, 1\}^3}$. This represents a function on 3 Boolean variables, which takes value $1/4$ if there are exactly one or three variables set to 1, and it takes value 0 otherwise. (Note that the perfect matching polynomial counts the number of perfect matchings if all edge weights $w(e) = 1$.)

Now suppose we wish to represent the Boolean OR function of 3 inputs. In the framework of holographic algorithms this means that we want the “signature” $(0, 1, 1, 1, 1, 1, 1, 1)$ indexed by three bits $b_1 b_2 b_3 \in \{0, 1\}^3$. This signature is a vector of dimension 8, representing the truth table of the OR function of 3 inputs. While this signature is not directly expressible by perfect matchings, it is possible to express it indirectly as a linear “superposition” of perfect matchings using the matchgate Γ . We

choose two linearly independent basis vectors $\beta = \left[\begin{pmatrix} 1 + \omega \\ 1 - \omega \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right]$, where $\omega = e^{2\pi i/3}$. In the tensor product space spanned by $\beta^{\otimes 3}$, we can represent the OR signature $(0, 1, 1, 1, 1, 1, 1, 1)$ by perfect matchings as follows: We have

$$\left(\begin{bmatrix} 1 + \omega & 1 \\ 1 - \omega & 1 \end{bmatrix}^{-1} \right)^{\otimes 3} = \frac{1}{8} \begin{pmatrix} 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ -1 + \omega & 1 + \omega & 1 - \omega & -1 - \omega & 1 - \omega & -1 - \omega & -1 + \omega & 1 + \omega \\ -1 + \omega & 1 - \omega & 1 + \omega & -1 - \omega & 1 - \omega & -1 + \omega & -1 - \omega & 1 + \omega \\ -3\omega & -2 - \omega & -2 - \omega & \omega & 3\omega & 2 + \omega & 2 + \omega & -\omega \\ -1 + \omega & 1 - \omega & 1 - \omega & -1 + \omega & 1 + \omega & -1 - \omega & -1 - \omega & 1 + \omega \\ -3\omega & -2 - \omega & 3\omega & 2 + \omega & -2 - \omega & \omega & 2 + \omega & -\omega \\ -3\omega & 3\omega & -2 - \omega & 2 + \omega & -2 - \omega & 2 + \omega & \omega & -\omega \\ 3 + 6\omega & 3 & 3 & -1 - 2\omega & 3 & -1 - 2\omega & -1 - 2\omega & -1 \end{pmatrix},$$

and therefore (adding up the last 7 rows)

$$(0, 1, 1, 1, 1, 1, 1, 1) \left(\begin{bmatrix} 1 + \omega & 1 \\ 1 - \omega & 1 \end{bmatrix}^{-1} \right)^{\otimes 3} = \frac{1}{4} (0, 1, 1, 0, 1, 0, 0, 1) = (\Gamma_{b_1 b_2 b_3}).$$

It follows that $(0, 1, 1, 1, 1, 1, 1, 1) = (\Gamma_{b_1 b_2 b_3}) \beta^{\otimes 3}$. In this way each logical value 0 or 1 in the Boolean OR function $(0, 1, 1, 1, 1, 1, 1, 1)$ is expressed as a “superposition”, or a linear combination, of perfect matching values in $(\Gamma_{b_1 b_2 b_3})$.

Holographic algorithms with matchgates can only be applied to planar problems, since they ultimately use Fisher–Kasteleyn–Temperley (FKT) algorithm for planar perfect matchings.³ In this paper we go beyond matchgates, and extend the reach of holographic algorithms to general graphs, without the planarity condition. In Section 5 we will show how to solve in polynomial time a certain edge coloring problem. Such problems should be thought of as members of a class of general counting problems on graphs in the following framework. Let us assign a Boolean function F_v at each node v . Edges are considered as Boolean variables. We want to compute $\sum_{\sigma: E \rightarrow \{0,1\}} \prod_v F_v(\sigma|_{E(v)})$, where the evaluation $F_v(\sigma|_{E(v)})$ is on the incident edges of v . E.g., if we take F_v to be the AT-MOST-ONE function, then this is the counting problem of all matchings. Perfect matching corresponds to the EXACT-ONE function.

We will consider symmetric functions on Boolean variables. The functions can take arbitrary real or complex values. A symmetric function on d Boolean variables is denoted by $[f_0, f_1, \dots, f_d]$, where f_k is the value of the function where the number of input variables assigned to one is k . E.g., the EXACT-ONE function of arity 3 is $[0, 1, 0, 0]$. If we consider all 3-regular graphs G , then counting perfect matchings can be denoted as $\#[0, 1, 0, 0]$; this corresponds to the problem of evaluating the quantity $\sum_{\sigma: E \rightarrow \{0,1\}} \prod_v F_v(\sigma|_{E(v)})$ where every function F_v is the EXACT-ONE function $[0, 1, 0, 0]$. It can be shown that the class of 2–3 regular graphs (bipartite graphs with degree 2 and 3 on the left and right respectively) can simulate all 3-regular graphs. In terms of 2–3 regular graphs $G = (U, V, E)$, counting PERFECT MATCHING is $\#[1, 0, 1][[0, 1, 0, 0]$, where all vertices of U are given the binary EQUALITY function $[1, 0, 1]$ and all vertices of V are given $[0, 1, 0, 0]$. Counting VERTEX COVER for 3-regular graphs is $\#[0, 1, 1][[1, 0, 0, 1]$. The problem mentioned earlier #Rtw-Mon-3CNF is $\#[1, 0, 1][[0, 1, 1, 1]$.

³ A more intrinsic understanding has been achieved lately [8].

Expressed in this language, Valiant [19] showed that the problem $\#[1, 0, 1][0, 1, 1, 0]$ can be solved in P with holographic algorithms based on matchgates over planar graphs. This problem counts the number of *valid* 2-colorings of edges such that at each vertex it is not monochromatic. In this paper we will show that the problem $\#[1, 0, -1][0, 1, 1, 0]$ can be solved in P with holographic algorithms based on Fibonacci gates. This can be thought of as counting the number of *even minus odd* valid 2-colorings, where a Black/White coloring is even (resp. odd) if the total number of Black edges is even (resp. odd). Of course the number of even *plus* odd valid 2-colorings is just the number of *all* valid 2-colorings, thus Valiant's problem. We will further show that Valiant's problem $\#[1, 0, 1][0, 1, 1, 0]$ is $\#P$ -complete for non-planar graphs. We note that there is no planarity restriction for Fibonacci gates.

Our main contribution in this paper is to extend the framework of holographic algorithms by introducing Fibonacci gates. They play a similar role as matchgates in Valiant's theory, and constitute another class of P-time computable primitives. We develop a corresponding theory of holographic algorithms based on these Fibonacci gates. To understand what these Fibonacci gates can do in a computational setting we must understand its signature theory. We derive a complete characterization of all realizable signatures of Fibonacci gates under any choice of bases in \mathbf{GL}_2 . We then apply this theory to some non-trivial counting problems to obtain P-time algorithms. We note that the signatures used in these holographic algorithms are not realizable by matchgates under any basis transformation, and therefore we have properly extended the reach of holographic algorithms. (However, Fibonacci gates do not encompass matchgates; their applicabilities are incomparable.)

We believe that holographic algorithms based on Fibonacci gates provide a unifying perspective on such counting problems (when not restricting to planar graphs). In [7] (see also [6]) we show that Fibonacci gates can be used to prove dichotomy theorems, which state for a wide class of problems, every single member is either in P or $\#P$ -hard. For example, every problem in the class of $\#[a_0, a_1, a_2][b_0, b_1, b_2, b_3]$ with Boolean signatures is proved either $\#P$ -complete (using holographic reductions and interpolations) or in P, where all proofs (other than trivial ones) of membership in P use *holographic algorithms* based on either Fibonacci gates or matchgates over planar graphs. Problems which can be proved [6] to be in P by Fibonacci gates include: $\#[1, 1, 0][1, 1, 0, 1]$, $\#[0, 1, 1][1, 0, 1, 1]$, $\#[1, 0, 1][0, 1, 0, 1]$, $\#[1, 0, 1][1, 0, 1, 0]$, $\#[1, 0, 1][1, 0, 1, 1]$, $\#[1, 0, 1][1, 1, 0, 1]$, $\#[0, 1, 0][0, 1, 0, 1]$, $\#[0, 1, 0][1, 0, 1, 0]$, etc. While it is feasible to derive for each problem a separate algorithm, based on Fibonacci gates introduced here, the derivation becomes automatic: e.g., for the first problem listed above, $\#[1, 1, 0][1, 1, 0, 1]$, the verification is simply that both signatures satisfy $f_{i+2} = (-1)f_{i+1} + f_i$, for $i \geq 0$. More details can be found in Section 8.

Next we consider the signature theory under more general bases. Here we prove a general basis collapse theorem, showing that, for holographic algorithms based on Fibonacci gates, any holographic algorithm using a two-vector basis in arbitrary dimension can be simulated by a two-vector basis in dimension 2.

Finally we prove some hardness results. We prove a minor variation of the problem we solved in polynomial time using holographic algorithms based on Fibonacci gates is $\#P$ -hard. Our framework of counting problems can discuss signatures which (unlike matchgates and Fibonacci gates) do not necessarily correspond to P-time computable primitives. We use a technique called interpolation to prove this hardness. In particular we make use of a technical lemma due to Vadhan [15]. An interesting feature of these reductions is that they provide some natural examples where the Karp-type mapping (or many-one) reductions seem to be weaker than the oracle query Cook-type reductions.

