


# Eulerian Orientations and Hadamard Codes: A Novel Connection via Counting

Shuai Shao   

School of Computer Science and Technology & Hefei National Laboratory,  
University of Science and Technology of China, Hefei, China

Zhuxiao Tang 

School of the Gifted Young, University of Science and Technology of China, Hefei, China

---

## Abstract

We discover a novel connection between two classical mathematical notions, Eulerian orientations and Hadamard codes by studying the counting problem of Eulerian orientations ( $\#EO$ ) with local constraint functions imposed on vertices. We present two special classes of constraint functions and a chain reaction algorithm, and show that the  $\#EO$  problem defined by each class alone is polynomial-time solvable by the algorithm. These tractable classes of functions are defined inductively, and quite remarkably the base level of these classes is characterized perfectly by the well-known Hadamard code. Thus, we establish a novel connection between counting Eulerian orientations and coding theory. We also prove a  $\#P$ -hardness result for the  $\#EO$  problem when constraint functions from the two tractable classes appear together.

**2012 ACM Subject Classification** Mathematics of computing  $\rightarrow$  Graph theory

**Keywords and phrases** Eulerian orientations, Hadamard codes, Counting problems, Tractable classes

**Digital Object Identifier** 10.4230/LIPIcs.ITCS.2025.86

**Related Version** *Full Version*: <https://arxiv.org/abs/2411.02612>.

**Funding** Supported by the Innovation Program for Quantum Science and Technology, 2021ZD0302901.

**Acknowledgements** The first author wants to thank Jin-Yi Cai and Zhiguo Fu for many valuable discussions. The authors want to thank anonymous reviewers for their helpful comments.

## 1 Introduction

The notion of Eulerian orientations arises from the historically notable *Seven Bridges of Königsberg* problem, whose solution by Euler in 1736 [14] is considered to be the first result of graph theory. Given an undirected graph  $G$ , an Eulerian orientation of  $G$  is an assignment of a direction to each edge of  $G$  such that at each vertex  $v$ , the number of incoming edges is equal to the number of outgoing edges. A connected graph has an Eulerian orientation, called an Eulerian graph if and only if every vertex has even degree. This is the well-known Euler's theorem, whose first complete proof was given back in the 1800s [3]. In terms of computational complexity, Euler's theorem implies that the decision problem of Eulerian orientations of a graph is polynomial-time solvable (tractable). While for the counting problem, Mihail and Winkler showed that counting the number of Eulerian orientations of an undirected graph is  $\#P$ -complete in 1996 [18], more than one hundred years later after Euler's theorem. However, the counting problem becomes tractable when certain particular restrictions are imposed on edges. An intriguing example of such tractable problems comes from computing the partition function of the six-vertex model [19, 22, 20], one of the most intensively studied models in statistical physics. In this example, the graphs are 4 regular graphs and on each vertex exactly one edge incident to it is restricted to take the direction coming into the vertex. Then counting the number of Eulerian orientations obeying restrictions on these edges is proved to be tractable [7].



© Shuai Shao and Zhuxiao Tang;

licensed under Creative Commons License CC-BY 4.0

16th Innovations in Theoretical Computer Science Conference (ITCS 2025).

Editor: Raghu Meka; Article No. 86; pp. 86:1–86:13

Leibniz International Proceedings in Informatics



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

In this paper, we further study the counting restricted Eulerian orientations ( $\#EO$ ) problem, defined by constraint functions placed at each vertex that represent restrictions on edges incident to the vertex. We consider for which class of constraint functions, the  $\#EO$  problem is tractable. Before we formally define the problem and describe our results, we first take a detour to another classical notion in coding theory, the Hadamard code. The Hadamard code is an error-correcting code used for error detection and correction when transmitting messages over very noisy or unreliable channels. Because of its plentiful mathematical properties, the Hadamard code is not only used in coding and communication, but also intensely studied in various areas of mathematics and theoretical computer science. However, it is not known how the Hadamard code can be related to Eulerian orientations, and particularly, how it can be used to carve out tractable classes for the  $\#EO$  problem. Quite surprisingly, in this paper, we establish a novel connection between these two well-studied mathematical notions. We present new tractable classes for the  $\#EO$  problem and a corresponding algorithm based on a chain reaction. These classes are defined inductively and the base levels of them, defined as kernels are characterized perfectly by the Hadamard code (or more precisely, the balanced Hadamard code).

Now we formally define the  $\#EO$  problem. A 0-1 valued constraint function (or a signature)  $f$  of arity  $n$  is a map  $\mathbb{Z}_2^n \rightarrow \{0, 1\}$ . The support of  $f$  is  $\mathcal{S}(f) = \{\alpha \in \mathbb{Z}_2^n \mid f(\alpha) = 1\}$ . A signature  $f$  of arity  $2n$  is an *Eulerian orientation* (EO) signature if for every  $\alpha \in \mathcal{S}(f)$ ,  $\text{wt}(\alpha) = n$ . The problem  $\#EO(\mathcal{F})$  specified by a set  $\mathcal{F}$  of EO signatures is defined as follows. The input is a graph  $G$  where each vertex  $v$  of  $G$  is associated with some function  $f_v$  from  $\mathcal{F}$ . The incident edges to  $v$  are totally ordered and correspond to input variables to  $f_v$ . Each edge has two ends, and an orientation of the edge is denoted by assigning 0 to the head and 1 to the tail. Thus, locally at every vertex  $v$ , a 0 represents an incoming edge and a 1 represents an outgoing edge. An Eulerian orientation corresponds to an assignment to each edge (01 or 10) where the numbers of 0's and 1's at each  $v$  are equal. Then, the local constraint function  $f_v$  takes value 1 if the restriction imposed by  $f_v$  are satisfied by the local assignment on edges incident to  $v$ . Otherwise,  $f_v = 0$ . Since every  $f_v$  is an EO signature, it takes value 1 only if the numbers of input 0's and 1's are equal. Thus, only Eulerian orientations can contribute value 1. An Eulerian orientation contributes value 1 if  $f_v = 1$  for every vertex  $v$ . The  $\#EO$  problem outputs the number of Eulerian orientations contributing value 1.

► **Example 1.** Let  $\mathcal{F} = \{g_2, g_4, \dots, g_{2n}, \dots\}$ , where  $\mathcal{S}(g_{2n}) = \{\alpha \in \mathbb{Z}_2^{2n} \mid \text{wt}(\alpha) = n\}$ . Then  $\#EO(\mathcal{F})$  counts the total number of all Eulerian orientations of the input graph, which is  $\#P$ -hard.

► **Example 2.** Let  $f_2$  be a signature of arity 4 with support  $\mathcal{S}(f_2) = \{1100, 1010, 1001\}$ . Then,  $\#EO(\{f_2\})$  computes the partition function of a six-vertex model which is tractable. The reason for naming this function  $f_2$  will be explained in Section 4.

The  $\#EO$  problem has an intrinsic significance in counting problems. In [5], the framework of  $\#EO$  problems is formally introduced and is showed to be expressive enough to encompass all Boolean counting constraint satisfaction problems ( $\#CSP$ ) [12, 13, 4, 10] with arbitrary constraint functions that are not necessarily supported on half weighted inputs, although the  $\#EO$  problem itself requires all constraint functions to be supported on half weighted inputs. Also, the  $\#EO$  problem encompasses many natural problems from statistical physics and combinatorics, such as the partition function of the six-vertex model and the evaluation of the Tutte polynomial at the point  $(3, 3)$  [15]. Cai, Fu and Xia proved a complexity classification of the partition function of the six-vertex model on general 4-regular graphs [7].

