Available online at www.sciencedirect.com

ScienceDirect

Theoretical Computer Science 384 (2007) 111–125

Theoretical Computer Science

www.elsevier.com/locate/tcs

Computational complexity of counting problems on 3-regular planar graphs

Mingji Xia*, Peng Zhang, Wenbo Zhao

State Key Lab. of Computer Science, Institute of Software, Chinese Academy of Sciences, P.O. Box 8717, Beijing 100080, China
Graduate University of Chinese Academy of Sciences, Beijing, China

Abstract

A variety of counting problems on 3-regular planar graphs are considered in this paper. We give a sufficient condition which guarantees that the coefficients of a homogeneous polynomial can be uniquely determined by its values on a recurrence sequence. This result enables us to use the polynomial interpolation technique in high dimension to prove the #P-completeness of problems on graphs with special requirements. Using this method, we show that #3-Regular Bipartite Planar Vertex Covers is #P-complete. Furthermore, we use Valiant's Holant Theorem to construct a holographic reduction from it to #2,3-Regular Bipartite Planar Matchings, establishing the #P-completeness of the latter. Finally, we completely classify the problems #Planar Read-twice 3SAT with different ternary symmetric relations according to their computational complexity, by giving several more applications of holographic reduction in proving the #P-completeness of the corresponding counting problems.

Keywords: #P-completeness; Holographic reduction; Vertex cover; Matching

1. Introduction

Counting complexity is a natural extension of the complexity of decision problems. It is an active area and has received much attention since Valiant [15] introduced the complexity class #P. Many #P-complete problems have been discovered. Typical examples can be found in such references as [1,5,7,8,11,14].

On the other hand, Valiant [16] proposed a new theory of the holographic reduction that allows for gadgets with many-to-many correspondence, in which the individual correspondences among the solution fragments can no longer be identified. Using this reduction, he succeeded in designing polynomial time algorithms for a number of counting problems.

However the whole class #P is less well understood, perhaps due to the complexity of combinatorial structures of solutions to the problems. An interesting topic in this area is to determine the complexity of the counting problems with various restrictions. Many planar counting problems, such as #Planar 3SAT and #Planar Clique Cover etc., are

* Corresponding author at: State Key Lab. of Computer Science, Institute of Software, Chinese Academy of Sciences, P.O. Box 8717, Beijing 100080, China. Tel.: +86 10 82625471.
E-mail addresses: xmjljx@gmail.com (M. Xia), algzhang@gmail.com (P. Zhang), wbzhao1981@gmail.com (W. Zhao).

0304-3975/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2007.05.023

studied in [8]. Counting problems on planar bipartite graphs with bounded degree or regular graphs are investigated in [14]. In this paper, we consider counting problems on 3-regular planar graphs, that is, the counting version of Planar Read-twice 3SAT. Here, read-twice means that each variable occurs twice.

Vadhan [14] showed by polynomial interpolation that the problem counting vertex covers is #P-complete on planar bipartite graphs with maximum degree 4, and on k-regular graphs for all $k \geq 5$. Greenhill [7] improved the latter to $k = 3$. We show that #3-Regular Bipartite Planar Vertex Covers (#3RBP-VC) is #P-complete (Theorem 9).

Vadhan [14] also showed that the counting of matchings is #P-complete for planar bipartite graphs with maximum degree 6. We use #2,3-Regular Bipartite Planar λ-Matchings (#2,3RBP-λ-Matchings) to denote the problem of calculating the summation of weights of matchings for 2,3-regular bipartite planar graphs, where the weight of a matching is λ to the power t, and t is the number of unmatched vertices of degree 3. Valiant [16] gave a holographic algorithm for the problem #X-Matchings. Therefore, as a special case of #X-Matchings, #2,3RBP-(-3)-Matchings is also polynomial time computable. The authors are very grateful to Valiant for suggesting the problem whether #2,3RBP-λ-Matchings is #P-complete for all rational numbers λ except for 0 and -3 to us. We show that #2,3RBP-λ-Matchings is #P-complete for almost all rational numbers λ except for finitely many ones (Theorem 10).

We also consider the #Planar Read-twice 3SAT problems, where each clause is a ternary symmetric relation in three literals. Different relations may lead to different computational complexity of the corresponding problems. We establish the #P-completeness for three of them, while together with the known cases, we characterize the computational complexity for this class of problem. For example, we show that #Planar Read-twice Monotone 3CNF is #P-complete, improving a result in [8]. A special interesting case is the problem #Planar Read-twice Not-equal-2 3SAT, which is #P-complete, but we prove that a closely related problem, #Read-twice Positive Not-equal-2 3SAT, is polynomial time computable.

Our methodology contribution is a generalization of the polynomial interpolation technique to high dimension, which leads to our first result about #VC. All other #P-completeness results are proved by combining polynomial interpolation and Valiant's holographic reduction.

In Section 2, we introduce the problems, the holographic reduction, the polynomial interpolation theorem in high dimension and some basic facts. In Section 3, we prove that #3RBP-VC is #P-complete. In Section 4, we prove that #2,3RBP-λ-Matchings is #P-complete for almost all rational numbers λ except for finitely many ones. In Section 5, we build the results about #Read-twice 3SAT.

Our terminology and notations are standard; readers are referred to Papadimitriou [12]. This paper is a full version of [18], extended by similar results.

2. Preliminary

In this section, we introduce the problems, notations, basic facts and the main tools.

The beginning of the name of a problem indicates the restrictions on the input graph or formula. For example, #3-Regular Bipartite Planar Vertex Covers (#3RBP-VC) means the #Vertex Covers problem restricted to 3-regular bipartite planar graphs.

Suppose that $G(V, E)$ is a bipartite graph with bipartition V_1 and V_2. If all vertices in V_1 have degree 2, and all vertices in V_2 have degree 3, G is called a 2,3-regular bipartite graph. Edges in E are viewed as Boolean variables, and each vertex $v \in V$ is viewed as a function f_v in 2 or 3 variables corresponding to the edges incident to v. Graph G can be viewed as a function $\mathcal{F} = \prod_{v \in V} f_v$ whose input is truth assignment π for E. If we require $\{f_v | v \in V_1\} \subseteq A$ and $\{f_v | v \in V_2\} \subseteq B$, the problem of computing $\sum_{\pi \in \{0,1\}^{|E|}} \mathcal{F}(\pi)$ is denoted by #A-B-SAT. If G is planar, the problem is denoted by #Pl-A-B-SAT. For every vertex in V_1, merge its two incident edges. If the resulting graph is still bipartite, then the problem is denoted by #Pl-A-B-Bip-SAT.

We use the following notations to denote some binary function set A used in this paper. Rtw$\triangleq \{=_2, \neq_2\}$, Pos$\triangleq \{=_2\}$ and OR$\triangleq \{\vee\}$, where $=_k$ (resp. $\neq_k$) denote equivalence (resp. inequivalence) relation on k variables.

Obviously, #3RBP-VC is just the problem #Pl-OR-$\{=_3\}$-Bip-SAT. If an edge variable takes value 0 (resp. 1) when it is (resp. not) taken in matching, then #2,3RBP-λ-Matchings is the problem #Pl-OR-$\{F_\lambda\}$-SAT, where F_λ denotes the function that has value λ on $(1, 1, 1)$, 1 on $(0, 1, 1)$, $(1, 0, 1)$ and $(1, 1, 0)$, 0 on the other inputs. In Section 5, we consider the computational complexity of #Pl-Pos-$\{f\}$-SAT with different ternary symmetric relations f.

Let $A = (a_{ij})$ and B is a matrix or vector. The tensor product of A and B is

$$A \bigotimes B = \begin{pmatrix} a_{11}B & a_{12}B & \dots \\ a_{21}B & a_{22}B & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}.$$

Denote by $A^{\bigotimes k}$ the product of A tensored with itself k times.

The main tool of our reduction is Valiant's holographic reduction. We briefly outline the framework and the main theorem of this theory.