This paper is organized as follows: In Section 2 we define the framework of counting problems based on signatures. In Section 3 we introduce Fibonacci gates, and prove that they form a P-time computable primitive. In Section 4 we develop the signature theory for these Fibonacci gates. In Section 5, we consider some counting problems solvable in polynomial time using the theory just developed. In Section 6 we prove hardness results using interpolation reductions. In Section 7 we further develop the signature theory and prove a general basis collapse theorem.

2. Counting problems and holographic reductions

Many counting problems can be formulated in the following framework.

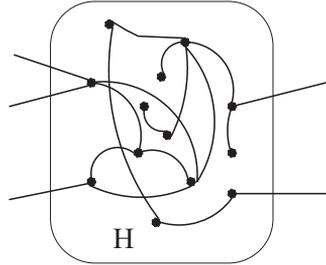


Fig. 2. An example of an \mathcal{F} -gate with five dangling edges.

A signature grid $\Omega = (H, \mathcal{F})$ is a tuple, where $H = (V, E)$ is a graph, and \mathcal{F} are functions assigned to vertices of H . We consider every edge $e \in E$ as a Boolean variable, and every vertex $v \in V$ is assigned a function F_v from \mathcal{F} , where $F_v : \{0, 1\}^{d(v)} \rightarrow \mathbf{F}$, for some field \mathbf{F} and $d(v)$ is the degree of v . (In this paper our field \mathbf{F} will be either \mathbb{R} or \mathbb{C} .) The $d(v)$ input variables of F_v are in 1–1 correspondence with the incident edges of v . Given an assignment $\sigma : E \rightarrow \{0, 1\}$ of all the edges, we have a valuation at each vertex v , which is the value of F_v on σ restricted to the edges incident to v . The value of Ω (or we simply say the value of H) under σ is the product of F_v on σ over all $v \in V$, and the value of Ω is the sum over all assignments σ . The counting problem on Ω is to compute this value.

Formally let $E = \{1, 2, \dots, m\}$, the edges connected to vertex v are denoted by $i_1^v, i_2^v, \dots, i_{d(v)}^v$. Then we define

$$\text{Holant}_{\Omega} = \sum_{x_1 x_2 \dots x_m \in \{0, 1\}^m} \prod_{v \in V} F_v(x_{i_1^v}, x_{i_2^v}, \dots, x_{i_{d(v)}^v}).$$

We also write Holant_H when there is no confusion. We can view each function F_v as a vector in $\mathbf{F}^{2^{d(v)}}$, or a tensor in $(\mathbf{F}^2)^{\otimes d(v)}$: We take the basis $\mathbf{e}_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $\mathbf{e}_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ of \mathbf{F}^2 corresponding to 0 and 1. Then in

$(\mathbf{F}^2)^{\otimes d(v)}$ with respect to the basis $\mathbf{e}_{b_1} \otimes \mathbf{e}_{b_2} \otimes \dots \otimes \mathbf{e}_{b_{d(v)}}$, where $b_i = 0, 1$, the function F_v corresponds to the vector whose coefficient at $\mathbf{e}_{b_1} \otimes \mathbf{e}_{b_2} \otimes \dots \otimes \mathbf{e}_{b_{d(v)}}$ is the function value $F(b_1, b_2, \dots, b_{d(v)})$. This vector is called a signature.

Many important counting problems can be viewed as computing Holant_{Ω} for appropriate signatures at each vertex. Denote by $\text{wt}(\sigma)$ the Hamming weight $\text{wt}(\sigma) = |\sigma^{-1}(1)|$ (or if we represent σ by a 0–1 sequence, then $\text{wt}(\sigma)$ is the number of 1’s). Let $F_v(\sigma) = 1$ if $\text{wt}(\sigma) = 1$, and $F_v(\sigma) = 0$ otherwise. Then Holant_{Ω} counts the number of perfect matchings in H . Suppose instead $F_v(\sigma) = 1$ if $\text{wt}(\sigma) \leq 1$, and $F_v(\sigma) = 0$ otherwise, then we are counting all (not necessarily perfect) matchings of the graph H . Many counting problems not directly defined in terms of graphs can also be formulated as holant problems. For example, the well-known #3SAT problem asks for the number of satisfying assignments in a given Boolean formula φ on Boolean variables x_1, \dots, x_n . Here φ has the form $\bigwedge_j C_j$, where each C_j (called a clause) is a disjunction (Boolean OR) of three literals (variables or their negations). For any instance φ , we can draw a bipartite graph, where a variable node x_i is connected to a clause node C_j iff x_i (or its negation) appears in the clause C_j . Then, to count #SAT, for each variable x_i we assign the EQUALITY function, and for each clause C_j we assign the truth table of C_j .

Now we define the notion of an \mathcal{F} -gate $\Gamma = (H, \mathcal{F})$, where $H = (V, E, D)$ is a graph with some dangling edges D . (See Fig. 2 for one example.) Other than these dangling edges, an \mathcal{F} -gate is the same as a signature grid. The role of dangling edges is similar to that of external nodes in Valiant’s notion [19], however we allow more than one dangling edges for a node. Consider a graph with dangling edges $H = (V, E, D)$, where each node is assigned a function in \mathcal{F} (we do not consider “dangling” leaf nodes at the end of a dangling edge among these), E are the regular edges, denoted as $1, 2, \dots, m$, and D are the dangling edges, denoted as $m + 1, m + 2, \dots, m + n$. Then we can define a function for this \mathcal{F} -gate $\Gamma = (H, \mathcal{F})$,

$$\Gamma(y_1, y_2, \dots, y_n) = \sum_{(x_1, x_2, \dots, x_m) \in \{0, 1\}^m} H(x_1, x_2, \dots, x_m, y_1, y_2, \dots, y_n),$$

where $(y_1, y_2, \dots, y_n) \in \{0, 1\}^n$ denotes an assignment on the dangling edges and $H(x_1, x_2, \dots, x_m, y_1, y_2, \dots, y_n)$ denotes the value of the signature grid on an assignment of all edges. We will also call this function $\Gamma(y_1, y_2, \dots, y_n)$ the signature of the \mathcal{F} -gate Γ . An \mathcal{F} -gate can be used in a signature grid as if it is just a single node with the particular signature. We note that even for a very simple signature set \mathcal{F} , the signatures for all \mathcal{F} -gates can be quite complicated and expressive. Matchgate signatures are an example.

Suppose $\Gamma_1 = (H_1, \mathcal{F}_1)$ and $\Gamma_2 = (H_2, \mathcal{F}_2)$ are \mathcal{F} -gates with dangling edge sets D_1 and D_2 . We can form a new \mathcal{F} -gate $\Gamma = (H, \mathcal{F})$ by merging some of their dangling edges to form regular edges. (See Fig. 3 for one example.) Suppose $|D_1| = k + p$, $|D_2| = k + q$ and the first k dangling edges of D_1 are merged with the first k dangling edges of D_2 correspondingly. Then

$$\begin{aligned} &\Gamma(y_1, y_2, \dots, y_p, z_1, z_2, \dots, z_q) \\ &= \sum_{(x_1, \dots, x_k) \in \{0, 1\}^k} \Gamma_1(x_1, \dots, x_k, y_1, y_2, \dots, y_p) \Gamma_2(x_1, \dots, x_k, z_1, z_2, \dots, z_q). \end{aligned}$$

We remark that a single node with a number of dangling edges can be viewed as the simplest \mathcal{F} -gate; also the whole signature grid $\Omega = (H, \mathcal{F})$ can be viewed as an \mathcal{F} -gate with zero dangling edges, and its Holant_Ω is exactly its signature (here it is only a single value).

A signature is called symmetric, if each signature entry only depends on the Hamming weight of the input. The signatures we defined above for matching or perfect matching or Boolean OR all have this property. We use a more compact notation $[f_0, f_1, \dots, f_n]$ to denote a symmetric signature on n inputs, where f_i is the value on inputs of Hamming weight i . In terms of the tensor expression, f_i is the coefficient in $\mathbf{e}_0^{n-i} \otimes \mathbf{e}_1^i$, and is the same coefficient in $\mathbf{e}_{b_1} \otimes \mathbf{e}_{b_2} \otimes \dots \otimes \mathbf{e}_{b_n}$ as long as $\text{wt}(b_1 b_2 \dots b_n) = i$.

A counting problem is now generally defined to be the computation of Holant_Ω for some signature grids Ω from some family of graphs and signatures on its vertices.

We will mostly consider bipartite graphs $H = (V_1, V_2, E)$ here. For any general graph, we can make it bipartite by adding an additional vertex on each edge. The signature for each new vertex is the EQUALITY function $(1, 0, 0, 1)$ on 2 inputs. Note that $(1, 0, 0, 1) = (1, 0)^{\otimes 2} + (0, 1)^{\otimes 2}$. In symmetric notation, it is $[1, 0, 1]$.