Besides these interesting concrete problems expressible as #EO problems, the study of the #EO framework plays a crucial role in a broader picture, the complexity classification program for a much more expressive counting framework, the Holant problem. Significant progress has been made in the complexity classification of Holant problems [9, 8, 1, 11, 2, 16, 6], and a full complexity dichotomy for all real-valued Holant problems is achieved [21]. This dichotomy is crucially based on a complexity dichotomy for the #EO problem [5] with complex-valued signatures assuming a physical symmetry called *arrow reversal symmetry* (ARS) since under a suitable holographic transformation, the ARS property corresponds to precisely real-valued signatures. In this paper, we consider the 0-1 valued #EO problem without assuming ARS, which may serve as a building block for the ultimate classification for all complex-valued Holant problems. For this problem, we give new tractable classes that are not covered by all the previously known tractable classes for Holant problems.

We use  $a(f)$  to denote the arity of  $f$ , and  $A(f) = \{1, 2, \dots, a(f)\}$  to denote the indices of variables of  $f$ . We use  $\delta_1$  and  $\delta_0$  to denote the unary signatures where  $\mathcal{S}(\delta_1) = \{1\}$  and  $\mathcal{S}(\delta_0) = \{0\}$  respectively. Pinning the  $i$ -th variable of  $f$  to 0 gives the signature  $f(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n)$  of arity  $n-1$ , denoted by  $f_i^0$ . Similarly, we define  $f_i^1$ . Let  $f$  and  $g$  be two signatures of support size  $n$  and  $m$  respectively. Suppose  $\mathcal{S}(f) = \{\alpha_1, \alpha_2, \dots, \alpha_n\}$  and  $\mathcal{S}(g) = \{\beta_1, \beta_2, \dots, \beta_m\}$ . Then the tensor product  $f \otimes g$  of  $f$  and  $g$  is the signature where  $\mathcal{S}(f \otimes g) = \{\alpha_i \beta_j | 1 \leq i \leq n, 1 \leq j \leq m\}$ . Note that when we take a tensor product of the functions, the variables need not be in the same order of the product. A nonzero signature  $g$  is a factor of  $f$ , denoted by  $g | f$ , if there is a signature  $h$  such that  $f = g \otimes h$  or there is a constant  $\lambda$  such that  $f = \lambda \cdot g$ . Otherwise,  $g$  is not factor of  $f$ , denoted by  $g \nmid f$ . In the following, we first state the definition of affine signatures, which are known to be tractable for the #EO problem. Then we give the new tractable classes.

► **Definition 3 (Affine).** A signature  $f(x_1, x_2, \dots, x_n)$  of arity  $n$  is affine, denoted by  $f \in \mathcal{A}$  if  $\mathcal{S}(f) = \{x \in \mathbb{Z}_2^n | AX = 0\}$ , where  $X = (x_1, x_2, \dots, x_n, 1)$  and  $A$  is a matrix over  $\mathbb{Z}_2$ . Note that  $f \in \mathcal{A}$  when  $\mathcal{S}(f) = \emptyset$ , is called a zero signature.

► **Definition 4 ( $\delta_1$ -affine and  $\delta_0$ -affine).** An EO signature  $f$  is  $\delta_1$ -affine, denoted by  $f \in \mathcal{D}_1$  if  $f = \delta_1 \otimes g$  for some  $g$  where for every  $i \in A(g)$ ,  $g_i^0 \in \mathcal{A} \cup \mathcal{D}_1$ . Symmetrically, an EO signature  $f$  is  $\delta_0$ -affine, denoted by  $f \in \mathcal{D}_0$  if  $f = \delta_0 \otimes g$  for some  $g$  where for every  $i \in A(g)$ ,  $g_i^1 \in \mathcal{A} \cup \mathcal{D}_0$ .

For example, the 4-ary signature  $f$  in Example 2 is an  $\delta_1$ -affine signature. Note that  $\mathcal{D}_1$  (or  $\mathcal{D}_0$ ) does not encompass  $\mathcal{A}$ , since an affine signature may not have a  $\delta_1$  (or  $\delta_0$ ) factor. Thus, our tractable classes are  $\mathcal{A} \cup \mathcal{D}_1$  and  $\mathcal{A} \cup \mathcal{D}_0$ , rather than  $\mathcal{D}_1$  and  $\mathcal{D}_0$ .

► **Theorem 5.** #EO( $\mathcal{A} \cup \mathcal{D}_1$ ) and #EO( $\mathcal{A} \cup \mathcal{D}_0$ ) are tractable.

The tractability result is established via a chain reaction algorithm, in which the presence of a  $\delta_1$  or  $\delta_0$  signature plays a role similar to a neutron in a nuclear chain reaction which initializes the reaction and generates a new neutron (a  $\delta_1$  or  $\delta_0$  signature respectively) by propagation which makes the chain reaction continue. Eventually, the chain reaction terminates when a stable state (a tractable instance of #EO( $\mathcal{A}$ )) is reached.

Although the reaction algorithm works for  $\delta_1$ -affine signatures or  $\delta_0$ -affine signatures alone, it does not work when both of them appear. This is because when both  $\delta_1$ -affine signatures and  $\delta_0$ -affine signatures appear, by connecting  $\delta_1$  and  $\delta_0$  using  $\neq$ , both of them are killed by each other and no new  $\delta_1$  or  $\delta_0$  are generated. Thus, the chain reaction is unsustainable. This is somewhat similar to the phenomenon of electron-positron annihilation in nuclear physics. In fact, we can show that the #EO problem is #P-hard when they both appear.

► **Theorem 6.**  $\#EO(\mathcal{D}_0 \cup \mathcal{D}_1)$  is  $\#P$ -hard.

Since the classes  $\mathcal{D}_1$  and  $\mathcal{D}_0$  are defined inductively, it is not an easy task to get explicit expressions for them in a form similar to other known tractable classes for counting problems such as affine signatures. Note that the tractability of  $\mathcal{D}_1$  or  $\mathcal{D}_0$  is obtained by eventually reducing the problem into a problem where all signatures are affine signatures. Thus, we focus on  $\delta_1$ -affine (or  $\delta_0$ -affine) signatures from which by pinning an arbitrary variable to 0 (or 1 respectively), we always get an affine signature. In other words, these  $\delta_1$ -affine (or  $\delta_0$ -affine) signatures are at the first level of the inductive hierarchy. We define them as kernels.

► **Definition 7** ( $\delta_1$ -affine and  $\delta_0$ -affine kernel). An EO signature  $f$  is a  $\delta_1$ -affine kernel, denoted by  $f \in K(\mathcal{D}_1)$  if it satisfies the following two properties:

1.  $f = \delta_1^{\otimes m} \otimes h$ , for some  $m \in \mathbb{Z}_+$  where  $h$  is a signature of positive arity,  $\delta_1 \nmid h$  and  $h \notin \mathcal{A}$ ;
2. for every  $i \in A(h)$ ,  $h_i^0 \in \mathcal{A}$ .

Symmetrically, we define  $\delta_0$ -affine kernels denoted by  $K(\mathcal{D}_0)$  by switching 0 and 1 everywhere. Notice that  $h \notin \mathcal{A}$  implies  $f \notin \mathcal{A}$ . Thus,  $\delta_1$ -affine and  $\delta_0$ -affine kernels are not affine.