Consider the problem #$\{g\}$-$\{f\}$-SAT. We call f the generator, and g the recognizer. The standard signatures of f and g, written by Sig(f) and Sig(g) respectively, are just their truth tables in column vector form. Given a basis $[n, p] = [\begin{pmatrix} n_0 \\ n_1 \end{pmatrix}, \begin{pmatrix} p_0 \\ p_1 \end{pmatrix}]$, let $Q = \begin{pmatrix} n_0 & p_0 \\ n_1 & p_1 \end{pmatrix}$. The ValG of generator f under this basis is defined by

$$Q^{\otimes 3}\mathrm{ValG}(f) = \mathrm{Sig}(f)\,. \tag{1}$$

The ValR of recognizer g under this basis is a row vector defined by

$$\mathrm{ValR}(g)(Q^{-1})^{\otimes 2} = \mathrm{Sig}(g)^{\mathrm{T}}\,. \tag{2}$$

Consider ValG(f) as the truth table of another ternary function f', and ValR(g) as the truth table of another ternary function g'. By the Holant Theorem #$\{g\}$-$\{f\}$-SAT and #$\{g'\}$-$\{f'\}$-SAT are the same problem.

Theorem 1 (*Valiant's Holant Theorem [16]*). *For any matchgrid Ω over any basis $\mathbf{b} = [n, p]$, if Ω has weighted graph G then*

$$\mathrm{Holant}(\Omega) = \mathrm{PerfMatch}(G).$$

In this paper we use the holographic reduction only in the restricted case for the 2,3-regular bipartite graphs. More details about the theory of the holographic reduction can be found in [16,17,2,3].

Polynomial interpolation is a technique for constructing Turing reductions, which is best illustrated by an example, reducing #Perfect Matchings to #Matchings.

Given an oracle which counts all matchings in a graph, we want to use this oracle to count the number of perfect matchings in a graph G. Suppose that G contains n vertices. Consider the graph G_i obtained by adding disjoint chains of length i to each vertex of G. Let x_i (resp. y_i) denote the number of matchings of a chain of length i, when one endpoint is (resp. not) deleted. Given a matching of G which matches m vertices of G, there are $x_i^m y_i^{n-m}$ ways to extend it to a matching of G_i. If c_m denotes the number of matchings of G which exactly match m vertices of G, then G_i has $\Sigma_{m=0}^{n} c_m x_i^m y_i^{n-m}$ matchings. By definition, c_n is just the number of perfect matchings of G. We only need to solve a system of linear equations, whose coefficients matrix is a Vandermonde matrix in x_i/y_i. So, we can get c_n, if there are sufficiently many x_i/y_i with different values.

Usually, the G_i are constructed from G by some simple rule such that the G_i are polynomial time computable. As a consequence, the (x_i, y_i) usually satisfy some linear relation such as $(x_{i+1}, y_{i+1})^{\mathrm{T}} = \mathbf{B}(x_i, y_i)^{\mathrm{T}}$, where $\mathbf{B}$ is a 2×2 matrix. Vadhan gave the following two lemmas for deciding whether or not x_i/y_i repeats.

Lemma 2 (*[14]*). *Let a, b, c, d be rational numbers and α, β be nonzero complex numbers. Let the sequence z_i be defined by*

$$z_i = \frac{a\alpha^i + b\beta^i}{c\alpha^i + d\beta^i}.$$

Then the sequence $\{z_i\}$ repeats iff $ad - bc = 0$ or α/β is a root of unity.

Lemma 3 (*[14]*). *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:*

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

Lemma 3 refines Lemma 2 by a detailed analysis of the second condition, i.e. whether α/β is a root of unity. By Lemma 3, x_i/y_i never repeats in the above example.

Sometimes to preserve some desired properties of G, we construct G_i from G by adding some sub-structures affecting two or more vertices. Lemmas 2 and 3 are not applicable to this situation. So, we generalize Lemma 2 to high dimension for some more exquisite and sophisticated applications of this technique.

Denote by $\mathcal{I}_n$ the set $\{(a, b, c) | a + b + c = n, a, b, c \in \mathbb{N}\}$ (throughout this paper, $\mathbb{N}$ contains 0), and let

$$F_n(x, y, z) = \sum_{(a,b,c)\in\mathcal{I}_n} c_{(a,b,c)} x^a y^b z^c$$

be a homogeneous polynomial, which contains $\binom{n+2}{2}$ monomials of degree n. $(x_i, y_i, z_i)^{\mathrm{T}}$, $i = 0, 1, \ldots, \binom{n+2}{2} - 1$ is a recurrence sequence of vectors satisfying

$$(x_{i+1}, y_{i+1}, z_{i+1})^{\mathrm{T}} = \mathbf{B}(x_i, y_i, z_i)^{\mathrm{T}},$$

where $\mathbf{B}$ is a 3×3 matrix. Given a sequence $(x_i, y_i, z_i)^{\mathrm{T}}$ and its corresponding values $F_n(x_i, y_i, z_i), i = 0, 1, \ldots, \binom{n+2}{2} - 1$, we get $\binom{n+2}{2}$ equations for $c_{(a,b,c)}$, $(a, b, c) \in \mathcal{I}_n$. We denote the coefficient matrix of this system of linear equations by $\mathbf{M}_n$.

Suppose

$$\mathbf{B} = \mathbf{E}\begin{pmatrix} \alpha & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & \gamma \end{pmatrix}\mathbf{E}^{-1},$$

where α, β and γ are three different eigenvalues of $\mathbf{B}$ and $\mathbf{E}$ is a 3×3 nonsingular matrix. Let $\mathbf{A}$ be the matrix satisfying the following constraint:

$$\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} = \mathbf{E}\begin{pmatrix} \alpha^i & 0 & 0 \\ 0 & \beta^i & 0 \\ 0 & 0 & \gamma^i \end{pmatrix}\mathbf{E}^{-1}\begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} = \mathbf{A}\begin{pmatrix} \alpha^i \\ \beta^i \\ \gamma^i \end{pmatrix}.$$

The following theorem gives a sufficient condition which guarantees that $|\mathbf{M}_n| \neq 0$, that is, $c_{(a,b,c)}$, the coefficients of F_n, can be recovered by solving the system of the linear equations.

Theorem 4. *If the following two conditions hold:*

(1) $|\mathbf{A}| \neq 0$*;*
(2) *for any* $(l, m, k) \in \mathbb{N}$, $l + m + k = 0$ *and* $(l, m, k) \neq (0, 0, 0)$, $\alpha^l \beta^m \gamma^k \neq 1$,

then, for any n, $|\mathbf{M}_n| \neq 0$.

Proof. Let a_{ij} and m_{ij} denote the (i, j)th entry of $\mathbf{A}$ and $\mathbf{M}_n$ respectively. And take elements of $\mathcal{I}_n$ as numbers between 1 and $\binom{n+2}{2}$. Then the $(i, (a, b, c))$th entry of matrix $\mathbf{M}_n$ is

$$\begin{aligned} m_{i,(a,b,c)} &= x_{i-1}^a y_{i-1}^b z_{i-1}^c \\ &= (a_{11}\alpha^{i-1} + a_{12}\beta^{i-1} + a_{13}\gamma^{i-1})^a \\ &\quad (a_{21}\alpha^{i-1} + a_{22}\beta^{i-1} + a_{23}\gamma^{i-1})^b \\ &\quad (a_{31}\alpha^{i-1} + a_{32}\beta^{i-1} + a_{33}\gamma^{i-1})^c. \end{aligned}$$

Expanding the last term of the above equation, we get

$$\sum_{(l,m,k)\in\mathcal{I}_n} t_{(l,m,k),(a,b,c)}(\alpha^l\beta^m\gamma^k)^{i-1},$$

where $t_{(l,m,k),(a,b,c)}$ only depends on (l, m, k), (a, b, c) and $\mathbf{A}$.

Define $\mathbf{T}$ to be the matrix whose $((l, m, k), (a, b, c))$th entry is $t_{(l,m,k),(a,b,c)}$ and $\mathbf{N}$ to be the matrix whose $(i, (l, m, k))$th entry is $(\alpha^l\beta^m\gamma^k)^{i-1}$, $i = 1, 2, \ldots, \binom{n+2}{2}$. By definition of matrix multiplication, we have $\mathbf{M}_n = \mathbf{NT}$. Hence, we only need to prove that $|\mathbf{N}| \neq 0$ and that $|\mathbf{T}| \neq 0$. Notice that $\mathbf{N}$ is a Vandermonde matrix in $\alpha^l\beta^m\gamma^k$, $(l, m, k) \in \mathcal{I}_n$. By assumption (2), all of them are different; thus $|\mathbf{N}| \neq 0$.