We use $\#\mathcal{H} : \mathcal{G}|\mathcal{R}$ to denote all the counting problems, expressed as holant problems on bipartite graphs $H = (V_1, V_2, E)$, where the graph H is from the graph family \mathcal{H} , \mathcal{G} and \mathcal{R} are sets of signatures, and each signature for a vertex in V_1 or V_2 is from \mathcal{G} or \mathcal{R} , respectively. If \mathcal{H} consists of all bipartite graphs, or the restriction is clear, we will simply use $\#\mathcal{G}|\mathcal{R}$ to denote the holant problem. E.g., when \mathcal{G} consists of a single signature of arity two: $\mathcal{G} = \{[x_0, x_1, x_2]\}$, and \mathcal{R} consists of a single signature

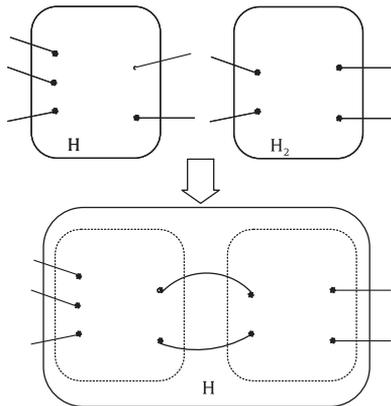


Fig. 3. Two \mathcal{F} -gates compose into one.

of arity three: $\mathcal{R} = \{[y_0, y_1, y_2, y_3]\}$, we will simply write $\#[x_0, x_1, x_2][y_0, y_1, y_2, y_3]$ to denote the counting problem where the class of graphs consists of all 2–3 regular graphs, and every node on the left (resp. right) is attached the signature $[x_0, x_1, x_2]$ (resp. $[y_0, y_1, y_2, y_3]$). An input instance of the holant problem is a signature grid and is denoted as $\Omega = (H, \mathcal{G}|\mathcal{R})$. Signatures in \mathcal{G} are called generators, which are denoted by column vectors (or contravariant tensors); signatures in \mathcal{R} are called recognizers, which are denoted by row vectors (or covariant tensors) [9].

Much of the power of holographic algorithms is derived from carefully designed cancelations in tensor spaces, as illustrated in Section 1. Let $T = [\mathbf{n}, \mathbf{p}]$, where \mathbf{n} and \mathbf{p} are two linearly independent vectors in the vector space \mathbf{F}^{2^k} of dimension 2^k over \mathbf{F} . When $k = 1$, such a two-vector set is a linear basis of \mathbf{F}^2 in the usual sense. For most of this paper, $k = 1$. When $k > 1$, such a two-vector set has also been called a basis in [19], and it is called a basis of size k ; we will follow this usage, although it is not a linear basis in the usual sense. Then we can define a holographic reduction as follows. Suppose $\#\mathcal{H} : \mathcal{G}|\mathcal{R}$ and $\#\mathcal{H}' : \mathcal{G}'|\mathcal{R}'$ are two holant problems defined for the same family of graphs. Suppose there exists a basis T of size k and a polynomial time computable mapping σ , such that: for every generator $G \in \mathcal{G}$ of arity g , $\sigma(G)$ is an \mathcal{F} -gate with gk dangling edges (not necessary a single node) and with signature $G' \in \mathcal{G}'$, for every recognizer $R \in \mathcal{R}$ of arity r , $\sigma(R)$ is an \mathcal{F} -gate with rk dangling edges and with signature $R' \in \mathcal{R}'$, and furthermore $G' = T^{\otimes g}G$ and $R = R'T^{\otimes r}$. Then we say that there is a holographic reduction from $\#\mathcal{H} : \mathcal{G}|\mathcal{R}$ to $\#\mathcal{H}' : \mathcal{G}'|\mathcal{R}'$. (Notice the reversal of directions when the transformation $T^{\otimes n}$ is applied. This is the meaning of *contravariance* and *covariance*.)

Theorem 2.1 (Valiant's holant theorem). *Suppose in a holographic reduction a signature grid Ω is mapped to a signature grid Ω' , then*

$$\text{Holant}_{\Omega} = \text{Holant}_{\Omega'}.$$

The proof of this theorem follows from general principles of contravariant and covariant tensors and their contractions [2]. In particular, if there is a holographic reduction from $\#\mathcal{H} : \mathcal{G}|\mathcal{R}$ to $\#\mathcal{H}' : \mathcal{G}'|\mathcal{R}'$, and there is a P-time algorithm for $\#\mathcal{H}' : \mathcal{G}'|\mathcal{R}'$, then there is a P-time algorithm for $\#\mathcal{H} : \mathcal{G}|\mathcal{R}$. Similarly, if the first holant problem is #P-hard, then so is the second.

In holographic algorithms proposed by Valiant [19], one reduces a given problem to the planar perfect matching problem, which has the FKT algorithm. Using that, many interesting problems are proved to be polynomial time solvable [19,20,1,3]. But in the framework of holographic reductions, we can reduce a given problem to any other holant problem, which is polynomial time solvable. In the next section, we will introduce another family of polynomial time solvable holant problems.

3. Fibonacci gates

In this section, we introduce a new set of signatures called Fibonacci gates. Then we give a polynomial time algorithm for holant problems on these signatures. A first form of this idea appeared in [21].

Let $\{f_k\}_{k=0}^n$ be a sequence, satisfying $f_{k+2} = f_{k+1} + f_k$ for all $k = 0, 1, \dots, n-2$. For any initial values f_0 and f_1 , such a sequence will be called a Fibonacci sequence. For any arity n a Fibonacci sequence defines a symmetric signature $F = [f_0, f_1, \dots, f_n]$. (We consider any symmetric signature of arity less than 2 to be Fibonacci as well.) This defines a function on n Boolean inputs $F : \{0, 1\}^n \rightarrow \mathbf{F}$ such that $F(\sigma) = f_{\text{wt}(\sigma)}$, for all $\sigma \in \{0, 1\}^n$. We call such functions Fibonacci gates or Fibonacci signatures. We use \mathcal{F} to denote all the Fibonacci signatures.

Theorem 3.1. *For any graph H , the holant problem $\#(H, \mathcal{F})$ can be computed in polynomial time.*

Proof. If H_1, H_2, \dots, H_l are all the connected components of a graph H , then $\text{Holant}_H = \prod_{i=1}^l \text{Holant}_{H_i}$. So we only need to consider connected graphs as inputs.

Suppose H has n nodes and m edges. First we cut all the edges in H . A node with degree d can be viewed as an \mathcal{F} -gate with d dangling edges. Now step by step we connect two dangling edges into

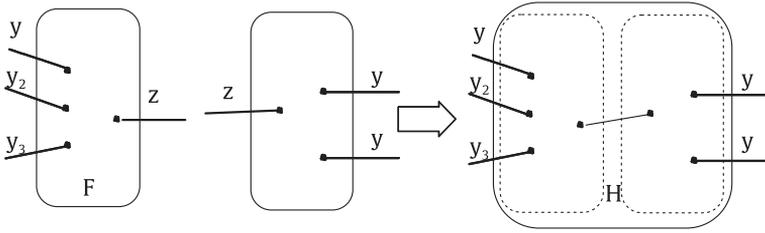


Fig. 4. First operation.

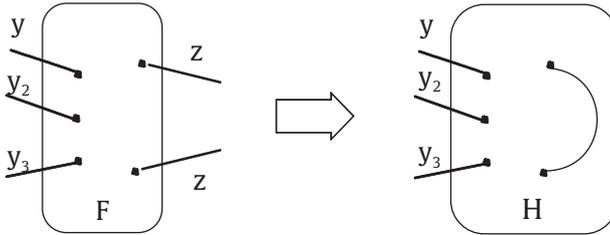


Fig. 5. Second operation.

one regular edge in the original graph, until we recover H after m steps. Our plan is to prove that all the intermediate \mathcal{F} -gates still have Fibonacci signatures and at every step we can compute the intermediate signature (we only need to compute the first two values of the signature) in polynomial time. Finally we get H , an \mathcal{F} -gate without any dangling edges, its signature (only one value) is the holant we want to compute. To carry out this plan, we only need to prove that it is true for one single step. There are two cases, depending on whether the two dangling edges to be connected are in the same component or not. These two operations are illustrated in Figs. 4 and 5.

In the first case, the two dangling edges belong to two components before their merging (Fig. 4). Let F have dangling edges y_1, \dots, y_s, z and G have dangling edges $y_{s+1}, \dots, y_{s+t}, z'$. After merging z with z' , we have a new gate H with dangling edges $y_1, \dots, y_s, \dots, y_{s+t}$. Inductively the signatures of gates F and G are both Fibonacci functions. We show that the resulting gate H also has a Fibonacci signature.

We first prove H is symmetric. We only need to show that the value of H is not changed if the value of two inputs are exchanged. Because F and G are symmetric, if both inputs are from $\{y_1, \dots, y_s\}$ or from $\{y_{s+1}, \dots, y_{s+t}\}$, the value of H is clearly not changed. Suppose one input is from $\{y_1, \dots, y_s\}$ and the other is from $\{y_{s+1}, \dots, y_{s+t}\}$. By symmetry of F and G we may assume these two inputs are y_1 and y_{s+1} . Thus we will fix an arbitrary assignment for $y_2, \dots, y_s, y_{s+2}, \dots, y_{s+t}$, and we want to show $H(0, y_2, \dots, y_s, 1, y_{s+2}, \dots, y_{s+t}) = H(1, y_2, \dots, y_s, 0, y_{s+2}, \dots, y_{s+t})$.

We can suppress the fixed $y_2, \dots, y_s, y_{s+2}, \dots, y_{s+t}$ and denote $F_{y_1z} = F(y_1, y_2, \dots, y_s, z)$, $G_{y_{s+1}z} = G(y_{s+1}, y_{s+2}, \dots, y_{s+t}, z)$, and $H_{y_1y_{s+1}} = H(y_1, \dots, y_s, y_{s+1}, \dots, y_{s+t})$. Then by the definition of holant, $H_{ab} = F_{a0}G_{b0} + F_{a1}G_{b1}$, for $a, b \in \{0, 1\}$. In particular, $H_{01} = F_{00}G_{10} + F_{01}G_{11}$, and $H_{10} = F_{10}G_{00} + F_{11}G_{01}$.

Because F and G are Fibonacci functions, $F_{11} = F_{01} + F_{00}$ and $G_{11} = G_{01} + G_{00}$. We have

$$H_{01} = F_{00}G_{10} + F_{01}G_{01} + F_{01}G_{00}, \quad \text{and} \quad H_{10} = F_{10}G_{00} + F_{01}G_{01} + F_{00}G_{01}.$$

By symmetry of F and G , $H_{01} = H_{10}$.