A main technical contribution of the paper is a complete characterization of  $\delta_1$ -affine and  $\delta_0$ -affine kernels using the Hadamard code. The Hadamard code refers to the code based on Sylvester's construction of Hadamard matrices. A Hadamard matrix  $H_n$  is a square matrix of size  $n$  such that all its entries are in  $\{1, -1\}$  and  $H_n H_n^T = nI_n$ . Sylvester's construction gives Hadamard matrices  $H_{2^k}$  of size  $2^k$  for all  $k \geq 0$ , where  $H_{2^{k+1}} = \begin{bmatrix} H_{2^k} & H_{2^k} \\ H_{2^k} & -H_{2^k} \end{bmatrix}$  and  $H_{2^0} = 1$ . The Hadamard code  $\mathcal{H}_{2^k}^1$  derived from the Hadamard matrices  $H_{2^k}$  by Sylvester's construction is a code over the binary alphabet  $\{0, 1\}$  of which the  $2^k$  codewords are rows of  $H_{2^k}$  where the entry 1 of  $H_{2^k}$  is mapped to 1 and the entry  $-1$  is mapped to 0. Symmetrically, we can also get the Hadamard code  $\mathcal{H}_{2^k}^0$  by mapping the 1 entry to 0 and  $-1$  to 1. Note that  $\mathcal{H}_{2^k}^1$  contains an all-1 codeword and  $\mathcal{H}_{2^k}^0$  contains an all-0 codeword. We call the former the 1-Hadamard code and the latter the 0-Hadamard code. In standard coding theory notation for block codes,  $\mathcal{H}_{2^k}^1$  and  $\mathcal{H}_{2^k}^0$  are  $[2^k, k, 2^{k-1}]$ -code, which is a binary linear code having block length  $2^k$ , message length  $k$ , and minimum Hamming distance  $2^k/2$ . By removing the all-1 codeword from  $\mathcal{H}_{2^k}^1$ , we get a balanced code, i.e., each codeword has Hamming weight  $2^{k-1}$ , half of its block length. We call it the balanced 1-Hadamard code, denoted by  $\mathcal{H}_{2^k}^{1b}$ . Symmetrically, we can get the balanced 0-Hadamard code, denoted by  $\mathcal{H}_{2^k}^{0b}$ . A binary code  $\mathcal{C}_n$  of block length  $n$  can be viewed equivalently as a 0-1 valued constraint function  $f$  of arity  $n$  by taking the support  $\mathcal{S}(f) = \{\alpha \in \mathbb{Z}_2^n \mid f(\alpha) \neq 0\}$  to be  $\mathcal{C}_n$ . An  $m$ -multiple of the code  $\mathcal{C}_n$  is a code of block length  $nm$ , each codeword of which is obtained from a codeword  $\alpha$  of  $\mathcal{C}_n$  by taking the concatenation of  $m$  copies of  $\alpha$ .

► **Theorem 8.** An EO signature  $f$  is a  $\delta_1$ -affine kernel if and only if  $\mathcal{S}(f)$  is an  $m$ -multiple of a balanced 1-Hadamard code  $\mathcal{H}_{2^k}^{1b}$  for some  $k \geq 3$  and  $m \geq 1$ , or  $\delta_0 \nmid f$  and  $|\mathcal{S}(f)| = 3$ .

A symmetric statement holds for the  $\delta_0$ -affine kernel by switching 0 and 1 everywhere.

Due to the existence of newly discovered non-trivial tractable classes corresponding to the Hadamard code in the  $\#EO$  problem without assuming ARS, the complexity classification for the complex-valued Holant problem is much more challenging. It's hard to imagine how many more tractable classes with remarkable properties are yet to be discovered. The results in this paper are a tiny step towards an ambitious goal. Very recently, independent of this work, Meng, Wang and Xia discover new larger tractable classes for the  $\#EO$  problem [17].

In addition, we note that the presented result in this paper is not the first time that a tractable class for counting problems is found to be related to coding theory. Even in the classification of real-valued Holant problems, a tractable family called local affine [11] was already discovered using the well-known Hamming code. Are there other interesting tractable classes for counting problems that can be carved out with the help of coding theory? In the other direction, it is also worth pursuing whether these new tractability results for counting problems can be applied back to coding theory? We leave the questions in both directions for further study.

The paper is organized as follows. In Section 2, we give some definitions and notations. In Section 3, we give a chain reaction algorithm which establishes the tractability result (Theorem 5) for  $\delta_1$ -affine and  $\delta_0$ -affine signatures. In Section 4, we prove the characterization theorem (Theorem 8) for  $\delta_1$ -affine and  $\delta_0$ -affine kernels. In Section 5, we prove the hardness result Theorem 6. In fact, the proof of the hardness result highly depends on the characterization of  $\delta_1$ -affine and  $\delta_0$ -affine kernels. Omitted proofs can be founded in the full version.

## 2 Preliminaries

A 0-1 valued signature is determined by its support  $\mathcal{S}(f)$ . In the paper, we use  $f$  and  $\mathcal{S}(f)$  interchangeably. We can also view  $\mathcal{S}(f)$  as a binary code of block length  $a(f)$  where each  $\alpha \in \mathcal{S}(f)$  is a codeword. We say  $\mathcal{S}(f)$  (or equivalently  $f$ ) has a constant Hamming weight  $\ell$  if  $\text{wt}(\alpha) = \ell$  for every  $\alpha \in \mathcal{S}(f)$ . For simplicity, we say  $f$  is  $\ell$ -weighted or constant-weighted in general. The following fact can be easily checked. Any factor of a constant-weighted signature is still constant-weighted. If a vertex  $v$  in a graph is labeled by a signature  $f = g \otimes h$ , we can replace the vertex  $v$  by two vertices  $v_1$  and  $v_2$  and label  $v_1$  with the factor  $g$  and  $v_2$  with  $h$ , respectively. The incident edges of  $v$  become incident edges of  $v_1$  and  $v_2$  respectively according to the partition of variables of  $f$  in the tensor product of  $g$  and  $h$ . This does not affect the evaluation of the instance. We represent  $f$  or  $\mathcal{S}(f)$  by a matrix which lists all vectors in  $\mathcal{S}(f)$  by its rows. All matrices that are equal up to a permutation of rows and columns represent the same signature. Normally, the columns indexing variables are ordered from the smallest index to the largest, and are omitted for convenience when the context is clear. For example, the signature  $f = \{1100, 1010, 1001\}$  can be written as  $\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$ .

For  $i \in A(f)$ , we use  $\#_i^0(f)$  and  $\#_i^1(f)$  to denote the number of 0s and 1s respectively in the  $i$ -th column of  $\mathcal{S}(f)$ . The complement of a signature  $f$ , denoted by  $\bar{f}$  is the signature with the support  $\mathcal{S}(\bar{f}) = \{\alpha \in \mathbb{Z}_2^n \mid \bar{\alpha} \in \mathcal{S}(f)\}$  where  $\bar{\alpha}$  is the standard complement of  $\alpha$ , i.e.,  $\bar{\alpha} \oplus \alpha = \bar{0}^n$ . Let  $\hat{f}(x_1, x_2, \dots, x_n) = x_1 x_2 \dots x_n \oplus f(x_1, x_2, \dots, x_n)$ . Then,  $\mathcal{S}(\hat{f}) = \{\bar{1}^n\} \Delta \mathcal{S}(f)$ . Similarly, let  $\check{f}(x_1, x_2, \dots, x_n) = (1 \oplus x_1)(1 \oplus x_2) \dots (1 \oplus x_n) \oplus f(x_1, x_2, \dots, x_n)$ . Then  $\mathcal{S}(\check{f}) = \{\bar{0}^n\} \Delta \mathcal{S}(f)$ . We use  $\neq$  to denote the binary disequality signature  $\{01, 10\}$ . Connecting  $\delta_1$  with the  $i$ -th variable of  $f$  using  $\neq$  is equivalent to pinning the  $i$ -th variable of  $f$  to 0, which gives the signature  $f_i^0$ . Extracting the  $i$ -th variable of  $f$  to 0 gives the signature  $(x_i \oplus 1)f(x_1, x_2, \dots, x_n)$  of arity  $n$ , denoted by  $f^{x_i=0}$ . Clearly  $f^{x_i=0} = \delta_0 \otimes f_i^0$ . Similarly we define  $f_i^1$  and  $f^{x_i=1}$ . We use  $f_{ij}^{ab}$  to denote the signature  $(f_i^a)_j^b = (f_j^b)_i^a$  for distinct  $i, j \in A(f)$  and any  $a, b \in \mathbb{Z}_2$ . The signature  $f^{x_i \neq x_j}$  is defined to be  $f_{ij}^{01} + f_{ij}^{10}$  which can be obtained by connecting the  $i$ -th and  $j$ -th variables of  $f$  using  $\neq$ . For a signature  $f$ , we define its  $m$ -multiple, denoted by  $f_{\times m}$  to be the signature  $\mathcal{S}(f_{\times m}) = \{\bar{\alpha}^m \mid \alpha \in \mathcal{S}(f)\}$  where  $\bar{\alpha}^m \in \mathbb{Z}_2^{nm}$  is obtained from  $\alpha$  by taking the concatenation of  $m$  copies of  $\alpha$ .