The (a, b, c)th column of $\mathbf{T}$ (denoted by $\vec{T}_{(a,b,c)}$) is nothing but the coefficient vector of polynomial f_{abc} in variables α, β and γ. Here,

$$\begin{aligned} f_{abc} \triangleq x^a y^b z^c &= (a_{11}\alpha + a_{12}\beta + a_{13}\gamma)^a \\ &\quad (a_{21}\alpha + a_{22}\beta + a_{23}\gamma)^b \\ &\quad (a_{31}\alpha + a_{32}\beta + a_{33}\gamma)^c. \end{aligned} \tag{3}$$

To prove $|\mathbf{T}| \neq 0$, we only need to prove that all $\vec{T}_{(a,b,c)}$ are linearly independent. Suppose

$$\sum_{(a,b,c)\in\mathcal{I}_n} h_{abc}\vec{T}_{(a,b,c)} = \vec{0}.$$

We prove that all h_{abc} are equal to zero. Since $\sum_{(a,b,c)\in\mathcal{I}_n} h_{abc}\vec{T}_{(a,b,c)}$ is the coefficient vector of polynomial $\sum_{(a,b,c)\in\mathcal{I}_n} h_{abc}f_{abc}$ (as a polynomial in α, β and γ), $\sum_{(a,b,c)\in\mathcal{I}_n} h_{abc}f_{abc}(\alpha, \beta, \gamma) = 0$, for any α, β and γ. Because $|\mathbf{A}| \neq 0$, for any x, y and z, there exists a vector $(\alpha, \beta, \gamma)^{\mathrm{T}}$, such that $(x, y, z)^{\mathrm{T}} = \mathbf{A}(\alpha, \beta, \gamma)^{\mathrm{T}}$. By Eq. (3)

$$\sum_{(a,b,c)\in\mathcal{I}_n} h_{abc}x^a y^b z^c = \sum_{(a,b,c)\in\mathcal{I}_n} h_{abc}f_{abc}(\alpha, \beta, \gamma) = 0.$$

This means that $\sum_{(a,b,c)\in\mathcal{I}_n} h_{abc}x^a y^b z^c \equiv 0$. So, all h_{abc} are zero and $\vec{T}_{(a,b,c)}$, $(a, b, c) \in \mathcal{I}_n$ are linearly independent, which implies that $|\mathbf{T}| \neq 0$.

$|\mathbf{M}_n| = |\mathbf{T}||\mathbf{N}| \neq 0$ holds. □

Theorem 4 is a natural extension of Lemma 2, which requires a completely different proof. We note that although Theorem 4 is proved in the case of three variables, it also holds for polynomials in any number of variables.

The following lemma is stated in the form of two polynomials, each in two variables. In fact, it holds for finitely many polynomials.

Lemma 5. *For any $p, q \in \mathbb{N}$ and any polynomial f, g and h, where*

$$f = \sum_{\substack{i+j=p,\\ i,j\in\mathbb{N}}} a_{ij}x^i y^j,\ g = \sum_{\substack{s+t=q,\\ s,t\in\mathbb{N}}} b_{st}z^s w^t \text{ and } h = \sum_{\substack{i+j=p,s+t=q,\\ i,j,s,t\in\mathbb{N}}} c_{ijst}x^i y^j z^s w^t.$$

If all the coefficients of both polynomials f and g can be recovered from their values on (x_k, y_k), $k = 1, 2, \ldots, m$ and on (z_u, w_u), $u = 1, 2, \ldots, n$ respectively, then the coefficients of h can be recovered from its values on (x_k, y_k, z_u, w_u), $k = 1, 2, \ldots, m$, $u = 1, 2, \ldots, n$.

Proof. Because the coefficient matrix of the linear equation system for h's, c_{ijst}, which is a matrix determined by (x_k, y_k, z_u, w_u), $k = 1, 2, \ldots, m$, $u = 1, 2, \ldots, n$, is just the tensor product of the coefficient matrixes of systems for f's, a_{ij}, and g's, b_{st}, which are matrices determined by (x_k, y_k), $k = 1, 2, \ldots, m$ and (z_u, w_u), $u = 1, 2, \ldots, n$ respectively. □

Sometimes, we need to combine several Turing reductions into one reduction. If all of the reductions are from polynomial interpolation, we can directly build one reduction by Lemma 5. The following lemma states that the main step needed in reduction is polynomial time computable.

Lemma 6. *Suppose that a system of linear equations has m rational coefficient equations in n variables, and has a unique solution. Then the solution can be computed in polynomial time, if m, n and the length of coefficients and constant items are bounded by a polynomial in n.*

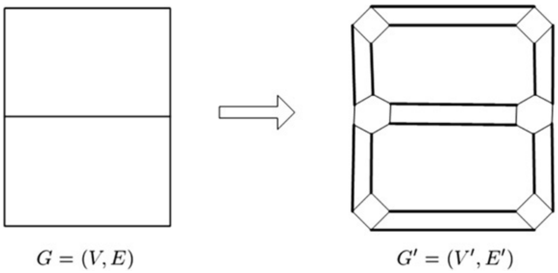

Fig. 1. In graph G', set E_c specifies the set of all thick edges, and set E_e specifies all the light edges.

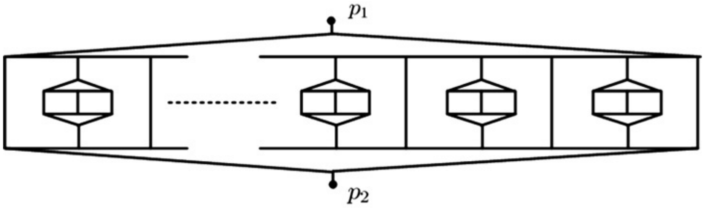

Fig. 2. An s-block containing s devices $\mathcal{G}$.

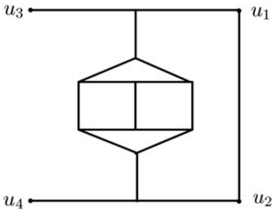

Fig. 3. Figure illustrating a device $\mathcal{G}$.

Proof. By a standard algorithm for finding the solution of a linear equation system. □

3. Vertex covers

In this section, we show that #3RBP-VC is #P-complete. We prove the result by reducing the #P-complete problem #Planar Bipartite VC [14] to it. The reduction here is a polynomial time Turing (precisely, truth table) reduction.

3.1. The reduction

Let $G(V, E)$ be a planar bipartite graph and $|E| = m$. Consider the graph $G'(V', E')$ derived from G by replacing each vertex $v \in V$ in G with a cycle C_v of $2d(v)$ vertices and connecting two adjacent vertices of C_v to the two adjacent vertices of cycle C_u respectively, where $d(v)$ is the degree of v, $u \in V$ and $(u, v) \in E$ (Fig. 1). Let set $E_c \subseteq E'$ specify the set of edges in cycles, and $E_e \subseteq E'$ specify the set of all the edges between cycles. Clearly, G' is a 3-regular planar bipartite graph, and $E' = E_c \cup E_e$, and $|E_c| = 2|E_e| = 4|E| = 4m$.

In Fig. 2, the gadget containing s devices $\mathcal{G}$ is called an s-block, and the vertices p_1 and p_2 are endpoints of the s-block. The device $\mathcal{G}$ is shown in Fig. 3.