Now we show that $H(y_1, \dots, y_{s+t})$ is also a Fibonacci function. Since we have proved that H is symmetric, we can choose any two inputs to prove it being Fibonacci. Again, we choose y_1 and y_{s+1} . (This assumes that y_1 and y_{s+1} exist. If after the merging, either the part from F or G has no dangling edges left, the proof is just as easy. This also includes the case when the arity of H is less than 2.) For any fixed values of all the other inputs, we have $H_{00} = F_{00}G_{00} + F_{01}G_{01}$, $H_{01} = F_{00}G_{10} + F_{01}G_{11}$, and

$H_{11} = F_{10}G_{10} + F_{11}G_{11}$. Now using the fact that both F and G are Fibonacci functions, it is easy to show that $H_{00} + H_{01} = H_{11}$.

If the first two terms of the signatures of F and G are f_0, f_1 and g_0, g_1 respectively, then the first two terms of the signature H can be easily computed as following: $h_0 = f_0g_0 + f_1g_1$ and $h_1 = f_1g_0 + f_2g_1 = f_1g_0 + (f_0 + f_1)g_1$.

Next we consider the second case, where the two dangling edges to be merged are in the same component (Fig. 5). Obviously, the signature for the new gate H is also symmetric. If $F = [f_0, f_1, \dots, f_n]$ is the Fibonacci signature before the merging operation, then the signature after the merging operation is $H = [f_0 + f_2, f_1 + f_3, \dots, f_{n-2} + f_n]$. It follows that H is also Fibonacci and we have already computed its signature. \square

Before moving on, we comment that we did not stop at two-term recurrence by choice. If one starts with a higher order recurrence, the above proof no longer works. Moreover, trivially one can interpolate any symmetric signature $[f_0, f_1, f_2, f_3]$ of arity 3 by a third order recurrence. Later in Section 6 we will prove #P-hardness for some of these problems. Assuming #P \neq P, which is weaker than NP \neq P, this shows that there is an essential difficulty extending these two-term recurrence to higher order ones.

Now we can use Fibonacci signatures to solve new problems. The goal is to use these Fibonacci gates in conjunction with holographic reductions to give P-time algorithms for various counting problems. For that we need the following

Definition 3.1. A generator G (resp. recognizer R) with arity n is realizable as a Fibonacci gate on basis T iff there exists a Fibonacci signature F such that $F^T = T^{\otimes n}G$ (resp. $R = FT^{\otimes n}$).

4. Realizability

In this section, we characterize all holant problems which can be solved by holographic algorithms with Fibonacci gates. Here we only consider bases of size 1, but the result is actually valid for any two-vector basis of size k , for any k . In Section 7, we will prove a basis collapse theorem, which shows that any holographic algorithm with Fibonacci gates using a basis of size k can be simulated on a basis of size 1.

Let ϕ (the golden ratio) and $\bar{\phi}$ be the two roots of $X^2 - X - 1 = 0$. Then for any Fibonacci sequence $\{f_i\}_{i=0}^n$, there exist two numbers A and B such that $f_i = A\phi^i + B\bar{\phi}^i$, where $i = 0, 1, \dots, n$. It follows that for any Fibonacci signature F , there exist two numbers A and B such that $F = A(1, \phi)^{\otimes n} + B(1, \bar{\phi})^{\otimes n}$. (The notation $(1, \phi)^{\otimes n}$ denotes the vector of length 2^n , where any entry indexed by an n -bit 0–1 sequence of Hamming weight k is ϕ^k . The symmetric signature notation for $(1, \phi)^{\otimes n}$ is $[1, \phi, \phi^2, \dots, \phi^n]$.)

Let $T = \begin{bmatrix} n_0 & p_0 \\ n_1 & p_1 \end{bmatrix} \in \mathbf{GL}_2$. Then for any realizable recognizer signature R , we have

$$\begin{aligned} R &= (A(1, \phi)^{\otimes n} + B(1, \bar{\phi})^{\otimes n})T^{\otimes n} \\ &= A((1, \phi)T)^{\otimes n} + B((1, \bar{\phi})T)^{\otimes n} \\ &= A(n_0 + n_1\phi, p_0 + p_1\phi)^{\otimes n} + B(n_0 + n_1\bar{\phi}, p_0 + p_1\bar{\phi})^{\otimes n}. \end{aligned}$$

So R is also symmetric, and writing in symmetric notation $R = [x_0, x_1, \dots, x_n]$, we have

$$x_i = A(n_0 + n_1\phi)^{n-i}(p_0 + p_1\phi)^i + B(n_0 + n_1\bar{\phi})^{n-i}(p_0 + p_1\bar{\phi})^i. \tag{1}$$

A matrix $T \in \mathbf{GL}_2$ defines a Möbius function $\ell_T(z) = \frac{p_1z+p_0}{n_1z+n_0}$, then $x_i = A'(\ell_T(\phi))^i + B'(\ell_T(\bar{\phi}))^i$, for some constants $A' = A(n_0 + n_1\phi)^n$ and $B' = B(n_0 + n_1\bar{\phi})^n$.

When we replace T by $(T^{-1})^T$, all results for recognizers work for generators. In particular, if $G = [x_0, x_1, \dots, x_n]^T$ is realizable as a Fibonacci gate on a basis T , then

$$x_i = A(p_1 - p_0\phi)^{n-i}(-n_1 + n_0\phi)^i + B(p_1 - p_0\bar{\phi})^{n-i}(-n_1 + n_0\bar{\phi})^i. \tag{2}$$

Theorem 4.1. A symmetric signature $[x_0, x_1, \dots, x_n]$ (for generator or recognizer) is realizable as a Fibonacci gate on some basis of size 1 iff there exist three constants a, b and c , such that $b^2 - 4ac \neq 0$, and for all k , where $0 \leq k \leq n - 2$,

$$ax_k + bx_{k+1} + cx_{k+2} = 0. \tag{3}$$

Proof. Here we only prove it for recognizers; the case for generator is similar.

“ \Rightarrow ”: From (1), we choose $a = (p_0 + p_1\phi)(p_0 + p_1\bar{\phi}), b = -(n_0 + n_1\phi)(p_0 + p_1\bar{\phi}) - (p_0 + p_1\phi)(n_0 + n_1\bar{\phi})$ and $c = (n_0 + n_1\phi)(n_0 + n_1\bar{\phi})$. Then $b^2 - 4ac \neq 0$ and we can verify that (3) is satisfied.

“ \Leftarrow ”: If $c \neq 0$, then $\{x_i\}$ is a second-order homogeneous linear recurrence sequence. Since $b^2 - 4ac \neq 0$, $\{x_i\}$ has the form $x_i = A'\alpha^i + B'\beta^i$ for some $\alpha \neq \beta$. By the theory of Möbius transformations, there exists a $T \in \mathbf{GL}_2$ such that $\ell_T(\phi) = \alpha$ and $\ell_T(\bar{\phi}) = \beta$. More explicitly, in (1), we can choose $A = A', B = B', n_0 = 1, n_1 = 0, p_0 = \frac{\beta\phi - \alpha\bar{\phi}}{\phi - \bar{\phi}}$ and $p_1 = \frac{\alpha - \beta}{\phi - \bar{\phi}}$. This implies that $\{x_i\}$ is realizable. The case $a \neq 0$ is similar. If $a = c = 0$, then $b \neq 0$. In this case all the $x_i = 0$ except x_0 and x_n . Then in (1), choosing $A = \frac{x_0}{(\phi - \bar{\phi})^n}, B = \frac{x_n}{(\phi - \bar{\phi})^n}, n_0 = \bar{\phi}, n_1 = -1, p_0 = \phi$ and $p_1 = -1$, we can show that $\{x_i\}$ is realizable. \square

Theorem 4.2. A set of symmetric generators G_1, G_2, \dots, G_s and recognizers R_1, R_2, \dots, R_t are simultaneously realizable as Fibonacci gates on some basis of size 1 iff there exist three constants a, b and c , such that $b^2 - 4ac \neq 0$ and the following two conditions are satisfied:

1. For any recognizer $R_i = [x_1^{(i)}, x_2^{(i)}, \dots, x_{n_i}^{(i)}]$ and any $k = 0, 1, \dots, n_i - 2, ax_k^{(i)} + bx_{k+1}^{(i)} + cx_{k+2}^{(i)} = 0$.
2. For any generator $G_j = [y_1^{(j)}, y_2^{(j)}, \dots, y_{m_j}^{(j)}]$ and any $k = 0, 1, \dots, m_j - 2, cy_k^{(j)} - by_{k+1}^{(j)} + ay_{k+2}^{(j)} = 0$.

Proof. “ \Rightarrow ”: Let $T = \begin{bmatrix} n_0 & p_0 \\ n_1 & p_1 \end{bmatrix}$ be a basis on which they are simultaneously realizable. Then all the

recognizers $R_i = [x_1^{(i)}, x_2^{(i)}, \dots, x_{n_i}^{(i)}]$ have the form (1), with only the coefficients A and B may be different. For each R_i , we can choose the same a, b and c as in Theorem 4.1. Then for any $k = 0, 1, \dots, n_i - 2, ax_k^{(i)} + bx_{k+1}^{(i)} + cx_{k+2}^{(i)} = 0$.

For the generators, replace T by $(T^{-1})^T$, we have the same result. If we define a', b' and c' according to $(T^{-1})^T$, then we can verify that $a' = -c/\det^2(T), b' = b/\det^2(T)$ and $c' = -a/\det^2(T)$. This uses properties of ϕ and $\bar{\phi}$, where ϕ is the golden ratio. It follows that $cy_k^{(j)} - by_{k+1}^{(j)} + ay_{k+2}^{(j)} = 0$.