Now we give some basic properties of affine signatures. Equivalently,  $f$  is affine if and only if it satisfies the property that for any  $\alpha, \beta, \gamma \in \mathcal{S}(f)$ ,  $\alpha \oplus \beta \oplus \gamma \in \mathcal{S}(f)$ . One can also check that affine signatures are closed under pinning, extracting, tensor product and factoring. Also,  $f \in \mathcal{A}$  if and only if  $f_{\times m} \in \mathcal{A}$  for any  $m \in \mathbb{Z}_+$ .

► **Lemma 9.** *If a signature  $f \in \mathcal{A}$  is constant-weighted and  $\delta_0, \delta_1 \nmid f$ , then  $\#_i^0(f) = \#_i^1(f)$  for any  $i \in A(f)$ . Moreover,  $f$  is an EO signature.*

► **Definition 10** (Pairwise opposite). *Let  $f$  be a signature of arity  $2n$ . We say  $f$  is pairwise opposite if we can partition the  $2n$  variables into  $n$  pairs such that in  $\mathcal{S}(f)$ , two variables of each pair always take opposite values.*

► **Lemma 11** (Lemma 5.7 in [5]). *If  $f$  is an affine EO signature, then  $f$  is pairwise opposite.*

### 3 A chain reaction algorithm

In the section, we show  $\#\text{EO}(\mathcal{D}_1 \cup \mathcal{A})$  and  $\#\text{EO}(\mathcal{D}_0 \cup \mathcal{A})$  are tractable by giving a chain reaction algorithm. We only consider  $\#\text{EO}(\mathcal{D}_1 \cup \mathcal{A})$ , since the other case is symmetric. Given an instance  $\Omega$  of  $\#\text{EO}(\mathcal{D}_1 \cup \mathcal{A})$ , we may assume that there is at least one vertex labelled by a  $\delta_1$ -affine signature  $f$ . Otherwise, all signatures in the instance are affine. Such an instance of  $\#\text{EO}(\mathcal{A})$  is tractable.

Since  $f$  is  $\delta_1$ -affine, consider a factor  $\delta_1$  of  $f$ . This  $\delta_1$  is connected to another variable of some signature (which may be  $f$  itself) using  $\neq_2$ . Then, the connected variable is pinned to 0. A key property that ensures our algorithm work is that after connecting  $\delta_1$  to another variable no matter whether it is a variable of  $f$  itself, an affine signature, or another  $\delta_1$ -affine signature, either the resulting signature is affine, or we can realize another  $\delta_1$ . In the later case, we call it a propagation step. After a propagation step, the new realized  $\delta_1$  can be used to pin another variable to 0. Thus, we get a chain reaction, and it terminates when an instance of  $\#\text{EO}(\mathcal{A})$  is reached. Note that, after each propagation step, the number of edges (total variables) in the instance is reduced by 2. Thus, the chain reaction terminates after at most  $|E|/2$  many steps where  $E$  is the edge set of the underlying graph in  $\Omega$ .

► **Lemma 12.** *Suppose that  $f = \delta_1 \otimes h$  is  $\delta_1$ -affine and  $g \in \mathcal{D}_1 \cup \mathcal{A}$ . For any variable  $x_i$  of  $g$ , by connecting it with  $\delta_1$  of  $f$  using  $\neq_2$ , we can get a zero signature or a signature  $h \otimes \delta_1 \otimes g_{ij}^{01}$  where  $g_{ij}^{01} \in \mathcal{D}_1 \cup \mathcal{A}$  for some  $j \in A(g) \setminus \{i\}$ .*

**Proof.** If  $g \in \mathcal{A}$ , then  $g$  is pairwise opposite by Lemma 11. Suppose  $x_j$  is opposite to  $x_i$ . Then pinning  $x_i$  to 0 makes  $x_j$  be a  $\delta_1$  factor, i.e.,  $g_i^0 = \delta_1(x_j) \otimes g_{ij}^{01}$ , where  $g_{ij}^{01} \in \mathcal{A}$ . Otherwise  $g \in \mathcal{D}_1 \setminus \mathcal{A}$ . Suppose  $g = \delta_1(x_j) \otimes g'$  for some variable  $x_j$ . If  $x_i = x_j$ , then by connecting the  $\delta_1$  factor of  $f$  and  $x_j$  of  $g$  using  $\neq$ , we get a zero signature. Now assume  $x_i \neq x_j$ . Then  $(g')_i^0 \in \mathcal{A} \cup \mathcal{D}_1$  since  $g \in \mathcal{D}_1$ . In this case, we get  $h \otimes \delta_1(x_j) \otimes g_{ij}^{01}$ , where  $g_{ij}^{01} = (g')_i^0 \in \mathcal{A} \cup \mathcal{D}_1$ . ◀

► **Theorem 13.**  *$\#\text{EO}(\mathcal{A} \cup \mathcal{D}_1)$  and  $\#\text{EO}(\mathcal{A} \cup \mathcal{D}_0)$  are tractable.*

**Proof.** We only prove  $\#\text{EO}(\mathcal{A} \cup \mathcal{D}_1)$  is tractable, the other case is symmetric.

Consider an instance  $\Omega$  of  $\#\text{EO}(\mathcal{A} \cup \mathcal{D}_1)$ . If all signatures in  $\Omega$  are affine, then we are done. Otherwise, there is a vertex  $u$  in  $\Omega$  that is labeled by a  $\delta_1$ -affine signature  $f = \delta_1 \otimes h$ . Without loss of generality, we name the variable associated with  $\delta_1$  after  $x_1$ . Suppose that in  $\Omega$ ,  $x_1$  is connected to another variable  $x_i$  of some  $g$  labeling the vertex  $v$  using  $\neq_2$ .

- If  $u = v$ , i.e.,  $f$  and  $g$  are labeled on the same vertex in  $\Omega$ , then by definition the resulting signature  $f^{x_1 \neq x_i} = f_{1i}^{10} = h_i^0$  obtained from  $f$  by connecting variables  $x_1$  and  $x_i$  is in  $\mathcal{D}_1 \cup \mathcal{A}$ . Then, the instance is reduced to a smaller instance with two fewer edges where  $f$  is replaced by  $f^{x_1 \neq x_i}$  of smaller arity. Since  $f^{x_1 \neq x_i} \in \mathcal{D}_1 \cup \mathcal{A}$ , the reduced instance is still an instance of  $\#EO(\mathcal{A} \cup \mathcal{D}_1)$ .
  - Otherwise,  $f$  and  $g$  are labeled on two different vertices. By Lemma 12, connecting  $\delta_1$  of  $f$  with a variable  $x_i$  of  $g$ , we get a zero signature or a signature  $h \otimes \delta_1 \otimes g'$  where  $g' = g_{ij}^{01} \in \mathcal{D}_1 \cup \mathcal{A}$  for some  $j \in A(g) \setminus \{i\}$ . If a zero signature is realized, then clearly we are done. Otherwise, by renaming variables and decomposing  $h \otimes \delta_1 \otimes g'$  into two parts  $f = h \otimes \delta_1$  and  $g'$ , again we can reduce the original instance to a smaller instance with two fewer edges where the edges incident to  $f$  is changed and  $g$  is replaced by  $g'$  of smaller arity. Still, since  $g' \in \mathcal{D}_1 \cup \mathcal{A}$ , the reduced instance is still an instance of  $\#EO(\mathcal{A} \cup \mathcal{D}_1)$ .
- Thus, in both cases, the instance can be solved recursively and the number of recursions is bounded by  $|E|/2$  where  $E$  is the edge set of the underlying graph of  $\Omega$ . ◀

#### 4 Characterization of $\delta_1$ -affine $\delta_0$ -affine kernels

In this section, we prove the main characterization theorem for  $\delta_1$ -affine and  $\delta_0$ -affine kernels using the Hadamard code. We first give an equivalent definition for the  $\delta_1$ -affine kernel. Note that in Definition 7 we require  $h \notin \mathcal{A}$ , while in the following lemma we require  $\delta_0 \nmid h$ .