Now we define graph $G_{s,t}$ to be the graph induced from graph $G' = (V', E')$ by replacing each edge $e_c \in E_c$ by an s-block and replacing each edge $e_e \in E_e$ by a t-block. Obviously, $G_{s,t}$ is also a 3-regular planar bipartite graph. Let

$$
\begin{aligned}
k_{(a,b,c),(a',b',c')} = |\{X \subseteq V'|\ & a \text{ edges of } E_c \text{ have two endpoints in } X, \\
& b \text{ edges of } E_c \text{ have exactly one endpoint in } X, \\
& c \text{ edges of } E_c \text{ have no endpoint in } X\text{; and similarly,} \\
& a' \text{ edges of } E_e \text{ have two endpoints in } X,
\end{aligned}
$$

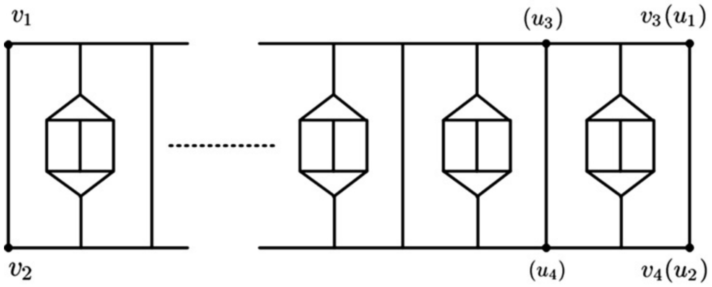

Fig. 4. A part of an s-block, $\mathcal{B}_s$.

$$b' \text{ edges of } E_e \text{ have exactly one endpoint in } X,$$
$$c' \text{ edges of } E_e \text{ have no endpoint in } X\}|,$$

then the number of vertex covers of $G_{s,t}$ is

$$\#\mathrm{VC}(G_{s,t}) = \sum_{\substack{(a,b,c)\in\mathcal{I}_{2m},\\(a',b',c')\in\mathcal{I}_m}} k_{(a,b,c),(a',b',c')} z_{s,11}^{a} z_{s,10}^{b} z_{s,00}^{c} z_{t,11}^{a'} z_{t,10}^{b'} z_{t,00}^{c'},$$

where $z_{s,ij}$ (resp. $z_{t,ij}$) denotes the number of vertex covers of the s-block (resp. t-block) such that the endpoints p_1 and p_2 are taken or not according to the values of i and j, that is,

$$z_{s,ij} = |\{X \text{ is a vertex cover of the } s\text{-block} \mid \chi_X(p_1) = i, \text{ and } \chi_X(p_2) = j\}|.$$

Here, $\chi_X(\cdot)$ is the characteristic function of set X.

By Lemma 8 in the next part, all the $z_{s,ij}, z_{t,ij}$ can be computed in polynomial time, and $(z_{s,00}, z_{s,10}, z_{s,11})^{\mathrm{T}}$ satisfies the conditions in Theorem 4. $\#\mathrm{VC}(G_{s,t})$, $s, t = 1, 2, \ldots, \binom{n+3}{3}$ can also be obtained in polynomial time by asking the oracle. Moreover, all of these numbers above are of polynomial length. By Theorem 4, Lemmas 5 and 6, all $t_{(a,b,c),(a',b',c')}$ can be computed in polynomial time.

The key observation to the proof is that

$$\#\mathrm{VC}(G) = \sum_{\substack{c(a,b,c)\in\mathcal{I}_{2m}, b=0\\(a',b',c')\in\mathcal{I}_m, c'=0}} k_{(a,0,c),(a',b',0)}. \tag{4}$$

Because $b = 0$ and $c' = 0$ make sure that the summation above is the number of all such subsets $X \subseteq V'$, which satisfy that all or none vertices of a circle C_v, $v \in V$ are contained in the subset X, and every two parallel edges between two circles are covered simultaneously by X from one side or both sides. Hence, all subsets X counted here can be 1-1 and onto mapped to vertex covers of G, and Eq. (4) follows.

3.2. The calculation

In this subsection, we establish the technique lemma used in the reduction, which ensures that the construction satisfies the conditions of Theorem 4. So, we need to compute the recurrence sequence of the s-block. We shall first compute the recurrence sequence of its main part.

A part of the s-block named $\mathcal{B}_s$ is shown in Fig. 4, where v_1, v_2, v_3, v_4 are four vertices in the four corners of $\mathcal{B}_s$. Let $x_{s,i_1i_2i_3i_4}$ specify the number of vertex covers of $\mathcal{B}_s$ such that the four vertices v_1, v_2, v_3, v_4 are taken or not according to the values of i_1, i_2, i_3, i_4, that is,

$$x_{s,i_1i_2i_3i_4} = |\{X \text{ is vertex cover of } \mathcal{B}_s| \quad \chi_X(v_1) = i_1, \chi_X(v_2) = i_2,$$
$$\chi_X(v_3) = i_3, \chi_X(v_4) = i_4\}|.$$

and the vector $\vec{x}_s = (x_{s,i_1i_2i_3i_4})_{16\times 1} = (x_{s,0000}\ x_{s,0001} \ldots x_{s,1111})^{\mathrm{T}}$.

To prove the main lemma, Lemma 8, we need a recursive characterization of the vector sequence $\vec{x}_s$.

Lemma 7. $\vec{x}_{s+1} = \mathbf{A}_{\mathcal{B}}\vec{x}_s$, $s = 0, 1, 2, \ldots$, *where*

$$\mathbf{A}_{\mathcal{B}} = \begin{pmatrix} \mathbf{A}_{\mathcal{G}} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{\mathcal{G}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_{\mathcal{G}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{\mathcal{G}} \end{pmatrix}, \mathbf{A}_{\mathcal{G}} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 38 & 64 & 38 & 64 \\ 38 & 38 & 64 & 64 \\ 38 & 64 & 64 & 107 \end{pmatrix} \textit{ and}$$

$$\vec{x}_0 = (0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1)^{\mathrm{T}}.$$

Proof. First define $x_{\mathcal{G},i_1i_2i_3i_4}$ as follows.

$$x_{\mathcal{G},i_1i_2i_3i_4} = |\{X \text{ is vertex cover of } \mathcal{G} |\ \chi_X(u_1) = i_1, \chi_X(u_2) = i_2, \chi_X(u_3) = i_3, \chi_X(u_4) = i_4\}|.$$

It is easy to verify the following:

$$\begin{pmatrix} x_{\mathcal{G},0000} & x_{\mathcal{G},0001} & x_{\mathcal{G},0010} & x_{\mathcal{G},0011} \\ x_{\mathcal{G},0100} & x_{\mathcal{G},0101} & x_{\mathcal{G},0110} & x_{\mathcal{G},0111} \\ x_{\mathcal{G},1000} & x_{\mathcal{G},1001} & x_{\mathcal{G},1010} & x_{\mathcal{G},1011} \\ x_{\mathcal{G},1100} & x_{\mathcal{G},1101} & x_{\mathcal{G},1110} & x_{\mathcal{G},1111} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 38 & 64 & 38 & 64 \\ 38 & 38 & 64 & 64 \\ 38 & 64 & 64 & 107 \end{pmatrix}.$$

Notice that $\mathcal{B}_{s+1}$ is just $\mathcal{B}_s$ connected by one more $\mathcal{G}$. So, we can compute $\vec{x}_{s+1}$ from $\vec{x}_s$ and $\mathbf{A}_{\mathcal{G}}$. Each item of $\vec{x}_{s+1}$ is the summation of the number of vertex covers of four cases corresponding to whether or not we are taking u_3 and u_4. So,

$$x_{s+1,i_1i_2i_3i_4} = x_{s,i_1i_200}x_{\mathcal{G},i_3i_400} + x_{s,i_1i_201}x_{\mathcal{G},i_3i_401} + x_{s,i_1i_210}x_{\mathcal{G},i_3i_410} + x_{s,i_1i_211}x_{\mathcal{G},i_3i_411}.$$

We have $\vec{x}_{s+1} = \mathbf{A}_{\mathcal{B}}\vec{x}_s$ by the definition of matrix multiplication.

Notice that $\mathcal{B}_0$ is just one edge, whose top vertex is labelled by v_1 and v_3 and bottom vertex is labelled by v_2 and v_4 simultaneously. So,

$$\vec{x}_0 = (0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1)^{\mathrm{T}}.$$

The lemma follows. □

Here, because there are two connecting points when connecting $\mathcal{G}$ to $\mathcal{B}_s$ in the construction of $\mathcal{B}_{s+1}$, $\mathbf{A}_{\mathcal{B}}$ is a block matrix with block size 2^2. Usually, k connecting points will lead to a block matrix with block size 2^k. So, $\mathcal{B}_s$ should satisfy that there is no way to cut it into a sequence of identical parts such that each pair of adjacent parts only share one vertex. Otherwise, $\mathbf{A}_{\mathcal{B}}$ has only two different eigenvalues, which are not sufficient to recover a polynomial in more than two variables.

Now we are ready to prove the main lemma. Denote vector $(z_{s,00}, z_{s,01}, z_{s,11})^{\mathrm{T}}$ by $\vec{z}_s$.

Lemma 8. $\vec{z}_s = \mathbf{A}(\lambda_1^s, \lambda_2^s, \lambda_3^s)^{\mathrm{T}}$ *for any* $s = 0, 1, 2, \ldots$, *and the rank of* $\mathbf{A}$ *is* 3, *and* $\lambda_1, \lambda_2, \lambda_3$ *satisfy the condition* 2 *of Theorem* 4.