“ \Leftarrow ”: If $c \neq 0$, then each recognizer sequence satisfies the same second-order homogeneous linear recurrence. Since $b^2 - 4ac \neq 0$, let α, β be the two distinct roots of $cX^2 + bX + a$. Each $\{x_k^{(i)}\}$ has the

form $x_k^{(i)} = A_i\alpha^k + B_i\beta^k$. Then all the $R_i = [x_1^{(i)}, x_2^{(i)}, \dots, x_{n_i}^{(i)}]$ are realizable on $T = \begin{bmatrix} 1 & \frac{\beta\phi - \alpha\bar{\phi}}{\phi - \bar{\phi}} \\ 0 & \frac{\alpha - \beta}{\phi - \bar{\phi}} \end{bmatrix}$ as

in the above proof.

Since $cy_k^{(j)} - by_{k+1}^{(j)} + ay_{k+2}^{(j)} = 0$ and $c \neq 0$, each reversed generator sequence is a second-order homogeneous linear recurrence sequence. Then $-\alpha$ and $-\beta$ are the two roots of $cX^2 - bX + a$. As a result, we know that each generator $\{y_k^{(j)}\}$ has the form $y_k^{(j)} = A'_j(-\alpha)^{m_j-k} + B'_j(-\beta)^{m_j-k}$. Then it is easy to verify that they are also realizable on T as generators.

The case $a \neq 0$ is similar. Finally if $a = c = 0$, then $b \neq 0$. In this case all the sequences have the form $[*, 0, \dots, 0, *]$, and they are all realizable on $T = \begin{bmatrix} \bar{\phi} & \phi \\ -1 & -1 \end{bmatrix}$. \square

We note in particular when $a = -c \neq 0$, then both the generator side and the recognizer side take the same form: for some m , $f_{k+2} = mf_{k+1} + f_k$, for $k = 0, \dots, \text{arity}(f) - 2$. These signatures can be used on general (not necessarily bipartite) graphs. One way to see this is as follows: Turn a general graph G into a bipartite graph, by adding a new vertex in the middle of every edge, and attach a binary EQUALITY $[1, 0, 1]$ at each new vertex. (Equivalently we form the incidence graph of G , where the left-hand are the edges of G , each given $[1, 0, 1]$, and the right-hand are the vertices of G .) We note that $[1, 0, 1]$ satisfies $f_2 = mf_1 + f_0$. See Section 8 for more discussions.

5. Some problems

In a formal sense, we already have a complete characterization of the power and expressibility of holographic algorithms with Fibonacci gates. In this section, we show some concrete problems formulated as holant problems can be solved by our new holographic algorithms. All the problems, even restricted to planar structures are not solvable by original holographic algorithms with matchgates, in the sense that the required signatures are not realizable by matchgates. Furthermore, some variations of the problems are $\#P$ -complete, which will be proved in Section 6. More discussions of these problems are given Section 8.

5.1. A coloring problem

Given a 3-regular graph, a 2-coloring (Black/White) of the edges is called *valid* if at each vertex the incident edges are not monochromatic. It is an *even* (resp. *odd*) coloring if it is a valid 2-coloring with an even (resp. odd) number of Black edges.

Input: A 3-regular graph.

Output: The number of even colorings minus the number of odd colorings.

Solution: For every edge we use the signature $[1, 0, -1]^T$ as a generator, and for every vertex we use the signature $[0, 1, 1, 0]$ as a recognizer. By choosing $a = c = 1$, $b = -1$ in Theorem 4.2, we know that they are simultaneously realizable. So we have a holographic algorithm with Fibonacci gates for this problem.

To see that this signature pair solves our problem, i.e., the problem is simply $\#[1, 0, -1][0, 1, 1, 0]$, we reason as follows. The input 3-regular graph G is replaced first by its incidence graph; this makes it bipartite. Every edge of G has the signature $[1, 0, -1]$, which means that it transmits a consistent color (Black/White), but with a multiplicative factor -1 if it is Black. At each vertex of G the requirement is that the number of Black edges, among three incident edges, is either one or two.

Here we have some comments for this problem:

1. The same signatures $[1, 0, -1]$ and $[0, 1, 1, 0]$ are not simultaneous realizable as matchgates.
2. Computing the number of all valid colorings is $\#P$ -hard. This problem can be denoted as $\#(3\text{-regular-graphs}, \{[1, 0, 1]\} \mid \{[0, 1, 1, 0]\})$, and the proof is in Theorem 6.1.
3. If we only consider planar graphs, then the number of all valid colorings *can* be computed in polynomial time by a holographic algorithm with matchgates. (Signatures $[1, 0, 1]$ and $[0, 1, 1, 0]$ are simultaneous realizable as matchgates.)

It has also been pointed out that this result implies that $\oplus P$ -Mon-NAE is in P , i.e., the parity of the number of solutions of Monotone formulae with Not-All-Equal gates (each Not-All-Equal gate is of arity 3 and each variable appears twice). If we replace the Not-All-Equal gates by OR, the problem is known to be $\oplus P$ -complete, even in the planar case; but without the replacement, the problem is in P for the planar case, shown by holographic algorithms using matchgates [19,20].

5.2. A satisfiability problem

We consider Rtw-CNF formulas, where Rtw (which stands for “read twice”) means that each variable appears in exact 2 clauses, but it may appear both positively or negatively. A variable may *lie* when it transmits inconsistent values to its clauses; when that happens, we call it a *lying variable*. An even-lying (resp. odd-lying) assignment is an assignment with an even number (resp. odd number) of lying variables. Each variable x_i is also associated with a weight w_i , such that when the transmitted value of x_i satisfies both clauses it is weighted with a factor w_i . The weight of an assignment is the product of weights w_i over such variables.

Input: A Rtw-CNF formula.

Output: The sum of weights of even-lying satisfying assignments minus the sum of weights of odd-lying satisfying assignments.

Solution: For each variable x_i , if it appears as a positive literal (x_i) in both clauses, or as a negative literal ($\neg x_i$) in both clauses, we use the signature $[1, -1, w_i]^T$ as a generator; if x_i appears once as a positive literal and once as a negative literal, we use the signature $[-1, 1, -w_i]^T$ as a generator. For every clause with k variables, we use the signature $[0, 1, 1, \dots, 1]$ (k 1’s) as a recognizer. By choosing $a = 0, b = 1, c = -1$ in Theorem 4.2, we know that they are all simultaneously realizable. So we have a holographic algorithm with Fibonacci gates for this problem.

To see that these signatures capture this problem, we note the following. There are three possibilities as to how a variable may appear in a Read-Twice CNF formula, as stated. If a variable x appears as the negative literal $\neg x$ in both clauses C and C' , we may replace x by a new variable x' which appears positively in both C and C' . The notion of *lying* and weight are unchanged by this substitution; since it is the consistency, and the satisfaction of the clauses, of the transmitted values that count. Now according to the four cases when the received values are 00, 01, 10, 11, we may easily check our assigned generator signature works for either (1) when x appears both positively, or (2) when x appears once positively and once negatively. The clauses all have the Boolean OR signature.

5.3. A family of parity problems

Here we consider a family of holant problems $\Gamma = (G, \mathcal{P})$, where each function in \mathcal{P} has the property that its value only depends on the parity of its input. Thus we have signatures of the form $[a, b, a, b, \dots]$. We show that these problems can all be solved by holographic algorithms with Fibonacci gates.

Input: A signature grid $\Gamma = (G, \mathcal{P})$ with parity signatures.

Output: Holant $_{\Gamma}$.

Solution: We apply the standard method to make the problem bipartite. For every edge in G we use the signature $[1, 0, 1]^T$ as a generator. For every vertex in G , we keep its signature and is viewed as a recognizer. By choosing $a = 1, b = 0, c = -1$ in Theorem 4.2, we know that they are all simultaneously realizable. So we have a holographic algorithm with Fibonacci gates for this problem. We note that the choice of Fibonacci gates are different from the previous problems as different parameters a, b and c are used.

6. Hardness of some problems

In this section, we prove that counting all valid colorings in the coloring problem introduced in Section 5 is #P-complete. In the holant language, we prove the following theorem.

Theorem 6.1. *The holant problem $\#(3\text{-regular-graphs}, \{[0, 1, 1, 0]\})$ is #P-complete.*

Note that each edge in a 3-regular graph is assigned a truth value, which corresponds to a Black/White edge coloring, and at every vertex it is subjected to the Not-All-Equal gate $[0, 1, 1, 0]$.

Our starting point is #NAE-3SAT. This is the well-known problem of counting the number of solutions to a Boolean formula which is a conjunction of clauses, each of which is the Not-All-Equal

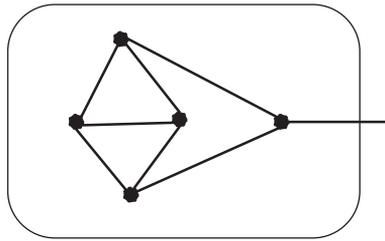


Fig. 6. The signature for every node in this $\{[0, 1, 1, 0], [1, 0, 0, 1]\}$ -gate is $[1, 0, 0, 1]$. And the signature of the gate is $[1, 1]$.

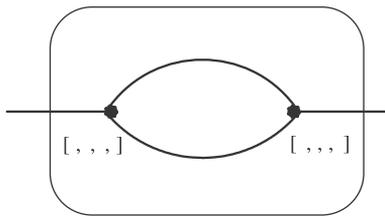


Fig. 7. An $\{[0, 1, 1, 0], [1, 0, 0, 1]\}$ -gate with signature $[0, 1, 0]$.