► **Lemma 14.** *An EO signature  $f \in K(\mathcal{D}_1)$  if and only if it satisfies the following two properties:*

1.  $f = \delta_1^{\otimes m} \otimes h$  for some  $m \in \mathbb{Z}_+$  where  $h$  is a signature of positive arity,  $\delta_1, \delta_0 \nmid h$ ;
2. for every  $i \in A(h)$ ,  $h_i^0 \in \mathcal{A}$ .

Note that the signature  $f_2$  in Example 2 is a  $\delta_1$ -affine kernel. Trivially, any EO signature of support size 3 with at least one  $\delta_1$  factor and no  $\delta_0$  factor is a  $\delta_1$ -affine kernel, since any signature obtained from it by pinning a variable to 0 has a support of size 1 or 2, which is affine. We say  $f \in K(\mathcal{D}_1)$  (or  $K(\mathcal{D}_0)$ ) is a trivial kernel if  $|\mathcal{S}(f)| = 3$ . Below, we prove the characterization theorem for non-trivial  $\delta_1$ -affine and  $\delta_0$ -affine kernels, which is essentially equivalent to Theorem 8.

► **Theorem 15.** *An EO signature  $f$  is a non-trivial  $\delta_1$ -affine (or  $\delta_0$ -affine) kernel if and only if  $\mathcal{S}(f)$  is an  $m$ -multiple of a balanced 1-Hadamard (or balanced 0-Hadamard respectively) code of size  $2^k - 1$  for some  $k \geq 3$  and  $m \geq 1$ .*

We give a proof outline for this theorem. First, we prove  $\delta_1$ -affine kernels admit certain arities and support sizes by carefully counting the number of 0s and 1s in each column of their supports. More specifically, we prove a non-trivial  $\delta_1$ -affine kernel  $f$  must have support size  $2^k - 1$  and arity  $m2^k$ , for some  $m, k \in \mathbb{Z}_+, k \geq 3$  (Lemma 17). Then, we define basic  $\delta_1$ -affine kernels (Definition 18) of support size  $2^k - 1$  and arity  $2^k$ . We show every non-trivial  $\delta_1$ -affine kernel is a multiple of some non-trivial basic kernel (Lemma 19). Finally, we characterize the basic  $\delta_1$ -affine kernel by the balanced 1-Hadamard code  $\mathcal{H}_{2^k}^{1b}$ . We define a special affine signature called butterfly (Definition 20), serving as a bridge connecting basic kernels and balanced Hadamard codes. We show  $\mathcal{H}_{2^k}^1$  (or  $\mathcal{H}_{2^k}^0$ ) is just a half of butterfly (Lemma 24). We also define left and right wings of butterfly (Definition 21) by removing the all-0 (or all-1) row from a half of butterfly. Notice  $\mathcal{H}_{2^k}^{0b}$  (or  $\mathcal{H}_{2^k}^{1b}$ ) is obtained by removing the all-0 (or all-1) codeword from  $\mathcal{H}_{2^k}^1$ . Thus, wings of butterfly coincide with  $\mathcal{H}_{2^k}^{0b}$  and  $\mathcal{H}_{2^k}^{1b}$  (Lemma 24). Finally, we prove basic kernels are precisely wings of butterfly (Lemma 26 and Lemma 28). Therefore, basic kernels, balanced Hadamard codes and wings of butterfly are precisely the same things.

► **Lemma 16.** *Suppose  $f \in K(\mathcal{D}_1)$ . If  $\delta_0 \nmid f^{x_i=1}$  for some  $i \in A(f)$ , then  $f^{x_i=1} \in K(\mathcal{D}_1)$ .*

► **Lemma 17.** *Suppose that  $f = \delta_1^{\otimes m} \otimes h \in K(\mathcal{D}_1)$  where  $m \geq 1$ ,  $\delta_1, \delta_0 \nmid h$ , and  $|\mathcal{S}(f)| > 3$ . Then there exists some  $k \geq 2$  such that  $|\mathcal{S}(f)| = 2^{k+1} - 1$  and  $a(f) = m2^{k+1}$ . Moreover, we have  $\delta_1 \nmid h_i^0, \delta_0 \nmid h_i^1$  and  $\#_i^0(h) = 2^k$  for every  $i \in A(h)$ .*

► **Definition 18** (Basic  $\delta_1$ -affine kernel). *A basic  $\delta_1$ -affine kernel  $f_k$  of order  $k$  (or a  $k$ -th basic  $\delta_1$ -affine kernel) is a  $\delta_1$ -affine kernel of arity  $2^k$  and support size  $2^k - 1$ .*

For example, the basic  $\delta_1$ -affine kernel of order 1 is  $f_1 = \delta_1 \otimes \delta_0$  and the basic  $\delta_1$ -affine kernel of order 2 is  $f_2 = \delta_1 \otimes \{(001), (010), (100)\}$ .

► **Lemma 19.** *An EO signature of arity  $m2^k$  is a non-trivial  $\delta_1$ -affine kernel if and only if it is an  $m$ -multiple of a non-trivial basic  $\delta_1$ -affine kernel of arity  $2^k$ .*

**Proof.** First we assume  $f = \delta_1 \otimes g$  is a non-trivial basic  $\delta_1$ -affine kernel. Then  $|\mathcal{S}(f_{\times m})| = |\mathcal{S}(f)| > 3$ . We want to show  $f_{\times m} \in K(\mathcal{D}_1)$ . Note that  $f_{\times m} = \delta_1^{\otimes m} \otimes g_{\times m}$  and  $\delta_1, \delta_0 \nmid g_{\times m}$ . Pinning 0 to the  $i$ -th variable of one copy of  $g$  in  $g_{\times m}$ , each of the remaining  $m - 1$  copies contributes a  $\delta_0$  factor. i.e.,  $(g_{\times m})_i^0 = \delta_0^{\otimes(m-1)} \otimes (g_i^0)_{\times m}$ . Clearly  $(g_{\times m})_i^0 \in \mathcal{A}$  since  $g_i^0 \in \mathcal{A}$ . It follows that  $f_{\times m} \in K(\mathcal{D}_1)$ .

Conversely, assume  $f = \delta_1^{\otimes m} \otimes h$  is a non-trivial  $\delta_1$ -affine kernel, where  $m \in \mathbb{Z}_+$ ,  $\delta_1, \delta_0 \nmid h$ . We want to show there exists a non-trivial basic  $\delta_1$ -affine kernel  $f_0$  such that  $f = (f_0)_{\times m}$ . By Lemma 17, we have  $|\mathcal{S}(f)| = 2^k - 1$  and  $\#_i^0(h) = 2^{k-1}$  for every  $i \in A(h)$ . Fix any  $i \in A(h)$ , consider  $f^{x_i=0} = \delta_1^{\otimes m} \otimes h^{x_i=0}$ . It is an affine EO signature of support size  $2^{k-1}$ . By Lemma 17,  $\delta_1 \nmid h^{x_i=0}$ . Factoring out  $\delta_0$  in  $h^{x_i=0}$ , we suppose  $h^{x_i=0} = \delta_0^{\otimes m_0} \otimes h'$ ,  $\delta_1, \delta_0 \nmid h'$ . At the same time, notice  $h'$  is constant-weighted. Thus,  $h'$  is an EO signature by Lemma 9. Since  $f^{x_i=0} = \delta_1^{\otimes m} \otimes \delta_0^{\otimes m_0} \otimes h'$  is also EO, we have  $m = m_0$ .