Proof. The four eigenvalues of matrix $\mathbf{A}_{\mathcal{G}}$ are

$$\lambda_1 = 26, \lambda_2 = \frac{209 + \sqrt{32\,793}}{2}, \lambda_3 = \frac{209 - \sqrt{32\,793}}{2} \text{ and } \lambda_4 = 0.$$

Because $\lambda_1^2 \neq \lambda_2\lambda_3$, $\lambda_1, \lambda_2, \lambda_3$ satisfy condition 2 of Theorem 4.

Notice that $\mathbf{A}_{\mathcal{G}}$ can be decomposed into the following form:

$$\mathbf{A}_{\mathcal{G}} = \mathbf{E}\Lambda\mathbf{E}^{-1}, \text{ where } \Lambda = \begin{pmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \lambda_3 & 0 \\ 0 & 0 & 0 & \lambda_4 \end{pmatrix}.$$

By Lemma 7,

$$\vec{x}_s = \begin{pmatrix} \vec{0}_{4\times 1} \\ \mathbf{E}\Lambda^s\mathbf{E}^{-1}\cdot(0,1,0,0)^{\mathrm{T}} \\ \mathbf{E}\Lambda^s\mathbf{E}^{-1}\cdot(0,0,1,0)^{\mathrm{T}} \\ \mathbf{E}\Lambda^s\mathbf{E}^{-1}\cdot(0,0,0,1)^{\mathrm{T}} \end{pmatrix} = \begin{pmatrix} & \mathbf{0}_{5\times 3} & \\ \frac{1}{2} & C & D \\ -\frac{1}{2} & C & D \\ 0 & A & -A \\ 0 & 0 & 0 \\ -\frac{1}{2} & C & D \\ \frac{1}{2} & C & D \\ 0 & A & -A \\ 0 & 0 & 0 \\ 0 & A & -A \\ 0 & A & -A \\ 0 & E & F \end{pmatrix} \begin{pmatrix} \lambda_1^s \\ \lambda_2^s \\ \lambda_3^s \end{pmatrix},$$

where

$$\begin{aligned} A &= \frac{64}{\sqrt{32\,793}},\ C = \frac{32\,793 - 5\sqrt{32\,793}}{131\,172},\ D = \frac{32\,793 + 5\sqrt{32\,793}}{131\,172}, \\ E &= \frac{5+\sqrt{32\,793}}{2\sqrt{32\,793}} \text{ and } F = \frac{-5+\sqrt{32\,793}}{2\sqrt{32\,793}}. \end{aligned}$$

Because an s-block is just a $\mathcal{B}_s$ with several more edges, it is easy to see that

$$\begin{aligned} \vec{z}_s &= \begin{pmatrix} 0&0&0&0&0&1&1&1&0&1&1&1&0&1&1&1 \\ 0&0&0&0&0&1&1&1&0&1&2&2&0&1&2&2 \\ 0&0&0&0&0&2&1&2&0&1&2&2&0&2&2&4 \end{pmatrix} \vec{x}_s \\ &= \begin{pmatrix} 0 & 4A+4C+E & -4A+4D+F \\ \frac{1}{2} & 6A+5C+2E & -6A+5D+2F \\ 1 & 8A+6C+4E & -8A+6D+4F \end{pmatrix} \begin{pmatrix} \lambda_1^s \\ \lambda_2^s \\ \lambda_3^s \end{pmatrix}. \end{aligned} \tag{5}$$

It is straightforward that the rank of the matrix in Eq. (5) is 3.

Thus, we complete the proof of this lemma. □

Now we have:

Theorem 9. *#3RBP-VC is #P-complete.* □

4. Matchings

In this section, we give a reduction from #3RBP-VC to #2,3RBP- λ-Matchings for all rational numbers λ except for finitely many ones.

The idea is that we first realize a ternary equivalence relation under some basis, and compute the binary relation which is binary OR under this basis. So, we only need to realize such a binary relation. We use low-dimension polynomial interpolation to show that we can use an arbitrary finite unary relation in the construction for such a binary relation.

Now, we simulate the equivalence relation using matchings by holographic reduction. Unfortunately, if $\lambda \neq 0$, there is no basis such that the ValG(F_λ) is the truth table of the equivalence relation. So, we use the gadget shown in Fig. 5(a) as the generator.

The functions at solid nodes should be F_λ, but now we just consider F_1. The functions at white nodes are OR relation. It is easy to verify that the whole generator is a function with standard signature $(1,3,3,8,3,8,8,18)^{\mathrm{T}}$. By solving the equation $(1,3,3,8,3,8,8,18)^{\mathrm{T}} = Q^{\otimes 3}(1,0,0,0,0,0,0,1)^{\mathrm{T}}$, which is

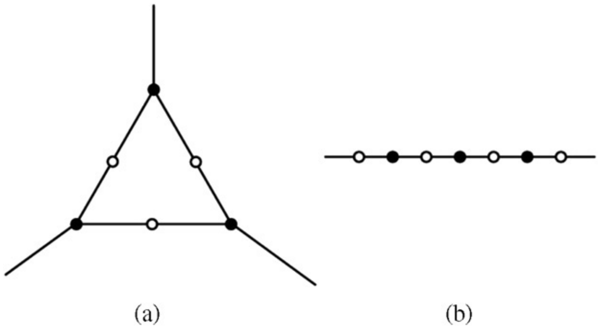

Fig. 5. Simulating equivalence relation and OR relation.

$$\begin{cases} p_0^3 + n_0^3 &= 1 \\ p_0^2 p_1 + n_0^2 n_1 &= 3 \\ p_0 p_1^2 + n_0 n_1^2 &= 8 \\ p_1^3 + n_1^3 &= 18, \end{cases}$$

we get

$$[n, p] = \left[\begin{pmatrix} \frac{1}{2}\sqrt[3]{4} \\ \frac{1}{2}\sqrt[3]{4}(3+\mathrm{i}) \end{pmatrix}, \begin{pmatrix} \frac{1}{2}\sqrt[3]{4} \\ \frac{1}{2}\sqrt[3]{4}(3-\mathrm{i}) \end{pmatrix} \right].$$

So, we can regard this gadget as a ternary equivalence relation generator under this basis.

By the definition of ValR, Eq. (2), the recognizer with standard signature $\frac{1}{2\sqrt[3]{2}}(-4+6\mathrm{i}, 3-\mathrm{i}, 3-\mathrm{i}, -1)^{\mathrm{T}}$ has ValR $(0, 1, 1, 1)^{\mathrm{T}}$ under this basis, that is, it can be regarded as an OR relation. We use the gadget shown in Fig. 5 (b) to realize it. As mentioned above, we have only considered the case that functions on the solid nodes of generator are F_1. If they are F_λ instead of F_1, to cancel this effect, we make the recognizer and the functions on the white nodes of generator supply more factor λ depending on the number of inputs that are zero. That is, we need a recognizer with standard signature $\frac{1}{2\sqrt[3]{2}}((-4+6\mathrm{i})\lambda^2, (3-\mathrm{i})\lambda, (3-\mathrm{i})\lambda, -1)^{\mathrm{T}}$, and a modified OR relation $(0, \lambda, \lambda, 1)$ for the white nodes of the generator.

Before realizing them, we explain why they can cancel the effect brought out by changing from F_1 to F_λ. This happens before the holographic reduction. Given an instance $G = (V, E)$ of #3RBP-VC, we replace the vertices and edges of G by generators and recognizers respectively to get G'. Since the value of G' as a function is the product of values of each small function at the vertices, we can regard the factors λ in the value of the recognizer and the function at the white nodes of the generator as coming from F_λ functions. In so doing, F_λ becomes a function with value λ on (1, 1, 1), (0, 1, 1), (1, 0, 1) and (1, 1, 0), and 0 on the other inputs, which has the same effect as F_1 after removing a global factor λ to the power $|V|$, and the recognizers turn back to the function $\frac{1}{2\sqrt[3]{2}}(-4+6\mathrm{i}, 3-\mathrm{i}, 3-\mathrm{i}, -1)^{\mathrm{T}}$. By the Holant Theorem, the number of vertex covers of G is just the value of the #3OR-$\{F_\lambda\}$-SAT problem on input G'.