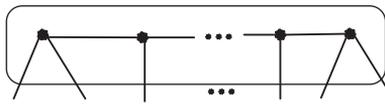


Fig. 8. This gate realizes the equality signature $[1, 0, 0, \dots, 0, 1]$ of at least 3 variables. The signature for every node in this is $[1, 0, 0, 1]$. There are $d - 2$ nodes in the gate if we want to realize the equality signature for d variables.

function on 3 literals. It is #P-complete by the dichotomy theorem [10] on Constraint Satisfiability Counting Problems. Our hardness result will be a reduction from this problem.

Our first step is to represent this problem #NAE-3SAT as a holant problem over the signature set $\{[0, 1, 1, 0], [1, 0, 0, 1]\}$. Given an instance of #NAE-3SAT we represent it by a graph with variable nodes and NAE-clause nodes as follows: For each NAE-clause node, we use the NAE signature $[0, 1, 1, 0]$. For a variable node, if it occurs just once, we use the gadget in Fig. 6 to realize the equality gate $[1, 1]$. If a variable occurs twice, we can represent it by an edge, and append it with the gadget in Fig. 7 if it is once positive and once negative. If a variable occurs in $d \geq 3$ clauses as literals, we use the gadget in Fig. 8, and each negated literal occurrence is further appended by the gadget in Fig. 7. Note that the holant of this signature grid evaluates to exactly the answer of #NAE-3SAT instance. It follows that

Lemma 6.1. *The holant problem $\#(3\text{-regular-graph}, \{[0, 1, 1, 0], [1, 0, 0, 1]\})$ is #P-complete.*

The next step in the proof uses a technique called interpolation [16,17]. Given a signature grid Ω for the above holant problem, we will construct a sequence of signature grids Ω_i for the holant problem $\#(3\text{-regular-graph}, \{[0, 1, 1, 0]\})$. And we show that from a polynomial number of queries to $\text{Holtant}_{\Omega_i}$, where $i = 0, 1, \dots$, we can compute Holtant_{Ω} in polynomial time. This will prove Theorem 6.1.

Let n be the number of nodes in Ω with the signature $[1, 0, 0, 1]$. Let x_j be the number of truth assignments to all edges of Ω where there are exactly j nodes among those n nodes which receive an assignment $(0, 0, 0)$ or $(1, 1, 1)$ on its three incident edges, and on all nodes other than these n nodes in Ω (which have a signature $[0, 1, 1, 0]$) the truth assignment satisfies the local ternary NAE condition. Clearly the answer we seek is $\text{Holtant}_{\Omega} = x_n$.

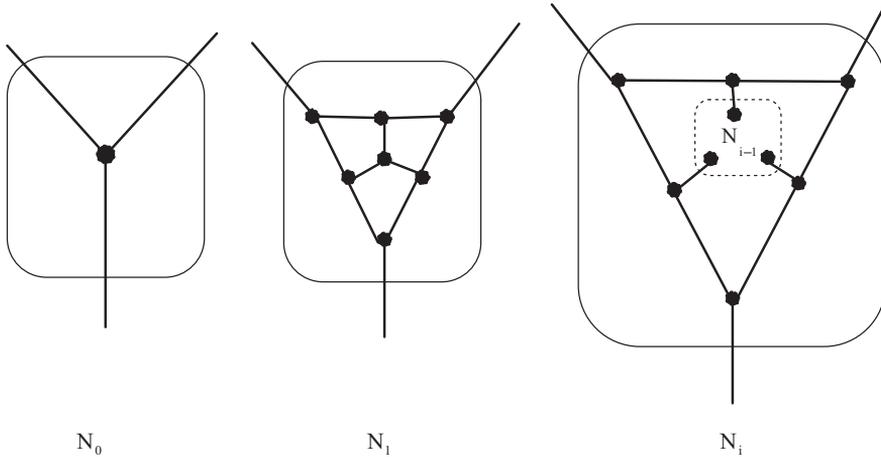


Fig. 9. The recursive construction of the $\{[0, 1, 1, 0]\}$ -gates N_i . The signature for every node in these gates is $[0, 1, 1, 0]$.

Consider the sequence of $\{[0, 1, 1, 0]\}$ -gate N_i in Fig. 9. The sequence N_i is recursively constructed. N_0 is simply a single node with the signature $[0, 1, 1, 0]$ and 3 dangling edges. For $i \geq 1$, we construct N_i by inscribing a copy of N_{i-1} inside of a triangle as shown in Fig. 9. Every node in the $\{[0, 1, 1, 0]\}$ -gate N_i is assigned the signature $[0, 1, 1, 0]$ as we must. Obviously N_i has a symmetric signature as can be seen recursively by a geometric rotation or reflection directly on the picture. If we denote the signature of N_i by $[a_i, b_i, c_i, d_i]$, there is in fact a further symmetry. We claim that $a_i = d_i$ and $b_i = c_i$, which we prove recursively. Clearly N_0 has this property. Assume this holds for N_{i-1} . For N_i , if we flip all 0 and 1 bits, we get a 1–1 map on all satisfiable truth assignments counted in a_i and d_i , and also a 1–1 map on all satisfiable truth assignments counted in b_i and c_i . Thus the signature of N_i is $[a_i, b_i, b_i, a_i]$.

Now we construct our sequence of signature grids Ω_i : For each of those n nodes in Ω_i with a signature $[1, 0, 0, 1]$, we replace it by a copy of N_i . It is easy to see that $\text{Holtant}_{\Omega_i} = \sum_{j=0}^n x_j a_i^j b_i^{n-j}$. Thus from the values $\text{Holtant}_{\Omega_i}$ for $i = 0, 1, \dots, n$, we get a linear system in x_0, x_1, \dots, x_n . Provided the coefficient matrix is non-singular, we can compute all these x_j , and $\text{Holtant}_{\Omega} = x_n$ is what we want.

To show that it is non-singular, we derive a recurrence relation for (a_i, b_i) . To compute a_{i+1} , fix all three inputs $x_{i+1}, y_{i+1}, z_{i+1}$ of the gate N_{i+1} to 1, and then fix one of the eight possible assignments to x_i, y_i, z_i , the inputs of N_i , then multiply the number of satisfiable truth assignments for the remaining six edges in N_{i+1} with the value of N_i on this input (it is either a_i or b_i), then sum over all eight assignments, we get $a_{i+1} = 20a_i + 60b_i$. (This takes some careful accounting.) And similarly we have $b_{i+1} = 20a_i + 75b_i$. By this relation and Lemma 9.2 by Vadhan [15], we know that the coefficient matrix of the above linear system is non-singular (Vandemonde). This finishes the hardness proof.

7. Basis collapse

In this section, we prove a basis collapse theorem for holographic algorithms with Fibonacci gates. These results parallel those obtained for matchgate signatures [4,5]. The motivation for allowing bases of higher dimension, at least in the case of matchgates, was that they may potentially increase the scope of holographic algorithms based on matchgates. We showed that in the case of two-vector bases, they do not actually increase the power of holographic algorithms.

Here we do the same for Fibonacci gates. But the proof techniques are quite different. In the proof for matchgates, we construct size 1 bases and matchgates from the given high dimensional bases and matchgates, then prove that they realize the same signatures. Here we make use of Theorem 4.2, which characterizes all the signatures simultaneously realizable on a size 1 basis. We prove the collapse

theorem by showing that signatures realizable on high dimensional bases also satisfy the conditions of Theorem 4.2. Here the properties of the golden ratio ϕ and its conjugate $\bar{\phi}$ are crucial.

Theorem 7.1. *Let $T = [\mathbf{n}, \mathbf{p}]$ be a basis of size k . Suppose a set of symmetric generators G_1, G_2, \dots, G_s and recognizers R_1, R_2, \dots, R_t are simultaneously realizable as Fibonacci gates on T . Then they are also simultaneously realizable on a basis of size 1.*

Proof. For any recognizer R_i , there exists some Fibonacci signature F_i such that $R_i = F_i T^{\otimes n}$.

$$\begin{aligned} R_i &= (A_i(1, \phi)^{\otimes nk} + B_i(1, \bar{\phi})^{\otimes nk})T^{\otimes n} \\ &= A_i(1, \phi)^{\otimes nk}T^{\otimes n} + B_i(1, \bar{\phi})^{\otimes nk}T^{\otimes n} \\ &= A_i(\langle (1, \phi)^{\otimes k}, \mathbf{n} \rangle, \langle (1, \phi)^{\otimes k}, \mathbf{p} \rangle)^{\otimes n} + B_i(\langle (1, \bar{\phi})^{\otimes k}, \mathbf{n} \rangle, \langle (1, \bar{\phi})^{\otimes k}, \mathbf{p} \rangle)^{\otimes n}, \end{aligned}$$

where we denote $\langle \cdot, \cdot \rangle$ for inner product (or the tensor contraction function between a dual pair). Let $\Phi = (1, \phi)^{\otimes k}$, $\bar{\Phi} = (1, \bar{\phi})^{\otimes k}$ be row vectors, and $R_i = [x_1^{(i)}, x_2^{(i)}, \dots, x_n^{(i)}]$. Then $x_n^{(i)} = A_i \langle \Phi, \mathbf{n} \rangle^{n_i-h} \langle \Phi, \mathbf{p} \rangle^h + B_i \langle \bar{\Phi}, \mathbf{n} \rangle^{n_i-h} \langle \bar{\Phi}, \mathbf{p} \rangle^h$. Choose $a = \langle \Phi, \mathbf{p} \rangle \langle \bar{\Phi}, \mathbf{p} \rangle = \mathbf{p}^T \Phi^T \bar{\Phi} \mathbf{p}$, $b = -\langle \Phi, \mathbf{p} \rangle \langle \bar{\Phi}, \mathbf{n} \rangle - \langle \Phi, \mathbf{n} \rangle \langle \bar{\Phi}, \mathbf{p} \rangle = -\mathbf{p}^T \Phi^T \bar{\Phi} \mathbf{n} - \mathbf{n}^T \Phi^T \bar{\Phi} \mathbf{p}$, and $c = \langle \Phi, \mathbf{n} \rangle \langle \bar{\Phi}, \mathbf{n} \rangle = \mathbf{n}^T \Phi^T \bar{\Phi} \mathbf{n}$, we can verify that $ax_k^{(i)} + bx_{k+1}^{(i)} + cx_{k+2}^{(i)} = 0$ are satisfied for all recognizers.