On one hand, if  $x_j$  is a variable taking constant 0 in  $\mathcal{S}(f^{x_i=0})$ , then the values of  $x_j$  and  $x_i$  must be identical in  $\mathcal{S}(h)$ , because the  $j$ -th column already contributes  $2^{k-1}$  many 0s in  $\mathcal{S}(f^{x_i=0})$  and the remaining  $2^{k-1} - 1$  rows must be all 1s. On the other, if  $x_j$  is a variable of  $h'$ , then  $x_j$  and  $x_i$  are not identical in  $\mathcal{S}(h)$  since they are not identical in  $\mathcal{S}(h^{x_i=0})$ . Therefore, there are exactly  $m_0 = m$  copies of the  $i$ -th column in  $\mathcal{S}(h)$ . Since  $i \in A(h)$  is arbitrary, we have  $h = (h_0)_{\times m}$  for some signature  $h_0$ . Clearly  $\delta_1, \delta_0 \nmid h_0$  since  $\delta_1, \delta_0 \nmid h$ . Notice for every  $i \in A(h_0)$ ,  $(h_0^{x_i=0})_{\times m} = ((h_0)_{\times m})^{x_i=0} = h^{x_i=0} = \delta_0 \otimes h_i^0 \in \mathcal{A}$ , so  $h_0^{x_i=0} \in \mathcal{A}$  and then  $(h_0)_i^0 \in \mathcal{A}$ . Therefore,  $f_0 = \delta_1 \otimes h_0$  is a basic  $\delta_1$ -affine kernel and  $f = (f_0)_{\times m}$ . ◀

Below, we show balanced Hadamard codes characterize basic kernels via the bridge of butterfly.

► **Definition 20** (Butterfly). *Let  $k \geq 1$ . An affine signature  $g$  of arity  $2^{k+1}$  is a butterfly and denoted by  $B_k$ , if it has exactly  $k$  free (linearly independent) variables  $x_1, x_2, \dots, x_k$  and any two different variables do not always take identical values in  $\mathcal{S}(g)$ .*

The butterfly  $B_k$  is unique up to a permutation of variables. It is precisely the affine signature of  $k$  free variables with the maximum arity, assuming any two different variables are not identical.

► **Definition 21** (Wings of butterfly). *We represent the butterfly  $B_k$  by a matrix in the following way. Let  $(x_2, x_3, \dots, x_{k+1})$  be free variables of  $B_k$ , and  $x_1$  be constant 0. Moreover, let  $x_m = \lambda_2 x_2 + \lambda_3 x_3 + \dots + \lambda_{k+1} x_{k+1}$  and  $x_{m+2^k} = x_m + 1$  for  $1 \leq m \leq 2^k$ , where  $\lambda_2, \dots, \lambda_{k+1} \in \mathbb{Z}_2$ . Through a permutation of rows, we let the first row be  $0^{2^k} 1^{2^k}$ .*



Then,  $B_k$  can be split into the left half and the right half, each of which consisting of  $2^k$  columns. The left wing of  $B_k$ , denoted by  $L_k$  is the signature obtained from the left half of  $B_k$  by removing the first row  $\vec{0}^{2^k}$ . Symmetrically, we define the right wing of  $B_k$  denoted by  $R_k$ .

► **Remark 22.** Recall that  $\widetilde{L}_k$  and  $\widehat{R}_k$  are obtained from  $L_k$  and  $R_k$  by adding an all-0 or all-1 row respectively. They are the left and right halves of  $B_k$  respectively which are affine. In fact,  $\widetilde{L}_k$  is precisely the affine signature of  $k$  free variables with maximum arity, assuming every variable is a linear combination of free variables and any two variables are not identical. While  $\widehat{R}_k$  is the complement of  $\widetilde{L}_k$ .

► **Example 23.** We give a butterfly with 2 free variables  $x_2, x_3$ .

$$B_2 = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}.$$

The other variables are decided by the following linear or affine linear combinations.  $x_1 = 0x_2 + 0x_3$ ,  $x_4 = x_2 + x_3$ ,  $x_5 = 0x_2 + 0x_3 + 1$ ,  $x_6 = x_2 + 0x_3 + 1$ ,  $x_7 = 0x_2 + x_3 + 1$ ,  $x_8 = x_2 + x_3 + 1$ .

The left and right wings of  $B_2$  are  $L_2 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$  and  $R_2 = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}$ .

► **Lemma 24.** As a signature, the balanced 1-Hadamard (or 0-Hadamard) code is the right (or left) wing of butterfly up to a permutation of variables.

**Proof.** We only need to prove  $\mathcal{H}_{2^k}^0$  is the left half of  $B_k$ . Note that  $\mathcal{H}_{2^k}^0$  is a linear code, then it is affine as a signature. By the definition of Hadamard matrix and Hadamard code, every two different rows of  $\mathcal{H}_{2^k}^0$  do not take identical values. By Sylvester's construction,  $\mathcal{H}_{2^k}^0$  is symmetric as a matrix. Thus, every two different variables of  $\mathcal{H}_{2^k}^0$  do not take identical values in its support. Because  $\mathcal{H}_{2^k}^0$  is of arity  $2^k$ , support size  $2^k$ , and has an all-0 vector in its support, it is precisely the affine signature of  $k$  free variables with maximum arity, assuming every variable is a linear combination of free variables and any two variables are not identical. Therefore,  $\mathcal{H}_{2^k}^{0b} = L_k$  is just removing the all-0 row of  $\mathcal{H}_{2^k}^0$ . ◀

So far we show the equivalence of balanced Hadamard code and wings of butterfly. Next we focus on the relationship between basic kernels and wings of butterfly. For convenience, we only consider the basic  $\delta_1$ -affine kernel and the right wing of butterfly in the following. The other case is symmetric.

► **Lemma 25.** Let  $k \geq 3$  and  $R_k$  be the right wing of  $B_k$ , then  $R_k$  is an EO signature. Moreover, every two different rows (or columns) of  $R_k$  do not take identical or opposite values.

**Proof.** Because  $\mathcal{H}_{2^k}^{1b}$  is an EO signature and every two different rows and columns do not take identical or opposite values, this lemma is true because  $R_k = \mathcal{H}_{2^k}^{1b}$  by Lemma 24. ◀

► **Lemma 26.** Let  $R_k$  be the right wing of a butterfly. Then  $R_k$  is the  $k$ -th basic  $\delta_1$ -affine kernel.

**Proof.** For  $k = 1, 2$  it can be checked by definition. Now we assume  $k \geq 3$ . First by Lemma 25  $R_k$  is EO. Then we show  $R_k \in K(\mathcal{D}_1)$ . Notice  $R_k = \delta_1 \otimes r_k$ ,  $\delta_0, \delta_1 \uparrow r_k$ , and  $(r_k)_i^0 = (\widehat{r}_k)_i^0 \in \mathcal{A}$ , since  $\widehat{r}_k \in \mathcal{A}$ . Finally, because  $R_k$  is of arity  $2^k$  and support size  $2^k - 1$ , it is a basic  $\delta_1$ -affine kernel of order  $k$ . ◀

## 86:10 Eulerian Orientations and Hadamard Codes

Below, we prove the more complicated direction that a basic  $\delta_1$ -affine kernel of order  $k$  is just  $R_k$  up to a permutation of variables. We prove this by induction. We first prove certain properties for the  $k+1$ -th  $\delta_1$ -affine kernel assuming the  $k$ -th  $\delta_1$ -affine kernel is unique up to a permutation of variables.

► **Lemma 27.** *Suppose  $k \in \mathbb{Z}_+, k \geq 2$ . If the  $k$ -th basic  $\delta_1$ -affine kernel  $f_k$  is unique up to a permutation of variables, then any  $(k+1)$ -th basic  $\delta_1$ -affine kernel  $f = \delta_1 \otimes h$  satisfies the following properties for every  $l \in A(h)$ .*

1.  $f^{x_i=1} = (f_k)_{\times 2}$  up to a permutation of variables.
2.  $f^{x_i=0} = B_k$  up to a permutation of variables.
3. Two variables  $x_i$  and  $x_j$  take opposite values in  $\mathcal{S}(f^{x_i=0})$  if and only if they take identical values in  $\mathcal{S}(f^{x_i=1})$ .