In the recognizer, the functions at white nodes are OR relations, the function at the middle solid node is $F_{z,w}$, and the functions at the other solid nodes are $F_{1,y}$, where $F_{z,w}$ denotes that the function has value z on input (1, 1), w on inputs (1, 0) and (0, 1), and value 0 on input (0, 0). This recognizer has standard signature $(2wy + 2w + zy^2 + 2zy + z, wy^2 + 4wy + 2w + 2zy^2 + 3zy + z, wy^2 + 4wy + 2w + 2zy^2 + 3zy + z, 4wy^2 + 6wy + 2w + 4zy^2 + 4zy + z)^{\mathrm{T}}$. If

$$\begin{cases} w &= -\lambda(4\lambda+1+\mathrm{i})^2/\sqrt[3]{2}(2\lambda+1+\mathrm{i})^2 \\ y &= -(2\lambda+1+\mathrm{i})/(4\lambda+1+\mathrm{i}) \\ z &= (1-\mathrm{i})(4\lambda+1+\mathrm{i})^2((2\mathrm{i}-10)\lambda^2-4\lambda-1-\mathrm{i})/4\sqrt[3]{2}(2\lambda+1+\mathrm{i})^2, \end{cases}$$

then the standard signature is $\frac{1}{2\sqrt[3]{2}}((-4+6\mathrm{i})\lambda^2, (3-\mathrm{i})\lambda, (3-\mathrm{i})\lambda, -1)^{\mathrm{T}}$. If

$$\begin{cases} w &= -\lambda \\ y &= -1 \\ z &= 1, \end{cases}$$

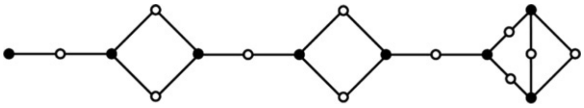

Fig. 6.

then the standard signature is $(0, \lambda, \lambda, 1)$. So, the gadget shown in Fig. 5(b) is also used as the modified OR relation $(0, \lambda, \lambda, 1)$ at the white nodes of the generator.

We can simulate $F_{z,w}$ by polynomial interpolation as was done in the example of reduction from #Perfect Matchings to #Matching in Section 2. But this time, we want $\sum_a c_a z^a w^{n-b}$ instead of $\sum_a c_a 1^a 0^{n-b} = c_n$. Only four kinds of $F_{z,w}$ function are used in the above process. So, by Lemma 5, we can simulate all of them. A more complicated chain is used to do polynomial interpolation, which will be connected to all solid nodes of degree 2 in graph G', making it 2,3-regular bipartite. Fig. 6 shows the chain of length 2. Functions F_λ lie on solid nodes, and OR relations lie on white nodes.

Let x_i (resp. y_i) denotes the value of the chain of length i with the left node (resp. not) matched by the other part of the whole graph. We have

$$\begin{pmatrix} x_{i+1} \\ y_{i+1} \end{pmatrix} = \begin{pmatrix} 2\lambda + 4 & \lambda + 3 \\ 2\lambda^2 + 5\lambda + 2 & \lambda^2 + 4\lambda + 2 \end{pmatrix} \begin{pmatrix} x_i \\ y_i \end{pmatrix}$$

and

$$\begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = \begin{pmatrix} \lambda^3 + 9\lambda^2 + 23\lambda + 15 \\ \lambda^7 + 10\lambda^3 + 31\lambda^2 + 32\lambda + 8 \end{pmatrix}.$$

By Lemma 3, for all but finitely many rational numbers λ, the chain is fit for polynomial interpolation, because no zero polynomial in λ appears to the left side of the inequalities of Lemma 3, and there are only finitely many solutions of λ, which dissatisfy one of the inequalities. We have proved the following theorem.

Theorem 10. *#2,3RBP-λ-Matchings is #P-complete for all rational numbers λ except for finitely many ones.* □

Assume that #P is not equal to FP; because of Valiant's holographic algorithm for #X-Matchings, Theorem 10 does not hold for $\lambda = -3$. In fact, even without this assumption, the holographic algorithm for #X-Matchings shows that the proof does not apply to $\lambda = -3$. Given the arbitrary instance graph H_i of #X-Matchings, one vertex v of H_i will be connected to the original graph. To compute x_i (resp. y_i) of H_i, replace the vertices and edges of H_i by the recognizers and generators in the holographic algorithm for #X-Matchings, and connect a generator just generating a p (resp. n) to the vertex v to get a new graph, where $p = (1, 0)$ and $n = (-1, 1)$ are the basis used in the holographic algorithm for #X-Matchings, and x_i (resp. y_i) is just the perfect matching polynomial of this new graph. It is easy to verify that either $x_i = -y_i$ or $x_i = 0$. This is a property of #X-Matchings, which indicates that we cannot use #X-Matchings for this kind of polynomial interpolation.

5. Ternary symmetric relation

In this section, we consider the complexity of #3Pl-Rtw-$\{f\}$-SAT problems with the ternary symmetric Boolean relation f. Here, ternary symmetric Boolean relations are relations of the form

$$f(a, b, c) = \begin{cases} 1 & \text{if } a + b + c \in S \\ 0 & \text{o.w.,} \end{cases}$$

where $a, b, c \in \{0, 1\}$ and $S \subseteq \{0, 1, 2, 3\}$. We use f_S to denote such a relation. For simplicity, we also use the character vector of S to denote the truth table of f_s and f_s itself. For example, $[1, 0, 1, 1]$ means $(1, 0, 0, 1, 0, 1, 1, 1)$, denoting the relation "$a + b + c$ not equal 1". This notation is also used for symmetric functions in Boolean variables.

The dual relation of a relation f is $1 - f$, that is, the complement set of f, if f is viewed as a set. Obviously, the computational complexities of #Rtw-$\{f\}$-SAT and #Rtw-$\{1 - f\}$-SAT are identical. So, we only consider one of the relations and its dual relation. There are ten such relations: F_Φ, $F_{\{0\}}$, $F_{\{1\}}$, $F_{\{0,1\}}$, $F_{\{0,2\}}$, $F_{\{0,3\}}$, $F_{\{1,2\}}$, $F_{\{0,1,2\}}$, $F_{\{0,1,3\}}$ and $F_{\{0,1,2,3\}}$. Among them, #Rtw-$\{f\}$-SAT problems with f equal to F_Φ, $F_{\{0\}}$, $F_{\{0,2\}}$, $F_{\{0,3\}}$ or $F_{\{0,1,2,3\}}$ are obviously polynomial time computable. #Pl-Rtw-$\{F_{\{1\}}\}$-SAT is just the #Planar Perfect Matchings problem restricted to a 3-regular graph, so it is also polynomial time computable by the FKT algorithm [6,9,10,13]. #Pl-Rtw-$\{F_{\{1,2\}}\}$-SAT is just #Planar 3-NAE SAT, which is also polynomial time computable by a holographic algorithm [16].

The remaining three problems corresponding to $F_{\{0,1\}}$, $F_{\{0,1,2\}}$ and $F_{\{0,1,3\}}$ are left. #Pl-Pos-$\{F_{\{0,1\}}\}$-SAT is just #3-Regular Planar Matchings. #Pl-Pos-$\{F_{\{0,1,2\}}\}$-SAT is just what is called #Planar Read-twice Monotone 3CNF in the introduction section. The last problem is #Pl-Rtw-$F_{\{0,1,3\}}$. We prove that the three problems are #P-complete by using the same method as that in Section 4. It is interesting to notice that the decision version of the three problems without the restriction planarity are in P, given by Cook [4].

First, we show that they are fit for the low-dimension polynomial interpolation as the matching example in Section 2. The gadget in Fig. 6 is a chain of length 2. We do not consider the leftmost solid node. The functions at white nodes are binary equivalence relations, so we can think of them as variables occurring twice positively (except the leftmost white node, named v_1, which only occurs once). Let x_i (resp. y_i) denote the numbers of satisfying truth assignments to the chain of length i, such that $v_1 = 0$ (resp. $v_1 = 1$). It is easy to compute the recurrence sequence (x_i, y_i). If the functions at solid nodes are $F_{\{0,1\}}$,

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} 3 & 1 \\ 1 & 1 \end{pmatrix}^i \begin{pmatrix} 5 \\ 3 \end{pmatrix}.$$

If the functions at solid nodes are $F_{\{0,1,2\}}$,

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} 4 & 3 \\ 3 & 3 \end{pmatrix}^i \begin{pmatrix} 13 \\ 10 \end{pmatrix}.$$

If the functions at solid nodes are $F_{\{0,1,3\}}$,

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} 3 & 1 \\ 1 & 2 \end{pmatrix}^i \begin{pmatrix} 5 \\ 5 \end{pmatrix}.$$

All of them satisfy the conditions in Lemma 3. So, when constructing recognizers, we can use arbitrary finitely many kinds of unary functions.