For the generators, we have $F_j^T = T^{\otimes m_j} G_j$. If $m_j = 1$, then G_j is of arity 1 and it is realizable on any size 1 basis. Now we consider the case $m_j = 2$. Then G_j is of arity 2 and we can assume that $G_j = (g_0, g_1, g_1, g_2)^T$. Then we have

$$F_j^T = (T \otimes T)G_j = g_0 \mathbf{n} \otimes \mathbf{n} + g_1 (\mathbf{n} \otimes \mathbf{p} + \mathbf{p} \otimes \mathbf{n}) + g_2 \mathbf{p} \otimes \mathbf{p}.$$

Assume $F_j^T = A'_j \begin{bmatrix} 1 \\ \phi \end{bmatrix}^{\otimes 2k} + B'_j \begin{bmatrix} 1 \\ \bar{\phi} \end{bmatrix}^{\otimes 2k}$, we have for any $x, y \in \{0, 1\}^k$,

$$g_0 n_x n_y + g_1 (n_x p_y + p_x n_y) + g_2 p_x p_y = A'_j \phi^{wt(x)+wt(y)} + B'_j \bar{\phi}^{wt(x)+wt(y)}. \tag{4}$$

By the sufficiency condition of Theorem 4.2, we only need to prove $cg_0 - bg_1 + ag_2 = 0$. Substituting a, b and c in the above equation, we have the following:

$$\begin{aligned} &cg_0 - bg_1 + ag_2 \\ &= \sum_{x,y \in \{0,1\}^k} \left[g_0 n_x n_y \phi^{wt(x)} \bar{\phi}^{wt(y)} + g_1 (p_x n_y \phi^{wt(x)} \bar{\phi}^{wt(y)} + n_x p_y \phi^{wt(x)} \bar{\phi}^{wt(y)}) \right. \\ &\quad \left. + g_2 p_x p_y \phi^{wt(x)} \bar{\phi}^{wt(y)} \right] \\ &= \sum_{x,y \in \{0,1\}^k} \phi^{wt(x)} \bar{\phi}^{wt(y)} (g_0 n_x n_y + g_1 (n_x p_y + p_x n_y) + g_2 p_x p_y) \\ &= \sum_{x,y \in \{0,1\}^k} \phi^{wt(x)} \bar{\phi}^{wt(y)} \left(A'_j \phi^{wt(x)+wt(y)} + B'_j \bar{\phi}^{wt(x)+wt(y)} \right) \\ &= A'_j \sum_{x,y \in \{0,1\}^k} \phi^{2wt(x)} (-1)^{wt(y)} + B'_j \sum_{x,y \in \{0,1\}^k} (-1)^{wt(x)} \bar{\phi}^{2wt(y)} \\ &= A'_j \sum_{x \in \{0,1\}^k} \phi^{2wt(x)} \sum_{y \in \{0,1\}^k} (-1)^{wt(y)} + B'_j \sum_{x \in \{0,1\}^k} (-1)^{wt(x)} \sum_{y \in \{0,1\}^k} \bar{\phi}^{2wt(y)} = 0, \end{aligned}$$

proving the theorem for $m_j = 2$. Here we used the crucial relation $\phi \bar{\phi} = -1$.

Now we can assume that $m_j \geq 3$. Again we have $F_j^T = T^{\otimes m_j} G_j$. Since $\text{rank}(T) = 2$, we can find

some $\tilde{T} = \begin{bmatrix} \tilde{\mathbf{n}} \\ \tilde{\mathbf{p}} \end{bmatrix}$, where $\tilde{\mathbf{n}}$ and $\tilde{\mathbf{p}}$ are 2^k dimensional row vectors, such that $\tilde{T}T = I$. So $G_j = \tilde{T}^{\otimes m_j} F_j^T$.

$$G_j = \tilde{T}^{\otimes m_j} F_j^T = \begin{bmatrix} \tilde{\mathbf{n}} \\ \tilde{\mathbf{p}} \end{bmatrix}^{\otimes m_j} \left(A'_j \begin{bmatrix} 1 \\ \phi \end{bmatrix}^{\otimes m_j k} + B'_j \begin{bmatrix} 1 \\ \bar{\phi} \end{bmatrix}^{\otimes m_j k} \right) = A'_j \begin{bmatrix} \tilde{\mathbf{n}}\Phi^T \\ \tilde{\mathbf{p}}\Phi^T \end{bmatrix}^{\otimes m_j} + B'_j \begin{bmatrix} \tilde{\mathbf{n}}\bar{\Phi}^T \\ \tilde{\mathbf{p}}\bar{\Phi}^T \end{bmatrix}^{\otimes m_j}.$$

Denoting $\alpha = \tilde{\mathbf{n}}\Phi^T$, $\beta = \tilde{\mathbf{p}}\Phi^T$, $\gamma = \tilde{\mathbf{n}}\bar{\Phi}^T$ and $\delta = \tilde{\mathbf{p}}\bar{\Phi}^T$, we have $G_j = A'_j \begin{bmatrix} \alpha \\ \beta \end{bmatrix}^{\otimes m_j} + B'_j \begin{bmatrix} \gamma \\ \delta \end{bmatrix}^{\otimes m_j}$.

Here if $A'_j = 0$ or $B'_j = 0$, then G_j can be decomposed into an arity 1 tensor and is realizable on any size 1 basis. So we can assume that $A'_j B'_j \neq 0$.

Substituting this back to $F_j^T = T^{\otimes m_j} G_j$, we have

$$A'_j \begin{bmatrix} 1 \\ \phi \end{bmatrix}^{\otimes m_j k} + B'_j \begin{bmatrix} 1 \\ \bar{\phi} \end{bmatrix}^{\otimes m_j k} = A'_j (\alpha \mathbf{n} + \beta \mathbf{p})^{\otimes m_j} + B'_j (\gamma \mathbf{n} + \delta \mathbf{p})^{\otimes m_j}.$$

From Lemma 9.1, we know that $\alpha \mathbf{n} + \beta \mathbf{p} = \omega_1 \Phi^T$, $\gamma \mathbf{n} + \delta \mathbf{p} = \omega_2 \bar{\Phi}^T$, where ω_1 and ω_2 are roots of unity $\omega_1^{m_j} = \omega_2^{m_j} = 1$, or $\alpha \mathbf{n} + \beta \mathbf{p} = \omega'_1 \bar{\Phi}^T$, $\gamma \mathbf{n} + \delta \mathbf{p} = \omega'_2 \Phi^T$, where $(\omega'_1)^{m_j} = \frac{B'_j}{A'_j}$ and $(\omega'_2)^{m_j} = \frac{A'_j}{B'_j}$.

In the following proof we will assume the first case, the second case is similar. Write $G_j = [y_1^{(j)}, y_2^{(j)}, \dots, y_{m_j}^{(j)}]$, we have $y_h^{(j)} = A'_j \alpha^{m_j-h} \beta^h + B'_j \gamma^{m_j-h} \delta^h$. Now we verify that $cy_h^{(j)} - by_{h+1}^{(j)} + ay_{h+2}^{(j)} = 0$. By the definition of a , b and c ,

$$cy_h^{(j)} - by_{h+1}^{(j)} + ay_{h+2}^{(j)} = \mathbf{n}^T \Phi^T \bar{\Phi} \mathbf{n} y_h^{(j)} + (\mathbf{p}^T \Phi^T \bar{\Phi} \mathbf{n} + \mathbf{n}^T \Phi^T \bar{\Phi} \mathbf{p}) y_{h+1}^{(j)} + \mathbf{p}^T \Phi^T \bar{\Phi} \mathbf{p} y_{h+2}^{(j)}.$$

We substitute $y_h^{(j)}$. Here we only do the calculation on the A'_j part here. The B'_j part is similar.

$$\begin{aligned} & \text{The coefficient of } A'_j \\ &= \mathbf{n}^T \Phi^T \bar{\Phi} \mathbf{n} \alpha^{m_j-h} \beta^h + (\mathbf{p}^T \Phi^T \bar{\Phi} \mathbf{n} + \mathbf{n}^T \Phi^T \bar{\Phi} \mathbf{p}) \alpha^{m_j-h-1} \beta^{h+1} + \mathbf{p}^T \Phi^T \bar{\Phi} \mathbf{p} \alpha^{m_j-h-2} \beta^{h+2} \\ &= \mathbf{n}^T \Phi^T \bar{\Phi} \mathbf{n} \alpha^{m_j-h} \beta^h + \mathbf{n}^T \Phi^T \bar{\Phi} \mathbf{p} \alpha^{m_j-h-1} \beta^{h+1} + \mathbf{p}^T \Phi^T \bar{\Phi} \mathbf{n} \alpha^{m_j-h-1} \beta^{h+1} \\ & \quad + \mathbf{p}^T \Phi^T \bar{\Phi} \mathbf{p} \alpha^{m_j-h-2} \beta^{h+2} \\ &= \alpha^{m_j-h-1} \beta^h \mathbf{n}^T \Phi^T \bar{\Phi} (\alpha \mathbf{n} + \beta \mathbf{p}) + \alpha^{m_j-h-2} \beta^{h+1} \mathbf{p}^T \Phi^T \bar{\Phi} (\alpha \mathbf{n} + \beta \mathbf{p}) \\ &= \alpha^{m_j-h-1} \beta^h \omega_1 \mathbf{n}^T \Phi^T \bar{\Phi} \Phi^T + \alpha^{m_j-h-2} \beta^{h+1} \omega_1 \mathbf{p}^T \Phi^T \bar{\Phi} \Phi^T = 0. \end{aligned}$$