► **Lemma 28.** *The  $k$ -th basic  $\delta_1$ -affine kernel  $f_k$  is precisely  $R_k$  up to a permutation of variables.*

**Proof.** For  $k = 1$ , we have  $f_1 = \delta_1 \otimes \delta_0$  and this lemma is clear. For  $k = 2$ ,  $f_2 = \delta_1 \otimes \{(001), (010), (100)\}$  and this lemma is true by Example 23. Now assume this lemma holds for positive integers less or equal than  $k (k \geq 2)$ . By Lemma 27, we know any  $(k+1)$ -th  $\delta_1$ -affine kernel  $f_{k+1} = \delta_1 \otimes h_{k+1}$  satisfies the following three properties:

$$f_{k+1} = \begin{array}{|c|c|} \hline 0 & \cdots & 0 & 1 & \cdots & 1 \\ \hline \overline{f_k} & & f_k & & & \\ \hline f'_k & & f'_k & & & \\ \hline \end{array}$$

1. Extracting any variable of  $h_{k+1}$  to 0, up to a permutation of rows, the first  $2^k$  rows of  $f_{k+1}$  is butterfly  $B_k$ .
2. Extracting any variable of  $h_{k+1}$  to 1, up to a permutation of rows, the last  $2^k - 1$  rows is  $(f_k)_{\times 2}$ . However, when we write the above butterfly into the specific form, we fix the order of columns. Then we denote the last  $2^k - 1$  rows by two  $f'_k$ , where  $f'_k = f_k$  up to a permutation of rows and columns.
3. Every two different variables take identical values in  $\mathcal{S}((f_{k+1})^{x_i=1})$  if and only if they take opposite values in  $\mathcal{S}((f_{k+1})^{x_i=0})$ ,  $i \in A(h_{k+1})$ .

Because every two different columns of  $f_k$  (as well as  $f'_k$ ) do not take identical or opposite values by Lemma 25, we can easily check this proposition still holds for  $f_{k+1}$ . Next we show the signature  $\widehat{f_{k+1}}$  is affine. We first notice  $\widehat{f_{k+1}}$  has the following properties, which holds for every  $i \in A(\widehat{f_{k+1}})$ :

1.  $(\widehat{f_{k+1}})^{x_i=0} = (f_{k+1})^{x_i=0} = B_k \in \mathcal{A}$ .
2.  $(\widehat{f_{k+1}})^{x_i=1} = (\widehat{f_k})_{\times 2} \in \mathcal{A}$ , since  $f_k$  is the right wing of  $B_k$  by induction hypothesis, hence  $\widehat{f_k}$  is the right half of  $B_k$ , which is affine.

We prove  $\widehat{f_{k+1}} \in \mathcal{A}$  by contradiction. If  $\widehat{f_{k+1}} \notin \mathcal{A}$ , there exists  $\alpha', \beta', \gamma' \in \mathcal{S}(\widehat{f_{k+1}})$ , such that  $\theta' = \alpha' \oplus \beta' \oplus \gamma' \notin \mathcal{S}(\widehat{f_{k+1}})$ . If there exists a position  $j$  such that  $\alpha'_j = \beta'_j = \gamma'_j = 0$ , assume  $j = 1$  without loss of generality. Then  $\alpha' = 0\alpha, \beta' = 0\beta, \gamma' = 0\gamma, \theta' = 0\theta$ , where  $\alpha, \beta, \gamma \in \mathcal{S}((\widehat{f_{k+1}})_j^0)$  and  $\theta = \alpha \oplus \beta \oplus \gamma$ . Because  $(\widehat{f_{k+1}})_j^0 \in \mathcal{A}$ , we have  $\theta \in \mathcal{S}((\widehat{f_{k+1}})_j^0)$ .

Then  $\theta' = 0\theta \in \mathcal{S}(\widehat{f_{k+1}})$ , contradiction! Similarly, if there exists a position  $j$  such that  $\alpha'_j = \beta'_j = \gamma'_j = 1$ , we also get a contradiction. Then, by enumerating all possible combinations of  $\alpha', \beta', \gamma' \in \mathcal{S}(\widehat{f_{k+1}})$ , we can show  $\widehat{f_{k+1}} \in \mathcal{A}$ . Because  $\mathcal{S}(\widehat{f_{k+1}})$  contains an all-1 vector;  $|\mathcal{S}(\widehat{f_{k+1}})| = a(\widehat{f_{k+1}}) = 2^{k+1}$ ; and there is no identical pair in  $\mathcal{S}(\widehat{f_{k+1}})$ ,  $f_{k+1}$  is precisely the right wing of  $B_{k+1}$ . ◀

So far by Lemma 26 and Lemma 28 we prove the equivalence between basic kernels and wings of butterfly. Combining with Lemma 24 we get the proof of Theorem 15.

## 5 Hardness result

Recall that the binary disequality signature ( $\neq$ ) is  $\{10, 01\}$ . The symbol " $\leq_T$ " stands for Turing reduction.

► **Definition 29** (ARS). *A signature  $f$  satisfies ARS if  $f(\alpha) = f(\bar{\alpha})$  for every  $\alpha \in \mathcal{S}(f)$ .*

First, we show that  $\delta_1 \otimes \delta_0$  is always realizable from an EO signature not satisfying ARS.

► **Lemma 30.** *If an EO signature  $f$  does not satisfy ARS, then  $\#EO(f, \delta_1 \otimes \delta_0) \leq_T \#EO(f)$ .*

With the help of  $\delta_1 \otimes \delta_0$ , we have the following reduction.

► **Lemma 31.** *If an EO signature  $f$  does not satisfy ARS, then  $\#EO(\{f, f^{x_i=1}, f^{x_i=0}\}) \leq_T \#EO(f)$  for every  $i \in A(f)$ .*

Next we will prove  $\#EO(f, g)$  is  $\#P$ -hard if  $f \in \mathcal{D}_1 \setminus \mathcal{A}$  and  $g \in \mathcal{D}_0 \setminus \mathcal{A}$ , which implies that  $\#EO(\mathcal{D}_1 \cup \mathcal{D}_0)$  is  $\#P$ -hard. We first consider a basic case where  $f$  and  $g$  are basic  $\delta_1$ -affine and  $\delta_0$ -affine kernels of order 2 respectively.

► **Lemma 32.** *Let  $f_2(x_1, x_2, x_3, x_4) = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$  and  $g_2(y_1, y_2, y_3, y_4) = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$ . Then  $\#EO(f_2, g_2)$  is  $\#P$ -hard.*

**Proof.** Connecting the variables  $x_1$  and  $y_1$ ,  $x_2$  and  $y_2$  using  $\neq$  respectively, we realize a signature  $h$  of arity 4 where

$$h = \begin{bmatrix} & x_3 & x_4 & y_3 & y_4 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}.$$

It is known that  $\#EO(h)$  is  $\#P$ -hard by the complexity classification for the six-vertex model [7]. Then,  $\#EO(f_2, g_2)$  is  $\#P$ -hard. ◀

This  $\#P$ -hardness result can be generalized to any multiples of  $f_2$  and  $g_2$ .

► **Lemma 33.** *Let  $f = (f_2)_{\times a}, g = (g_2)_{\times b}$ , where  $a, b \in \mathbb{Z}_+$ . Then  $\#EO(f, g)$  is  $\#P$ -hard.*

**Proof.** Connect  $f$  and  $g$  using  $\neq$  block by block. That is, connect variables  $x_i$  and  $y_i$  using  $\neq$  for  $1 \leq i \leq 4$ , where  $x_i, y_i$  are variables of one block  $f_2$  and  $g_2$  as denoted in Lemma 32.