For $F_{\{0,1\}}$, we use the gadget in Fig. 5(a) as the generator, whose standard signature is $[4, 2, 1, 1]^{\mathrm{T}}$. By Eq. (1), the ValG of it is $[1, 0, 0, 1]^{\mathrm{T}}$ (ternary equivalence relation), under the basis

$$[n, p] = \left[\begin{pmatrix} \sqrt[3]{4} \\ \frac{1}{2}\sqrt[3]{4} \end{pmatrix}, \begin{pmatrix} 0 \\ \frac{1}{2}\sqrt[3]{4} \end{pmatrix} \right].$$

By Eq. (2), the recognizer with standard signature $\frac{1}{2\sqrt[3]{2}}[-1, 0, 4]$ has ValR $[0, 1, 1]$ (OR relation) under this basis. We use the gadget in Fig. 5(b) as the recognizer. The first and the third solid nodes are connected to a unary function $f_{x,y}$ by an edge variable, and the second is connected to a unary function $f_{z,w}$, where $f_{z,w}$ denotes the function that $f_{z,w}(0) = z$, $f_{z,w}(1) = w$. The functions at solid nodes are $F_{\{0,1\}}$, and the functions at white nodes are just binary equivalence relations. The standard signature of this recognizer is $[3x^2z + x^2w + 4xyz + 2xyw + y^2z + y^2w, 2x^2z + x^2w + xyz + xyw, x^2z + x^2w]$, and there exist solutions for x, y, z, w, such that the standard signature is $\frac{1}{2\sqrt[3]{2}}[-1, 0, 4]$. The reduction from #3RBP-VC to #Pl-Pos-$\{F_{\{0,1\}}\}$-SAT is completed.

For $F_{\{0,1,2\}}$, we directly use it as a generator. The standard signature is $[1, 1, 1, 0]^{\mathrm{T}}$. By Eq. (1), the ValG of it is $[1, 0, 0, 1]^{\mathrm{T}}$, under the basis

$$[n, p] = \left[\begin{pmatrix} 0 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right].$$

By Eq. (2), the recognizer with standard signature $[3, -1, 0]$ has ValR $[0, 1, 1]$ under this basis. The recognizer is the same as the one for $F_{\{0,1\}}$, except that $F_{\{0,1\}}$ are replaced by $F_{\{0,1,2\}}$. The standard signature of it is $[4x^2z + 3x^2w + 8xyz + 6xyw + 4y^2z + 3y^2w, 4x^2z + 3x^2w + 6xyz + 5xyw + 2y^2z + 2y^2w, 4x^2z + 3x^2w + 4xyz + 4xyw + y^2z + y^2w]$, and there exist solutions to realize $[3, -1, 0]$. So, #Pl-Pos-$\{F_{\{0,1,2\}}\}$-SAT is #P-complete.

For $F_{\{0,1,3\}}$, we also use it as generator. The standard signature is $[1, 1, 0, 1]^{\mathrm{T}}$. By Eq. (1), the ValG of it is $[1, 0, 0, 1]^{\mathrm{T}}$, under the basis

$$\left[\begin{pmatrix} \frac{1+\sqrt{5}}{20}b \\ \frac{1}{10}b \end{pmatrix}, \begin{pmatrix} \frac{1-\sqrt{5}}{20}a \\ \frac{1}{10}a \end{pmatrix}\right],$$

where $a = \sqrt[3]{500 + 100\sqrt{5}}$ and $b = \sqrt[3]{500 - 100\sqrt{5}}$.

By Eq. (2), the recognizer with standard signature

$$\left[-\frac{40}{ab} + \frac{20}{a^2}, \frac{20}{ab} - \frac{10(1+\sqrt{5})}{a^2}, \frac{40}{ab} + \frac{10(3+\sqrt{5})}{a^2}\right]$$

has ValR $[0, 1, 1]$ under this basis. We use the gadget in Fig. 5(b) as the recognizer. The first and the third solid nodes are connected to a unary function $f_{x,y}$ by an edge variable, and the second is connected to a unary function $f_{z,w}$. The functions at solid nodes are $F_{\{0,1,3\}}$, and the functions at the first and the forth white nodes are binary equivalence relations, while the functions at the second and the third white nodes are binary inequivalence relations. The standard signature of it is $[3x^2z + x^2w + 2xyz + 2xyw + y^2w, x^2z + x^2w + 2xyz + 2xyw + y^2z, x^2w + 2xyz + y^2z + y^2w]$, and there exist solutions to realize the standard signature we want. So, #Pl-Rtw-$\{F_{\{0,1,3\}}\}$-SAT is #P-complete. We have proved the following theorem.

Theorem 11. *#Pl-Pos-$\{F_{\{0,1\}}\}$-SAT, #Pl-Pos-$\{F_{\{0,1,2\}}\}$-SAT and #Pl-Rtw-$\{F_{\{0,1,3\}}\}$-SAT are #P-complete.* □

Notice that, in the above theorem, the conclusion for $F_{\{0,1,3\}}$ is different from the other two relations. #Pos-$\{F_{\{0,1,3\}}\}$-SAT is what we call #Read-twice Positive Not-equal-2 3SAT in the introduction section. Now, we prove #Pos-$\{F_{\{0,1,3\}}\}$-SAT is polynomial time computable. We use #Pos'-$\{F_{\{0,1,3\}}\}$-SAT to denote a generalized version of this problem. The inputs are bipartite graphs with bipartition V_1 and V_2, such that all vertices in V_1 have degree no more than 2, and all vertices in V_2 have degree 3. The function at a degree 1 vertex is a constant function with value 1. The edge incident to a degree 1 vertex is called a dangling edge.

Given an instance of #Pos'-$\{F_{\{0,1,3\}}\}$-SAT, we can denote it by the Boolean formula $\varphi(x_1x_2\ldots x_n, y_1y_2\ldots y_m)$, in which $x_1, x_2, \ldots x_n$ occur once (corresponding to some of dangling edges), and $y_1, y_2, \ldots y_m$ occur at most twice. φ is the conjunction of $F_{\{0,1,3\}}$ in positive literals. We do not distinguish the formula and the corresponding graph. Suppose that T is a truth assignment for $x_1, x_2, \ldots x_n$, and T' is a truth assignment for $y_1, y_2, \ldots y_m$. Given T, $i, j \in 1, 2, \ldots n$ and $a, b \in \{0, 1\}$, the number of truth assignments T' such that φ is satisfied by truth assignment $x_i = a, x_j = b, x_k = T(x_k)(k \notin \{i, j\}), y_k = T'(y_k)$, is denoted by $\#(y_1y_2\ldots y_m)\varphi(x_i = a, x_j = b, T, y_1y_2\ldots y_m)$, or simply $\#\varphi(x_i = a, x_j = b)$. We say the instances of this problem are connected, if the corresponding graph is connected.

Lemma 12. *For any connected instance $\varphi(x_1x_2\ldots x_n, y_1y_2\ldots y_m)$ of #Pos'-$\{F_{\{0,1,3\}}\}$-SAT, any $i, j \in 1, 2, \ldots n$, and any truth assignment T for $x_1, x_2, \ldots x_n$,*

$$\begin{aligned}\#\varphi(x_i = 0, x_j = 1) &= \#\varphi(x_i = 1, x_j = 0),\\ \#\varphi(x_i = 0, x_j = 0) &= \#\varphi(x_i = 1, x_j = 0) + \#\varphi(x_i = 1, x_j = 1)\end{aligned}$$

hold.

Proof. It is easy to check that it holds for $\varphi(x_1x_2x_3) = F_{\{0,1,3\}}(x_1x_2x_3)$. We prove it by induction.

Consider two operations: turning a degree 1 vertex into a degree 3 vertex by adding two dangling edges to it, and merging two dangling edges into a non-dangling edge which connects the two degree 3 vertices. Obviously, any connected instance of #Pl-Pos'-$\{F_{\{0,1,3\}}\}$-SAT can be generated from $F_{\{0,1,3\}}(x_1x_2x_3)$ by the two kinds of operation. So, we only need to prove that the properties are preserved by the two operations.