The last equation uses the fact that $\bar{\Phi} \Phi^T = [((1, \bar{\phi}), (1, \phi))]^{\otimes k} = 0$. So by Theorem 4.2, we know that all the symmetric generators G_1, G_2, \dots, G_s and recognizers R_1, R_2, \dots, R_t are simultaneously realizable as Fibonacci gates on a basis of size 1. This completes the proof. \square

8. Some further results and discussions

All results on Fibonacci gates can be extended to generalized Fibonacci gates which also admit polynomial time algorithms. These are defined as follows: For any fixed parameter m , we consider any sequence $[f_0, f_1, \dots, f_n]$, where it satisfies the following recurrence, for $i = 0, \dots, n - 2$,

$$f_{i+2} = m \cdot f_{i+1} + f_i.$$

One can repeat the proof in Section 3 to show that for all m , the generalized Fibonacci gates with parameter m define a tractable family of holant problems. If the vertices of a signature grid Ω are assigned signatures from this family (with the same m , but all possible initial two values f_0 and f_1), the counting problem Holant_{Ω} can also be computed in polynomial time. Furthermore, based on a set of generalized Fibonacci gates with parameter m , we can also develop holographic transformations. We note that the basic Fibonacci gates correspond to taking $m = 1$.

However, these generalized Fibonacci gates can already be obtained by applying holographic transformations to the basic Fibonacci gates, except in the cases $m = \pm 2i$. Therefore they do not constitute a significant extension of the scope of holographic algorithms based on the basic Fibonacci gates. However, the form $f_{i+2} = m \cdot f_{i+1} + f_i$ is very useful in practice.

It can even be proved that this class of generalized Fibonacci gates is precisely those symmetric function gates which are closed under the compositional closure when we merge signature grids.

It has been pointed to us that there are alternative algorithms for some problems in Section 5. E.g., for the first (coloring) problem, one can use the signature $[0, i, -1, 0]$ for each degree 3 vertex, and then the problem can also be stated as $\#[1, 0, 1] \mid [0, i, -1, 0]$. Then one applies the generalized Fibonacci gates with $m = i$, where $f_{i+2} = m f_{i+1} + f_i$. For the third (parity) problem, one can use generalized Fibonacci gates with $m = 0$.

For the second problem, in the *special case* when all weights $w_i = 1$, it is possible to perform the following *diagonal* basis transformation

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

Under this basis, the (unweighted) problem $\#[1, -1, 1] \mid [0, 1, 1, \dots, 1]$ is transformed to $\#[1, 1, 1] \mid [0, -1, 1, -1, \dots]$, which can be solved using generalized Fibonacci gates with $m = -1$. (There is also a simpler solution at this point, if one realizes that $[1, 1, 1] = (1, 1) \otimes (1, 1)$ and the problem decouples, and it becomes essentially trivial.) But to our knowledge the problem with weights is non-trivial and can not be solved in this alternative fashion.

Our perspective in this paper is to develop a coherent theory based on the basic Fibonacci gates under holographic reductions. Our viewpoint is that this one single method can be added to the general toolkit of algorithm design. One *could* choose to develop separate algorithms, one problem at a time, sometimes with clever derivations. But we feel there is an elegance and utility in seeing things in the uniform way under holographic reductions. There is a value in this approach where the theory gives a uniform treatment of these problems, even when alternative methods can be found to solve the same problems individually. We also believe that the usefulness of the theory of holographic algorithms is validated by the dichotomy theorems we have since obtained [7].

Acknowledgments

We sincerely thank Les Valiant for his comments and ideas regarding this work, and regarding holographic algorithms in general. We benefited greatly from discussions with him. We are also very grateful to Martin Dyer and Angsheng Li. Discussions with Martin and Angsheng helped a great deal in developing some of the ideas reported here. We thank the anonymous referees for their careful reading and insightful comments, which greatly helped us to improve the presentation of the paper. We also thank many people who have commented on the preliminary draft of this paper.

Appendix

9. Technical lemmas

We quote two lemmas.

Lemma 9.1. Let \mathbf{a} , \mathbf{b} , \mathbf{c} , \mathbf{d} be four vectors and suppose \mathbf{c} , \mathbf{d} are linearly independent with each other. If for some $n \geq 3$ we have $\mathbf{a}^{\otimes n} + \mathbf{b}^{\otimes n} = \mathbf{c}^{\otimes n} + \mathbf{d}^{\otimes n}$. Then $\mathbf{a} = \omega_1 \mathbf{c}$ and $\mathbf{b} = \omega_2 \mathbf{d}$ or $\mathbf{a} = \omega_1 \mathbf{d}$ and $\mathbf{b} = \omega_2 \mathbf{c}$ for some $\omega_1^n = \omega_2^n = 1$.

This is a special case of more general results; see [11].

Lemma 9.2 (Vadhan). Let A, B, C, D, x_0 and y_0 be rational numbers. Define the sequences (x_i, y_i) recursively by $x_{i+1} = Ax_i + By_i$ and $y_{i+1} = Cx_i + Dy_i$. Then the sequence $\{z_i = x_i/y_i\}$ never repeats as long as all of the following conditions hold:

$$\left\{ \begin{array}{l} AD - BC \neq 0 \\ D^2 - 2AD + A^2 + 4BC \neq 0 \\ D + A \neq 0 \\ D^2 + AD + A^2 + BC \neq 0 \\ D^2 + A^2 + 2BC \neq 0 \\ D^2 - AD + A^2 + 3BC \neq 0 \\ By_0^2 - Cx_0^2 - (A - D)x_0y_0 \neq 0 \end{array} \right. .$$

References

- [1] J.-Y. Cai, Vinay Choudhary, Some results on matchgates and holographic algorithms, in: Proceedings of ICALP 2006, Part I, Lecture Notes in Computer Science, vol. 4051, pp. 703–714 (also available at ECCS TR06-048, 2006).
- [2] J.-Y. Cai, Vinay Choudhary, Valiant's holant theorem and matchgate tensors, Theor. Comput. Sci. 384 (1) (2007) 22–32.
- [3] J.-Y. Cai, Pinyan Lu, Holographic algorithms: from art to science, in: The Proceedings of STOC, 2007, pp. 401–410.
- [4] J.-Y. Cai, Pinyan Lu, Bases collapse in holographic algorithms, Comput. Complex. 17 (2) (2008) 254–281.
- [5] J.-Y. Cai, Pinyan Lu, Holographic algorithms: the power of dimensionality resolved, Theor. Comput. Sci. 410 (18) (2009) 1618–1628.
- [6] J.-Y. Cai, Pinyan Lu, Mingji Xia, Holographic algorithms by fibonacci gates and holographic reductions for hardness, in: FOCS 2008: Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science, 2008, pp. 644–653.
- [7] J.-Y. Cai, Pinyan Lu, Mingji Xia, Holant problems and counting CSP, in: STOC 2009: The 41st Annual ACM Symposium on Theory of Computing, 2009, pp. 715–724.
- [8] J.-Y. Cai, Pinyan Lu, Mingji Xia, Holographic algorithms with matchgates capture precisely tractable planar #CSP, in: FOCS 2010: Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science, 2010, pp. 427–436.
- [9] C.T.J. Dodson, T. Poston, Tensor Geometry, Graduate Texts in Mathematics, vol. 130, second ed., Springer-Verlag, New York, 1991.
- [10] Nadia Creignou, Miki Hermann, Complexity of generalized satisfiability counting problems, Inf. Comput. 125 (1) (1996) 1–12.
- [11] Joe Harris, Algebraic Geometry: A First Course, Graduate Texts in Mathematics, vol. 133, Springer, 1992.
- [12] P.W. Kasteleyn, The statistics of dimers on a lattice, Physica 27 (1961) 1209–1225.
- [13] P.W. Kasteleyn, Graph theory and crystal physics, in: F. Harary (Ed.), Graph Theory and Theoretical Physics, Academic Press, London, 1967, pp. 43–110.
- [14] H.N.V. Temperley, M.E. Fisher, Dimer problem in statistical mechanics – an exact result, Philoso. Mag. 6 (1961) 1061–1063.
- [15] S. Vadhan, The complexity of counting in sparse, regular, and planar graphs, SIAM J. Comput. 31 (2) (2001) 398–427.
- [16] Leslie G. Valiant, The complexity of computing the permanent, Theor. Comput. Sci. 8 (1979) 189–201.
- [17] Leslie G. Valiant, The complexity of enumeration and reliability problems, SIAM J. Comput. 8 (3) (1979) 410–421.
- [18] L.G. Valiant, Quantum circuits that can be simulated classically in polynomial time, SIAM J. Comput. 31 (4) (2002) 1229–1254.
- [19] L.G. Valiant, Holographic algorithms, SIAM J. Comput. 37 (5) (2008) 1565–1594. (A preliminary version appeared in Proceedings of the 45th IEEE Symposium on Foundations of Computer Science, 2004, pp. 306–315. A more detailed version appeared in ECCS Report TR05-099).
- [20] L.G. Valiant, Accidental algorithms, in: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science, 2006, pp. 509–517.
- [21] Mingji Xia, Peng Zhang, Wenbo Zhao, Computational complexity of counting problems on 3-regular planar graphs, Theor. Comput. Sci. 384 (1) (2007) 111–125.