## 86:12 Eulerian Orientations and Hadamard Codes

If  $a = b = 1$ , it is the case in Lemma 32. If  $a = b > 1$ , we can realize

$$h' = \begin{bmatrix} & x_3 & x_4 & y_3 & y_4 \\ 0 & 0 & 1 & 1 & \\ 1 & 0 & 0 & 1 & \\ 0 & 1 & 1 & 0 & \end{bmatrix}$$

which is also #P-hard by the complexity classification of the six vertex model.

If  $a \neq b$ , we may assume  $a > b$  without loss of generality. Then we can realize  $f' = (f_2)_{\times(a-b)}$ . Then connect  $f'$  and  $g$  block by block and continue this process. We finally realize  $(f_2)_{\times d}$  and  $(g_2)_{\times d}$  where  $d = \gcd(a, b)$  is computed by the Euclidean algorithm. Then it is reduced to the case  $a = b$ . ◀

This #P-hardness result can still be generalized to any basic  $\delta_1$ -affine and  $\delta_0$ -affine kernels of support size 3.

► **Lemma 34.** *Let  $f \in \mathcal{D}_1 \setminus \mathcal{A}$ ,  $g \in \mathcal{D}_0 \setminus \mathcal{A}$ , and  $|\mathcal{S}(f)| = |\mathcal{S}(g)| = 3$ . Then  $\#EO(f, g)$  is #P-hard.*

► **Theorem 35.** *If  $f \in \mathcal{D}_1 \setminus \mathcal{A}$ ,  $g \in \mathcal{D}_0 \setminus \mathcal{A}$ , then  $\#EO(f, g)$  is #P-hard.*

**Proof.** We first show we can realize a signature  $f' \in \mathcal{D}_1 \setminus \mathcal{A}$  of support size 3 if  $|\mathcal{S}(f)| > 3$ . If  $f \in K(\mathcal{D}_1)$ , suppose  $f = (f_k)_{\times m}$  by Lemma 19, where  $f_k$  is the  $\delta_1$ -affine kernel of order  $k$ . We can extract 1 in some variable  $x_i$  of  $f$  by Lemma 31 and realize  $f^{x_i=1}$ , which is a multiple of  $f_{k-1}$  by Lemma 27. Continuing this process we can finally realize multiple of  $f_2$ . If  $f \notin K(\mathcal{D}_1)$ , suppose  $f = \delta_1 \otimes h$ . Then there exists  $i \in A(h)$  such that  $h_i^0 \in \mathcal{D}_1 \setminus \mathcal{A}$ . We have  $3 < |\mathcal{S}(h_i^0)| < |\mathcal{S}(f)|$ . Repeating the process, either  $h_i^0 \in K(\mathcal{D}_1)$  or it can realize a signature in  $\mathcal{D}_1 \setminus \mathcal{A}$  of smaller support size. Finally we can realize  $f' \in \mathcal{D}_1 \setminus \mathcal{A}$  of support size 3. Similarly, we can realize  $g' \in \mathcal{D}_0 \setminus \mathcal{A}$  of support size 3. Then by Lemma 34 we prove  $\#EO(f, g)$  is #P-hard. ◀

---

### References

- 1 Miriam Backens. A new Holant dichotomy inspired by quantum computation. In *Proceedings of the 44th International Colloquium on Automata, Languages, and Programming (ICALP 2017)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.ICALP.2017.16.
- 2 Miriam Backens. A complete dichotomy for complex-valued Holant<sup>c</sup>. In *Proceedings of the 45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.ICALP.2018.12.
- 3 Norman Biggs, E Keith Lloyd, and Robin J Wilson. *Graph Theory, 1736-1936*. Oxford University Press, 1986.
- 4 Andrei Bulatov, Martin Dyer, Leslie Ann Goldberg, Markus Jalsenius, and David Richerby. The complexity of weighted Boolean #CSP with mixed signs. *Theoretical Computer Science*, 410(38-40):3949–3961, 2009. doi:10.1016/J.TCS.2009.06.003.
- 5 Jin-Yi Cai, Zhiguo Fu, and Shuai Shao. Beyond #CSP: A dichotomy for counting weighted eulerian orientations with ars. *Information and Computation*, 275:104589, 2020. doi:10.1016/J.IC.2020.104589.
- 6 Jin-Yi Cai, Zhiguo Fu, and Shuai Shao. From Holant to quantum entanglement and back. In *Proceedings of the 47th International Colloquium on Automata, Languages, and Programming (ICALP 2020)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ICALP.2020.22.

- 7 Jin-Yi Cai, Zhiguo Fu, and Mingji Xia. Complexity classification of the six-vertex model. *Information and Computation*, 259:130–141, 2018. doi:10.1016/J.IC.2018.01.003.
- 8 Jin-Yi Cai, Heng Guo, and Tyson Williams. A complete dichotomy rises from the capture of vanishing signatures. *SIAM Journal on Computing*, 45(5):1671–1728, 2016. doi:10.1137/15M1049798.
- 9 Jin-Yi Cai, Pinyan Lu, and Mingji Xia. Dichotomy for Holant\* problems of Boolean domain. In *Proceedings of the 22nd annual ACM-SIAM symposium on Discrete Algorithms*, pages 1714–1728. SIAM, 2011.
- 10 Jin-Yi Cai, Pinyan Lu, and Mingji Xia. The complexity of complex weighted Boolean #CSP. *Journal of Computer and System Sciences*, 80(1):217–236, 2014. doi:10.1016/J.JCSS.2013.07.003.
- 11 Jin-Yi Cai, Pinyan Lu, and Mingji Xia. Dichotomy for real Holant<sup>c</sup> problems. In *Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1802–1821. SIAM, 2018. doi:10.1137/1.9781611975031.118.
- 12 Nadia Creignou and Miki Hermann. Complexity of generalized satisfiability counting problems. *Information and computation*, 125(1):1–12, 1996. doi:10.1006/INCO.1996.0016.
- 13 Martin Dyer, Leslie Ann Goldberg, and Mark Jerrum. The complexity of weighted Boolean #CSP. *SIAM Journal on Computing*, 38(5):1970–1986, 2009. doi:10.1137/070690201.
- 14 Leonhard Euler. Solutio problematis ad geometriam situs pertinentis. *Commentarii academiae scientiarum Petropolitanae*, pages 128–140, 1741.
- 15 Michel Las Vergnas. On the evaluation at (3, 3) of the Tutte polynomial of a graph. *Journal of Combinatorial Theory, Series B*, 45(3):367–372, 1988. doi:10.1016/0095-8956(88)90079-2.
- 16 Jiabao Lin and Hanpin Wang. The complexity of Boolean Holant problems with nonnegative weights. *SIAM Journal on Computing*, 47(3):798–828, 2018. doi:10.1137/17M113304X.
- 17 Boning Meng, Juqiu Wang, and Mingji Xia. P-time algorithms for typical #EO problems. *arXiv preprint*, 2024. doi:10.48550/arXiv.2410.11557.
- 18 Milena Mihail and Peter Winkler. On the number of Eulerian orientations of a graph. *Algorithmica*, 16(4-5):402–414, 1996. doi:10.1007/BF01940872.
- 19 Linus Pauling. The structure and entropy of ice and of other crystals with some randomness of atomic arrangement. *Journal of the American Chemical Society*, 57(12):2680–2684, 1935.
- 20 Franz Rys. Über ein zweidimensionales klassisches Konfigurationsmodell. In *Helvetica Physica Acta*, volume 36(5), page 537, 1963.
- 21 Shuai Shao and Jin-Yi Cai. A dichotomy for real Boolean Holant problems. In *Proceedings of the 61st IEEE Annual Symposium on Foundations of Computer Science (FOCS 2020)*, pages 1091–1102. IEEE, 2020. doi:10.1109/FOCS46700.2020.00105.
- 22 John C Slater. Theory of the transition in  $\text{KH}_2\text{PO}_4$ . *The Journal of Chemical Physics*, 9(1):16–33, 1941.