For the first operation, suppose $\varphi(x_1x_2\ldots x_nz, y_1y_2\ldots y_m)$ and $\varphi' = F_{\{0,1,3\}}(z\ x_{n+1}x_{n+2})$ are two connected instances, which possess the properties. For any $i, j \in 1, 2, \ldots n+2$ and truth assignment T for $x_1, x_2, \ldots x_n + 2$, we prove that $\psi(x_1x_2\ldots x_{n+2}, y_1y_2\ldots y_mz) = \varphi \wedge \varphi'$ also possesses the properties.

Case 1, $i = n + 1$, $j = n + 2$. It is easy to check that the equations in the lemma hold for ψ in this case.

Case 2, $i, j \in \{1, 2, \dots n\}$. Since the equations for φ hold for any value of x_n, they also hold for ψ in this case, whose items are linear combinations of the ones for φ.

Case 3, $i \in \{n+1, n+2\}$, $j \in \{1, 2, \dots n\}$; without loss of generality, $i = n+1$, $j = 1$.

Case 3a, $T(x_{n+2}) = 0$. It is easy to see that

$$\begin{aligned}
\#\psi(x_1 = 0, x_{n+1} = 0) &= \#\varphi(x_1 = 0, z = 0) + \#\varphi(x_1 = 0, z = 1)\\
\#\psi(x_1 = 0, x_{n+1} = 1) &= \#\varphi(x_1 = 0, z = 0)\\
\#\psi(x_1 = 1, x_{n+1} = 0) &= \#\varphi(x_1 = 1, z = 0) + \#\varphi(x_1 = 1, z = 1)\\
\#\psi(x_1 = 1, x_{n+1} = 1) &= \#\varphi(x_1 = 1, z = 0).
\end{aligned}$$

So, the equations hold for ψ in this case.

Case 3b, $T(x_{n+2}) = 1$. $\varphi' = F_{\{0,1,3\}}(zx_{n+1}x_{n+2})$ is just $z = x_{n+1}$, because $T(x_{n+2}) = 1$. Obviously, the equations hold in this case.

The proof for the second operation is similar to that of the first one. □

Theorem 13. *#Pos'-$\{F_{\{0,1,3\}}\}$-SAT is polynomial time computable.*

Proof. Without loss of generality, we only consider the connected graphs. Given any connected instance $G = (V, E)$ of #Pos'-$\{F_{\{0,1,3\}}\}$-SAT. Take a spanning tree $H = (V, E' = \{y_1, y_2, \dots y_m\})$ of G. Obviously, all dangling edges belong to E'. Suppose that there are n edges $e_1, e_2, \dots e_n$ of G which are not in E'. Cut all of them into two dangling edges and denote the new instance by $\varphi(x_1x_2 \dots x_{2n}, y_1y_2 \dots y_m)$, where $x_1x_2 \dots x_{2n}$ are corresponding to all edges that have resulted by the cutting. Obviously, we can compute $\#(y_1y_2 \dots y_m)\ \varphi(T', y_1y_2 \dots y_m)$ in polynomial time for any truth assignment T' for $x_1x_2 \dots x_{2n}$, since φ is a tree.

The answer to G is just the summation of $\#(y_1y_2 \dots y_m)G(T, y_1y_2 \dots y_m)$ on all truth assignments T for $e_1, e_2, \dots e_n$. The weight of a truth assignment is just the number of variables assigned 1 by it. By Lemma 12, if T_1 and T_2 have the same weight, then

$$\#(y_1y_2 \dots y_m)G(T_1, y_1y_2 \dots y_m) = \#(y_1y_2 \dots y_m)G(T_2, y_1y_2 \dots y_m)\,.$$

Let T_i denote a truth assignment for $e_1, e_2, \dots e_n$ of weight i, and T'_i denote the corresponding truth assignment for $x_1x_2 \dots x_{2n}$. Then

$$\begin{aligned}
&\sum_T \#(y_1y_2 \dots y_m)G(T, y_1y_2 \dots y_m) = \sum_{i=0}^{n} \binom{n}{i} \#(y_1y_2 \dots y_m)G(T_i, y_1y_2 \dots y_m)\\
&= \sum_{i=0}^{n} \binom{n}{i} \#(y_1y_2 \dots y_m)\varphi(T'_i, y_1y_2 \dots y_m).
\end{aligned}$$

The theorem follows. □

6. Conclusion

The study of counting and its computational complexity is interesting and important. However, we only have a limited understanding of how the complexity of counting problems behaves in restricted cases. The results of this paper improve the situation somewhat, but there are still many open problems. We hope that the methods developed here are useful in obtaining more #P-completeness results for other restricted counting problems. The high-dimension polynomial interpolation is only used in Section 3, leading to a strong result. The results in other sections are not restricted to bipartite graphs. #3RBP-VC seems to be a good candidate for reductions. For example, consider #Pl-OR-$\{f\}$-SAT: if f can simulate a ternary function which takes nonzero values on $(0, 0, 0)$ and $(1, 1, 1)$, and value 0 on the other inputs, then this problem is #P-complete by a simple polynomial interpolation technique. But by a result of Cook [4] there are not many such ternary relations.

Acknowledgements

The authors are grateful to their supervisor Prof. Angsheng Li for advice and encouragement. All authors are supported by NSFC Grants no. 60325206 and no. 60310213.

References

[1] R. Bubley, M. Dyer, Graph orientations with no sink and an approximation for a hard case of #SAT, in: Proceedings of the Eighth Annual ACM–SIAM Symposium on Discrete Algorithms, 1997, pp. 248–257.

[2] J.-Y. Cai, V. Choudhary, Some results on matchgates and holographic algorithms, in: ICALP 2006, in: LNCS, vol. 4051, Springer-Verlag, 2006, pp. 703–714.

[3] J.-Y. Cai, V. Choudhary, Valiant's Holant theorem and matchgate tensors, in: Theory and Applications of Models of Computation: Third International Conference, TAMC 2006 in: LNCS, vol. 3959, Springer-Verlag, pp. 248–261.

[4] M. Cook, Networks of relations, Ph.D. Thesis, 2005.

[5] N. Creignou, M. Hermann, Complexity of generalized satisfiability counting problems, Information and Computation 125 (1996) 1–12.

[6] M.E. Fishier, Statistical mechanics of dimers on a plane lattice, Physical Review 124 (1961) 1664–1672.

[7] C.S. Greenhill, The complexity of counting colourings and independent sets in sparse graphs and hypergraphs, Computational Complexity 9 (2000) 52–72.

[8] H.B. Hunt III, M.V. Marathe, V. Radhakrishnan, R.E. Stearns, The complexity of planar counting problems, SIAM Journal on Computing 27 (1998) 1142–1167.

[9] P.W. Kasteleyn, The statistics of dimers on a lattice, Physica 27 (1961) 1209–1225.

[10] P.W. Kasteleyn, Graph theory and crystal physics, in: F. Harary (Ed.), Graph Theory and Theoretical Physics, Academic Press, London, 1967, pp. 43–110.

[11] M. Liskiewicz, M. Ogihara, S. Toda, The complexity of counting self-avoiding walks in subgraphs of two-dimensional grids and hypercubes, Theoretical Computer Science 304 (2003) 129–156.

[12] C. Papadimitriou, Computational Complexity, Addison-Wesley, 1994.

[13] H.N.V. Temperley, M.E. Fishier, Dimer problems in statistical mechanics an exact result, Philosophical Magazine 6 (1961) 1061–1063.

[14] S.P. Vadhan, The complexity of counting in sparse, regular, and planar graphs, SIAM Journal on Computing 31 (2001) 398–427.

[15] L.G. Valiant, The complexity of enumeration and reliability problems, SIAM Journal on Computing 8 (1979) 410–421.

[16] L.G. Valiant, Holographic Algorithms (extended abstract), FOCS 2004, pp. 306–315.

[17] L.G. Valiant, Accidental Algorithms, FOCS 2006, pp. 509–517.

[18] M. Xia, W. Zhao, #3-regular bipartite planar vertex cover is #P-complete, in: Theory and Applications of Models of Computation: Third International Conference, TAMC 2006, in: LNCS, vol. 3959, Springer-Verlag, pp. 356–364.